A Novel Clustering Based Collaborative Filtering Recommendation System Algorithm

Qi Wang, Wei Cao and Yun Liu

Abstract Traditional collaborative filtering algorithms compute the similarity of items or users according to a user-item rating matrix. However, traditional collaborative filtering algorithms face very severe data sparsity, which causes a discount of the performance of recommendation. In this paper, we proposed an improved clustering based collaborative filtering algorithm for dealing with data sparsity. We first clustered the users set into k clusters using K-means algorithm. Then we presented a formula to estimate those absent ratings in the user-item rating matrix and acquired a high density matrix. After that, we use the new rating matrix to calculate the similarity of items and predict the ratings of a target user on items which have not been rated and recommend Top-N items to the target user. We also implemented experiments and demonstrated that our proposed algorithm has better accuracy than traditional collaborative filtering algorithms.

Keywords Collaborative filtering · Recommendation systems · K-means algorithm · Data sparsity

Q. Wang · Y. Liu (✉)
School of Communication and Information Engineering, Beijing Jiaotong University, Beijing 100044, China
e-mail: liuyun@bjtu.edu.cn

Q. Wang
e-mail: 12125042@bjtu.edu.cn

Q. Wang · Y. Liu
Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China

W. Cao
China Information Technology Security Evaluation Center, Beijing, China
e-mail: caow@itsec.gov.cn

Introduction

Recommendation systems can automatically recommend to users what they might be interested in. Usually we divide recommendation system algorithms into content-based algorithms [1, 2] and collaborative filtering algorithms [3–6]. Content-based algorithms recommend to users items which are similar to what users have already bought or rated by analyzing the features of users or items. These algorithms can solve the problem called "cold start" and also won't face the challenge of data sparsity because they don't depend on the rating matrix. But they have a serious drawback that they can't deal with pictures, video, music and other products difficult to be analyzed and extracted features from. On the contrary, collaborative filtering algorithms utilize a user-item rating matrix to calculate the similarity between users or items and then predict those items which have not been rated or bought depending on the ratings of neighbors which have high similarity with the target users. However, the number of items which each user has bought is usually less than 1 % of the total number of items in a site, which causes severe data sparsity and a decrease of the performance.

In this paper, we proposed an improved clustering based collaborative filtering algorithm for dealing with data sparsity. We combined K-means algorithms and a formula dealing with data sparsity of the user-item matrix. After that we implemented some experiments and it was shown that our proposed algorithms have a better performance than the traditional algorithms.

The Proposed Algorithm

Traditional collaborative filtering algorithms [4] create a user-item rating matrix and calculate similarity between users or items, but very often this matrix is sparsely populated leading to poor coverage of the recommendation space and ultimately limiting recommendation effectiveness. Our proposed method combines K-means algorithm [7] and a formula which can assign an estimation rating to an unrated item, so we can get a high-density matrix and resolve the data sparsity.

User Clustering

The first step is user clustering, and clustering is a preliminary step for the subsequent step to gather those similar users. In this paper, we use K-means algorithm to cluster our user set in the user-item rating matrix. Furthermore, we use Euclidean distance to represent the distance between users. Let a centroid is denoted by $U_{mid} = (r_{01}, r_{02}, \ldots, r_{0n})$ and user i can be denoted by $U_i = (r_{i1}, r_{i2}, \ldots, r_{in})$, where r_{mn} states the rating of user m on item n, then the distance between the two users is given by,

Fig. 1 K-means clustering algorithm

```
Input: k original user centroids
1.for each user vector U
2.    for the k^th centroid C
3.       distance[k]=Euclidean(U,C);
4.    end
5.    find the shortest distance—distance[i]
6.    assign user U to cluster i
7.end
```

$$D = \sqrt{\sum_{j=1}^{n} (r_{ij} - r_{oj})^2} \tag{1}$$

Figure 1 shows the process of the K-means algorithm in our recommendation system and Fig. 2 shows the result of the K-means algorithm.

Constructing the Rating Matrix

Assuming there are M users and N items in the rating matrix, we define the sparsity level of the matrix as 1 – the number of ratings/M*N. Usually the sparsity level is very high, so in order to resolve this problem, we proposed a formula to calculate an estimation rating for an absent rating. The estimation rating of user c on item s, $R_{c,s}$ is computed as,

$$R_{c,s} = \bar{R}_c + \frac{1}{|U|}\sum_{\hat{c}\in U}(R_{\hat{c},s} - \bar{R}_{\hat{c}}) \tag{2}$$

where $\bar{R}_c$ is the average rating of the user c, U is the set of users that belong to the same cluster which is formed through the K-means algorithm and moreover have rated item s. Because $R_{c,s}$ is computed by the users in one same cluster and users belonging to one same cluster have higher similarity with each other, the estimation rating is more accurate. In addition, the rating scales of different users are varied, so in order to eliminate this inaccuracy, we let a rating subtract the average rating of a user and utilize the difference to calculate the predicting rating. Apparently, this method can eliminate data sparsity.

Similarity Computation

Similarity computation is the most important step for collaborative filtering algorithms. There are three different ways to compute the similarity between items. They are cosine-based similarity, Person correlation-based similarity and adjusted cosine similarity respectively as shown in Eqs. (3)–(5). We define the set

	Item2	Item3	…	…	Item i-1	Item i
U1	3					4
U345	3				5	
U143		3			2	4
…	…	…	…	…	…	…
Ui			5	6	6	
Uj		4	4	5	5	
Um			3		4	1
Un			2	3	5	5

Fig. 2 The result of clustering

of users that have rated both item i and item j as U, and the average rating of user u is denoted by $\bar{R}_u$. $\bar{R}_i$ and $\bar{R}_j$ denote the average rating of item i and item j respectively.

- Cosine-based Similarity

$$sim\,(i,j) = \cos(\vec{i},\vec{j}) = \frac{\vec{i}\cdot\vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2} \tag{3}$$

- Person Correlation-based Similarity

$$sim\,(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_i)(R_{u,j}-\bar{R}_j)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_i)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_j)^2}} \tag{4}$$

- Adjusted Cosine Similarity

$$sim\,(i,j) = \frac{\sum_{u\in U}(R_{u,i}-\bar{R}_u)(R_{u,j}-\bar{R}_u)}{\sqrt{\sum_{u\in U}(R_{u,i}-\bar{R}_u)^2}\sqrt{\sum_{u\in U}(R_{u,j}-\bar{R}_u)^2}} \tag{5}$$

In the paper [3], Sarwar et al. have demonstrated that adjusted cosine similarity performs best among them in the recommendation system through their experiments. So, in this paper, we will use adjusted cosine similarity as the similarity computation method.

Prediction and Recommendation

After the former similarity computation, we will get a $N * N$ similarity matrix, where N represents the total number of items. The last step is predicting and recommendation. Firstly, we predict the items which have not been bought or rated by the target user, after that we recommend the top-N items to the target user basing the predicting. We use weighted sum to calculate the predicting ratings. Let the target user is u and the unrated item is i. $N =$ {all the items that have high similarity with i and have been rated by user u in the mean time}.The predicting rating of user u on the item i is denoted by the Eq. (6).

$$R'_{u,i} = \frac{\sum_N (sim_{i,N} * R_{u,N})}{\sum_N (|sim_{i,N}|)} \tag{6}$$

Experimental Evaluation

Our data set is from the Movielens which is a web-based research recommender system. The data set includes 100,000 ratings of 943 users on 1,682 items and each user has rated 20 or more movies. The data set is divided into a training set and a test set. The 80 % of the data is used as the training set and the rest 20 % is used as the test set. All our experiments were implemented at Matlab7.0 and run in a PC with Intel Core processor having a speed of 2.2 GHz and 2 GB of RAM.

Metric

We use Mean Absolute Error (MAE) as our evaluation metric, and the MAE is defined as,

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |P_i - R_i| \tag{7}$$

where N states the total number of predicting ratings in the test set, P_i is the ith predicting rating, and R_i is the ith actual rating in the test set.

Experimental Results

First, we implemented an experiment to see the performance of recommendation with the increasing of the density of rating matrix. We let the density of rating

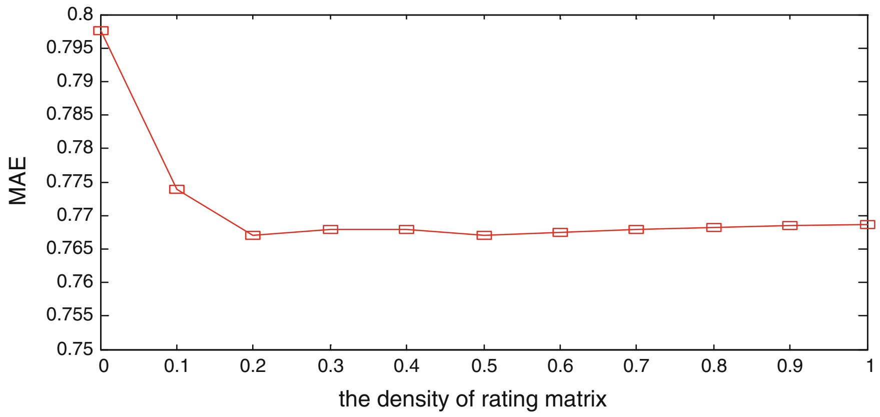

Fig. 3 Performance of the algorithm under different data sparsity

matrix increase at the speed of 10 % and then computed their MAE of predicting ratings. In addition, in this experiment, we set the parameter k = 30 in K-means algorithm and we didn't design an extra experiment to determine the optimal k. The results are shown in Fig. 3. From the Fig. 3 we can observe that with the increasing of the density of rating matrix the MAE of predicting decreases, which means the performance of recommendation getting better. Furthermore, the performance of recommendation improves as we increase the density of rating matrix from 0 to 20 %, after that the curve tends to be flat. With the increasing of the density of rating matrix, the computation also will increase rapidly, so we select the 20 % as the optimal choice of the density. On the other hand, the optimal value of sparsity level is 80 %.

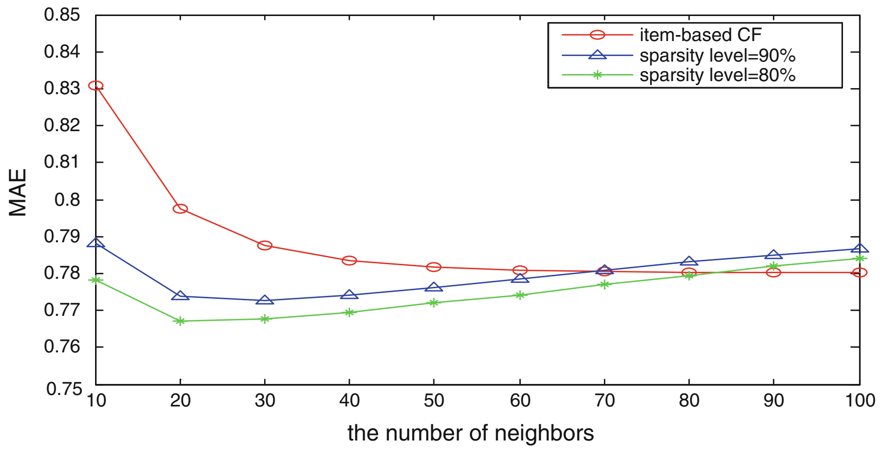

Fig. 4 Performance of the item-based algorithm and our proposed algorithm under the different number of neighbors

Then, we performed an experiment where we varied the number of neighbors and compared the results of the proposed algorithm with the traditional item-based collaborative filtering algorithm. The results are shown in Fig. 4. There are three curves in the figure which represent the normal item-based algorithm, the proposed algorithm where sparsity level equals 90 %, and the proposed algorithm where sparsity level equals 80 %. From the figure, we can observe that our proposed algorithm performs better than the item-based collaborative filtering when the number of neighbors is from 10 to 70, and our proposed algorithm can acquire the minimum MAE when the sparsity level = 80 % and the neighbors = 20.

Conclusion

In this paper, we proposed a new CF recommendation algorithm which introduces the K-means algorithm and estimation ratings to resolve data sparsity. By employing the experiments we observe the performance of our proposed algorithm is much better than the traditional CF algorithm and we also find when de sparsity level = 80 % and the number of neighbors = 20 the MAE of predicting ratings is minimum and the performance is the best. On the other hand, our proposed algorithm resolves data sparsity and acquires a more accurate recommendation. In addition, we can implement more experiments to determine the optimal number k of clusters in the K-means algorithm.

Acknowledgments This work has been supported by the National Natural Science Foundation of China under Grant 61172072, 61271308, the Beijing Natural Science Foundation under Grant 4112045, the Research Fund for the Doctoral Program of Higher Education of China under Grant W11C100030, the Beijing Science and Technology Program under Grant Z121100000312024.

References

1. Balabnovic M, Shoham Y (1997) Fab: content-based, collaborative recommendation. Comm ACM 40(3):66–72
2. Mooney RJ, Bennett PN, Roy L (1998) Book recommending using text categorization with extracted information. In: Procedings of recommender systems papers from 1998 workshop, technical report WS-98-08
3. Linden G, Smith B, York J (2003) Amazon. com recommendations: Item-to-item collaborative filtering. IEEE Internet Comput 7(1):76–80
4. Sarwar B, Karypis G, Konstan J et al (2001) Item-based collaborative filtering recommendation algorithms. In: Proceedings of 10th international WWW conference, Hong Kong, pp 1–5
5. Deshpande M, Karypis G (2004) Item-based top-N recommendation algorithms. ACM Trans Inf Syst 22(1):143–177

6. Koren Y (2010) Factor in the neighbors: scalable and accurate collaborative filtering. ACM Trans Knowl Disc Data 4:1–24
7. Kim KJ, Ahn HC (2008) A recommender system using GA K-means clustering in an online shopping market. Expert Syst Appl 34:1200–1209