Improved Slope One Collaborative Filtering Predictor Using Fuzzy Clustering

Tianyi Liang[1], Jiancong Fan[1], Jianli Zhao[1,⋆],
Yongquan Liang[1], and Yujun Li[2]

[1] School of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266510, China
liangtee@126.com,zhaojianli@gmail.com,{fanjiancong,lyq}@sdust.edu.cn
[2] Hisense State Key Laboratory of Digital Multi-media Technology
liyujun@hisense.com

Abstract. Slope One predictor, an item-based collaborative filtering algorithm, is widely deployed in real-world recommender systems because of its conciseness, high-efficiency and reasonable accuracy. However, Slope One predictor still suffers two fundamental problems of collaborative filtering : sparsity and scalability, and its accuracy is not very competitive. In this paper, to alleviate the sparsity problem for Slope One predictor, and boost its scalability and accuracy, an improved algorithm is proposed. Through fuzzy clustering technique, the proposed algorithm captures the latent information of users thereby improves its accuracy, and the clustering mechanism makes it more scalable. Additionally, a high-accuracy filling algorithm is developed as preprocessing tool to tackle the sparsity problem. Finally empirical studies on MovieLens and Baidu dataset support our theory.

Keywords: Slope One, fuzzy clustering, collaborative filtering, sparsity, scalability.

1 Introduction

Slope One predictor[1] is a kind of item-based collaborative filtering (CF)[2–4] algorithm proposed by Daniel Lemire and Anna Maclachlan in 2005. It is designed as a concise and understandable form to make it easy to implement and maintain. Many empirical studies prove Slope One predictor is high-efficient and its prediction accuracy can be comparable with some much more complex algorithms. Because of its simplicity and efficiency, it has been applied in many recommender systems, such as hitflip, Value Investing News and AllTheBests.

However, the extensive application also reveals several shortcomings of Slope One predictor:

– **Sensitive to data sparsity:** like most CF algorithms, the performance of Slope One predictor will decrease badly when the data is sparse.

⋆ Corresponding author.

H. Motoda et al. (Eds.): ADMA 2013, Part I, LNAI 8346, pp. 181–192, 2013.

- **High algorithm complexity(Scalability):** suppose there are n users and m items, Slope One predictor requires $O(nm^2)$ time steps and $O(nm(m-1)/2)$ storage units to make recommendations. That makes it not very suitable for the large scale recommender system which needs to deal with millions of users and items.
- **Unremarkable accuracy:** the accuracy of Slope One predictor is not very outstanding, therefore it is often used as preprocessing or smoothing technique in practice.

As a promising branch of CF algorithm, Slope One predictor is worth being improved. In this paper, we adopt Fuzzy Clustering[5] technique to boost Slope One predictor. Our improved algorithm is :

- **Less sensitive to data sparsity**
- **High scalability**
- **High accuracy**
- **Still simplicity, easy to implement and maintain**

In the rest of this paper, we first provide a brief review of CF, and detailed descriptions of Slope One predictor and Fuzzy Clustering technique. Then in Section 3 we propose our algorithm, and comprehensively evaluate it by experiments in Section 4.

2 Background

2.1 State of the Art in Collaborative Filtering

To improve the scalability and alleviate the sparsity problem in CF, many approaches have been proposed. Sarwar et al. [6] proposed an Item-based CF that generates recommendations through comparing the similarity between items rather than users. The advantage of Item-based is the item similarity is relatively static, thus the computation of item similarity can be performed offline, which makes Item-based CF more scalable than User-based CF[2, 3]. Besides, Item-based and Use-based CF are also called Memory-based CF.

Model-based CF[2–4] is a family of algorithms which apply machine learning and data mining technique in CF to get better performance. Typical Model-based CF includes Regression-based[2, 3], Clustering-based[7], Classification-based[2] and MDP-based[2] etc. SVD[2–4, 8] and its variations (e.g. SVD++)[9, 10] which use Matrix Factorization technique to learn latent information from the original user-rating matrix are really popular in recent year because of its excellent performance in Netflix contest and KDD-Cup. Usually Model-based CF has more powerful performance and scalability than Memory-based CF, but the model-training process is expensive, and deploying it needs more domain knowledge.

Hybrid recommender[2, 3, 11] is widely employed in practice. A hybrid recommender usually blend several CF models, and some systems, such as Fab[12], combine Content-based model[13] with CF to get better performance. The research on multi recommender models ensemble has become a hot area.

2.2 Principle of Slope One Predictor

Slope One predictor works on the intuitive principle of a "popular differential" between items[1]. Concretely speaking, the "popular differential" reflects that how much better one item is liked than another, it can be measured through subtracting the mean ratings of two items. Formally the predictor is based on a simplified regression model : $f(x) = x + b$ where b is defined as the mean deviation of the item to be predicted and other items. The algorithm performs the computation on users who have rated these items. Given an user u , to predict his rating on item j , the mean deviation is calculate by equation 1:

$$dev_{j,i} = \frac{1}{|U_j \cap U_i|} \sum_{n \in U_j \cap U_i} (r_{n,j} - r_{n,i}) \tag{1}$$

And the final prediction is :

$$p_{u,j} = \frac{1}{|R_u|} \sum_{i \in R_u} (r_{u,i} + dev_{j,i}) \tag{2}$$

where U_i and U_j are respectively the sets of users who rated item i and item j . R_u is the set of ratings of user u. To improve the accuracy, Weighted Slope One[1] revises equation 2 by taking the number of ratings into consideration:

$$p_{u,j} = \frac{1}{\sum\limits_{i \in R_u} Num_{j,i}} \sum_{i \in R_u} (r_{u,j} + dev_{j,i}) Num_{j,i} \tag{3}$$

Besides, Bi-Polar Slope One[1] improves accuracy by dividing items into user rated positively and negatively.

2.3 Fuzzy Clustering and Its Advantages

Clustering is a process of dividing data into different clusters and putting similar data elements into same cluster. Xue et al. [7] applied clustering technique to improve User-based CF from the following aspects:

- Increasing scalability: the scope of similarity calculation narrows to cluster rather than whole dataset.
- Increasing data density: the missing ratings in cluster can be smoothed by cluster mean rating.

For hard clustering technique, such as K-Means[5], each user must belong to exactly one clustering while in Fuzzy Clustering (FC) (or called soft clustering) each user can belong to more than one clusters. Given an user u , FC uses membership degree $w_{u,j}$ to represent the association strength between user u and cluster j. Suppose there are n users and k clusters, the results of FC satisfy the following three conditions simultaneously:

(1) For each user u and cluster j, $0 \leq w_{u,j} \leq 1$

(2) For each user u, $\sum_{j=1}^{k} w_{u,j} = 1$

(3) For each cluster j, $0 < \sum_{i}^{n} w_{u,j} < n$

Compared with hard clustering, fuzzy clustering is more suitable for the real-world recommender systems. For example, in a movie recommender system, a filmnik may not only likes action movies but also enjoys comedies. Putting him into only one cluster, "Action Fans Cluster" or "Comedy Fans Cluster", may be too rigorous, the better solution is letting him belonging to both of the two clusters.

3 Fuzzy Clustering-Based Slope One Predictor

3.1 Philosophy of Proposed Approach

As is described in Section 2.2, Slope One predictor is based on the "item popularity differential" principle which is measured by the mean deviation among ratings. The original deviation computing method (equation 1) is quite concise but not accurate enough, because it does not take the association between users into consideration, causing much valuable latent information of users are ignored. For instance, in a movie recommender system, users can be divided into groups according to their favorite movie genres, such as "action movie" group, "love movie" group and "commedy movie" group. For a movie, the popular differentials (the deviation values) about it in diffenent user groups reflect how much it is liked by these group of users. When to make predictions for a given user, if the system knows how much importance of each user group for him and translates the "importance" into numeric values, the popular differential can be calculated by weighted summation of each deviation from different user groups, and that will generate more accurate predictions. In order to implement this idea, our approach employs Fuzzy Clustering to divide users into different groups and uses the membership degrees as weight coefficients to adjust the deviations from different clusters, finally obtaining weighted mean deviations. Thus, formally, the equation 1 is updated to equation 4 :

$$dev_{j,i} = \sum_{k=1}^{K} (w_{u,k} \times sub\_dev_{j,i,k}) \quad (4)$$

where $w_{u,k}$ denotes the membership degree between user u and cluster k , $sub\_dev_{j,i,k}$ is the rating deviation between item j and i in cluster k. Compared with equation ref1 , the calculating method of sub_dev reduces calculation greatly. Details and complexity analysis of it are presented in Section 3.3.

Additionally, a quick and accurate filling algorithm is proposed to improve the performance of fuzzy clustering on sparse data, its details are described in Section 3.2. Finally, the framework of our algorithm (FC-SLP) is :

- **1. Data preprocess :** fill the sparse dataset using filling algorithm.
- **2. Cluster users :** perform fuzzy clustering on users.
- **3. Prediction:** generate predictions based on the works of step 1&2.

The symbols used in this paper are shown in Table 1:

Table 1. Symbols Used throughout This Paper

Symbol	Description
U_i	the set of users who rated item i
R_u	the rating set of user u
D	the dataset
$w_{u,j}$	the membership degree of user u to cluster j
c_j	the feature vector of centroid j
$wr_{m,j}$	the weighted mean rating of item m in cluster j, it is a feature value of c_j
RC_j	the rating set of cluster j
$r_{u,m}$	the rating on item m of user u
$\bar{r}_m$	the mean rating of item m
$sub\_dev_{j,i,k}$	the weight mean deviation of item j and item i in cluster k
$U_{bestC=j}$	for any user u in set $U_{bestC=j}$, $w_{u,j}$ is his **maximum** membership degree value.

3.2 Data Preprocess: Filling Algorithm

Filling the unknown ratings is the most direct way to densify the sparse data thereby boost the performance of fuzzy clustering. The effectiveness of this strategy mainly depends on the accuracy of the filling algorithm, thus, it is crucial for this strategy to select a appropriate filling algorithm. In our scheme, a high-accuracy filling algorithm (HAF) is proposed: formally, the items are divided into two sets : $D_{positive} = \{m \in D \mid \bar{r}_m \geq \bar{r}\}$ and $D_{negative} = \{m \in D \mid \bar{r}_m < \bar{r}\}$. Let $diff_{u,positive}$ and $diff_{u,negative}$ be the rating differentials of user u on "good" and "bad" items, they are computed respectively by equation 5 and 6 :

$$diff_{u,positive} = \frac{\sum\limits_{m \in R_u \cap D_{positive}} (r_{u,m} - \bar{r}_m)}{|R_u \cap D_{positive}|} \tag{5}$$

$$diff_{u,negative} = \frac{\sum\limits_{m \in R_u \cap D_{negative}} (r_{u,m} - \bar{r}_m)}{|R_u \cap D_{negative}|} \tag{6}$$

Finally the prediction is given by equation 7 :

$$p_{u,m} = \begin{cases} \bar{r}_m + diff_{u,positive} \ , & m \in D_{positive} \\ \bar{r}_m + diff_{u,negative} \ , & m \in D_{negative} \end{cases} \tag{7}$$

Evidently, through the filling algorithm, popular items (items with large quantity of ratings) will be filled with more accurate predictions than unpopular ones because popular items have more ratings. Thus, to improve the filling effectiveness, and keep the filling diversity, the wait-to-be-filled items are sampled according to the probability model :

$$ps_i = \frac{|U_i|}{\sum\limits_{n \in D} |U_n|} \tag{8}$$

where ps_i is the probability of item i to be sampled.

With the probability model, popular items have higher probability to be sampled, and unpopular items also have chance to be filled, which balances the accuracy and diversity.

3.3 Cluster Users through Fuzzy Clustering Technique

Define $E(C)$ is the object function :

$$E(C) = \sum_{i=1}^{n} \sum_{j=1}^{k} w_{u,j}^2 Similarity(u, c_j)^2 \tag{9}$$

where $Similarity(u, c_j)$ is the similarity calculation of user u and centroid j. The goal of our scheme is maximizing the function $E(C)$.

Cluster Users through Fuzzy Clustering
Input :
dataset D , cluster number K .
Output :
$C = \{c_1, c_2, ..., c_K\}$: a list of centroids.
$W_{N \times K} = \{w_{u,k}\}$: a membership degree matrix.
Procedure :
1. random select K users as initial centroids.
2. repeat:
3. re-compute the centroid of each cluster by equation 10 .
4. update matrix $W_{N \times K}$ by equation 11 .
5. until $\mathbf{E(C)}$ convergences.

For each feature value $wr_{m,j}$ of centroid c_j :

$$wr_{m,j} = \sum_{u \in U_m \cap n \in U_{bestC=j}} \frac{w_{u,j}}{\sum\limits_{n \in U_m \cap n \in U_{bestC=j}} w_{n,j}} \times r_{u,m} \tag{10}$$

For $w_{u,j}$ in matrix $W_{N\times K}$, it is updated by the following equation 11 :

$$w_{u,j} = \frac{Similarity(u, c_j)}{\sum_{j=1}^{K} Similarity(u, c_j)} \tag{11}$$

where $Similarity(u, c_j)$ is calculated by Weighted-Pearson[3]:

$$Similarity(u, c_j) = \begin{cases} S_{pearson}(u, c_j)\frac{|R_u \cap RC_j|}{50} & |R_u \cap RC_j| < 50 \\ S_{pearson}(u, c_j) & otherwise \end{cases} \tag{12}$$

After filling processing, the ratings of users consist of real ratings and filling ratings. Xue et al. [7] differentiated them by setting different weights, but the weights need to be tuned manually. In our scheme, the weights are determined by the user itself rather than empirical rules. Define $\lambda_{u,m}$ is the weight of rating m for user u :

$$\lambda_{u,m} = \begin{cases} 1 & m \in R_u \\ \left(\frac{|\hat{R}_u|}{|R_u| + |\hat{R}_u|}\right)^2 & m \in \hat{R}_u \end{cases} \tag{13}$$

where $\hat{R}_u$ is the set of filling ratings of user u . Equation 14 is a modified Pearson Correlation which takes the λ into consideration:

$$S_{pearson}(u, c_j) = \frac{\sum_{m \in R_u \cap RC_j} \lambda_{u,m} \cdot (r_{u,m} - \bar{r}_i)(wr_{j,m} - \overline{wr}_j)}{\sqrt{\sum_{m \in R_u \cap RC_j} \lambda_{u,m}^2 \cdot (r_{u,m} - \bar{r}_i)^2} \sqrt{\sum_{m \in R_u \cap RC_j} (wr_{j,m} - \overline{wr}_j)^2}} \tag{14}$$

3.4 Prediction

The improved deviation calculation method has been represented by equation 4 where $sub\_dev_{j,i,k}$ is defined as equation 15:

$$sub\_dev_{j,i,k} = wr_{j,k} - wr_{i,k} \tag{15}$$

Finally, the prediction is given by equation 16:

$$p_{u,j} = \frac{1}{|R_u|} \sum_{i \in R_u} (r_{u,i} + \sum_{c=1}^{K} w_{u,c} \times (wr_{j,k} - wr_{i,k})) \tag{16}$$

Consider a database consists of n users and m items, FC-SLP algorithm consumes $O(km^2)$ time steps and $O(km(m-1)/2)$ storage units (k is the number of clusters) to make predictions comparing original Slope One's $O(nm^2)$ and $O(nm(m-1)/2)$. Due to k is far less than n , therefore the cost of computation and storage of FC-SLP algorithm is largely reduced, which makes it more scalable.

4 Experiments

4.1 Dataset, Evaluation Metric and Algorithms

To evaluate our algorithm more comprehensively, the experiments are conducted on two popular datasets: MovieLens and Baidu contest[16]. The dataset provided by Baidu is used to support the movie recommendation algorithm contest organized by Baidu in 2013. It contains real 1262741 ratings of 9722 users on 7889 movies. We randomly select 30% of users from the whole Baidu dataset to be experimental data. Details of them are shown in Table 2 :

Table 2. Details of MovieLens and Baidu datasets

	MovieLens	Baidu
No. of users	943	2917
No. of items	1682	7800
No. of ratings	100000	365639
Rating scale	1-5	1-5
Sparsity	6.3%	1.6%
Domain	Movie rating	Movie rating

We take RMSE (Root Mean Square Error) as evaluation metric. The formula definition of RMSE is :

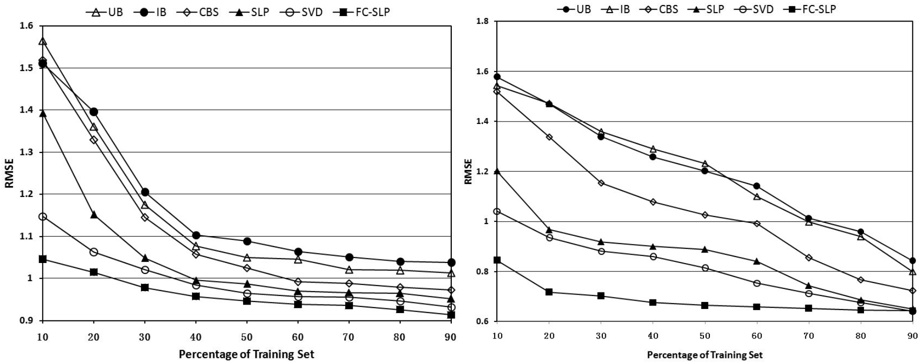
$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (p_i - r_i)^2}{n}} \tag{17}$$

A set of representative CF algorithms are chosen as comparisons, they are :

- User-based CF (UB)[3]
- Item-based CF (IB)[6]
- Cluster-based smoothing CF (CBS)[7]
- Weighted-Slope One (SLP)[1]
- SVD++[10]

4.2 Methodology

According to the principle of cross-validation, the dataset is divided into ten subsets (the ratings of each user are divided equally into ten parts), and each experiment is iterated for ten times. In each iteration, we randomly select N (0 < N < 10) subsets as test set and merge remaining ones as training set. For the experiments which need to continually change the percentage of training set, the value of N is from 1 to 9. The final results are the mean of ten times iterations.



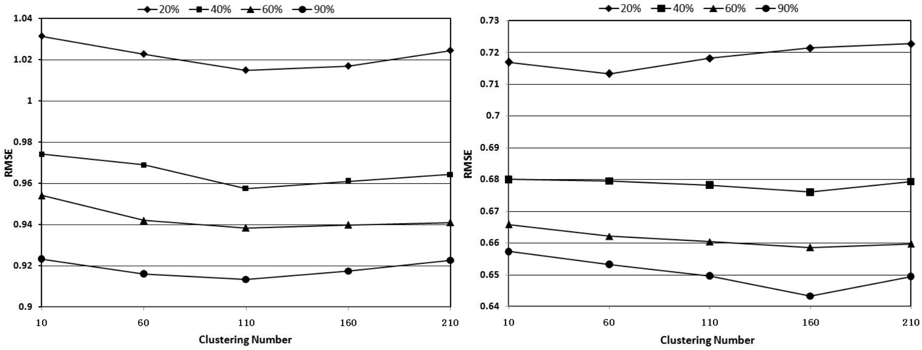
(a) MovieLens (b) Baidu

Fig. 1. Evolution of RMSE according to percentage of training set

4.3 Results and Discussions

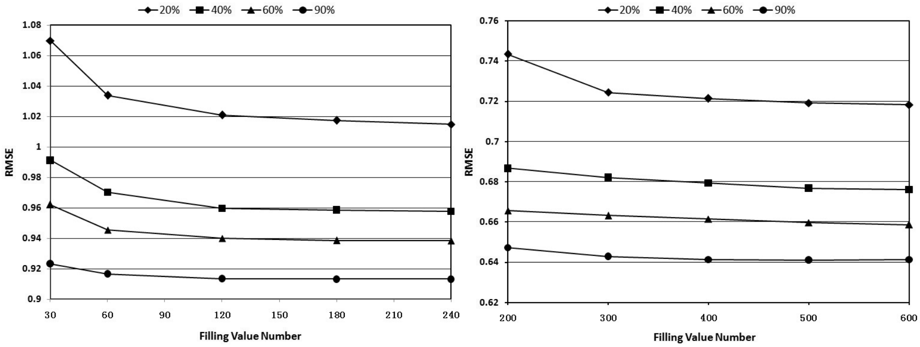
Experiment 1: Accuracy and Sparsity. In this experiment the percentage of training set is continually changing to simulate the different sparsity conditions. The results presented in Fig. 1 indicate that when the data is relatively dense (percentage>50%), the accuracy differentials among the Model-based CF(CBS, SLP, SVD++, FC-SLP) are not very obvious. However, as the data turns to be sparse, the differentials become large. At sparse conditions (percentage $< 50\%$), only FC-SLP algorithm maintains reasonable accuracy, its performance markedly exceeds other algorithms. The excellent results largely because, through the fuzzy clustering technique, FC-SLP algorithm has the capacity to capturing the latent information of users, and the preprocessing of filling significantly improves the performance of fuzzy clustering on sparse data, thereby boost the accuracy of FC-SLP algorithm at sparsity condition. The CBS algorithm, which is based on k-means clustering, shows mediocre results. That is because, as is discussed in Section 2.2, the hard clustering technique is not very appropriate for CF systems. Besides, there is no data preprocessing mechanism in CBS algorithm, which causes the effectiveness of k-means clustering declines badly on sparse data, thereby drag the performance of CBS.

Experiment 2: Accuracy and Cluster Number. A serious of experiments are conducted to explore the correlation between the cluster number (K) and the performance of FC-SLP algorithm. Results shown in Fig. 2 demonstrate that the number has certain effect on the performance, but this is no linear relationship between cluster numbers and accuracy, the optimal K value is an empirical value. The reasons behind this phenomenon is : the small cluster number makes the latent information extracted by fuzzy clustering too general, inversely, the large clustering number makes the latent information too discrete.



(a) MovieLens (b) Baidu

Fig. 2. Evolution of RMSE according to cluster number



(a) MovieLens (b) Baidu

Fig. 3. Evolution of RMSE according to filling value number

Experiment 3: Accuracy and Filling Value Number. Results depicted in Fig. 3 show that the number of filling values does affect the accuracy of FC-SLP algorithm. The accuracy keeps improving as the number of filling values increases, but the improvement will be more and more tinier. Considering the large quantity of filling values will extend the time of fuzzy clustering processing, thus, the number of filling values should not be too large.

Experiment 4: Comparison of Filling Algorithms. This experiment compares the accuracy of our proposed filling algorithm (HAF) and two typical CF filling algorithm : item average (IA) and item-user average (IUA). Results shown in Table 3 demonstrate that the proposed filling algorithm markedly outperforms other algorithms.

Table 3. RMSE of different filling algorithms

	MovieLens			Baidu		
	IA	IUA	HAF	IA	IUA	HAF
20%	1.0941	1.0622	0.9978	0.9794	0.9265	0.9187
40%	1.0433	0.9974	0.9852	0.9497	0.8962	0.8921
60%	1.0377	0.9825	0.9743	0.9018	0.8809	0.8723
90%	1.0261	0.9669	0.9573	0.7062	0.6730	0.6651

5 Conclusion

In this paper, we propose an improved Slope One predictor, which uses fuzzy clustering technique to alleviate the sparsity problem and boost scalability. A high-accuracy filling algorithm is developed as preprocessing tool to improve the effectiveness of fuzzy clustering on sparse data. Experiments on MovieLens and Baidu datasets demonstrate that our algorithm has outstanding prediction accuracy and scalability, what is more, it maintains high performance on sparse data.

On this basis, we aim to develop an automatic mechanism for setting optimal cluster number, replacing the pure empirical method in current scheme. The automatic mechanism is able to select the optimal cluster number through analyzing the dataset, without any manual intervention.

Acknowledgements. This work is supported by the national 973 plan project of China(No.2012CB724106), National Nature Science Foundation of China (No.61203305), Natural Science Foundation of Shandong Province (No.ZR2010FQ021, ZR2012FM003), Key science and technology project of Qingdao economic and Technological Development Zone(No.2013-1-25).

References

1. Lemire, D., Maclachlan, A.: Slope one predictors for online rating-based collaborative filtering. J. Society for Industrial Mathematics 5, 471–480 (2005)
2. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. J. Advances in Artificial Intelligence, 4 (2009)
3. Cacheda, F., Carneiro, V., Fernndez, D., et al.: Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems. J. ACM Transactions on the Web (TWEB) 5(1), 2 (2011)
4. Koren, Y., Bell, R.: Advances in collaborative filtering: Recommender Systems Handbook, pp. 145–186. M. Springer US (2011)
5. Han, J., Kamber, M., Pei, J.: Data mining: concepts and techniques. M. Morgan kaufmann (2012)

6. Sarwar, B., Karypis, G., Konstan, J., et al.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, pp. 285–295. ACM (2001)
7. Xue, G.R., Lin, C., Yang, Q., et al.: Scalable collaborative filtering using cluster-based smoothing. In: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 114–121. ACM (2005)
8. Sarwar, B., Karypis, G., Konstan, J., et al.: Application of dimensionality reduction in recommender system-a case study. Minnesota Univ. Minneapolis Dept. of Computer Science (2000)
9. Ma, C.-C.: A Guide to Singular Value Decomposition for Collaborative Filtering (2008)
10. Koren, Y.: Factorization meets the neighborhood: a multifaceted collaborative filtering model. In: Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 426–434. ACM (2008)
11. Wu, K.W., Ferng, C.S., Ho, C.H., et al.: A two-stage ensemble of diverse models for advertisement ranking. In: KDD Cup 2012 ACM SIGKDD KDD-Cup WorkShop (2012)
12. Balabanovi, M., Shoham, Y.: Fab: content-based, collaborative recommendation. J. Communications of the ACM 40(3), 66–72 (1997)
13. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: Brusilovsky, P., Kobsa, A., Nejdl, W. (eds.) Adaptive Web 2007. LNCS, vol. 4321, pp. 325–341. Springer, Heidelberg (2007)
14. Gao, M., Wu, Z.: Personalized context-aware collaborative filtering based on neural network and slope one. In: Luo, Y. (ed.) CDVE 2009. LNCS, vol. 5738, pp. 109–116. Springer, Heidelberg (2009)
15. Wu, J., Li, T.: A modified fuzzy C-means algorithm for collaborative filtering. In: Proceedings of the 2nd KDD Workshop on Large-Scale Recommender Systems and the Netflix Prize Competition. ACM (2012)
16. The download link of Baidu contest dataset, `http://pan.baidu.com/share/link?shareid=340221&uk=2000006609`