



A collaborative filtering recommendation algorithm based on information theory and bi-clustering

Mingyang Jiang¹ · Zhifeng Zhang¹ · Jingqing Jiang¹ · Qinghu Wang¹ · Zhili Pei¹

Received: 18 September 2018 / Accepted: 18 December 2018 / Published online: 4 February 2019
© Springer-Verlag London Ltd., part of Springer Nature 2019

Abstract

Collaborative filtering is the most popular and efficient recommendation algorithm to character the potential preference of the new users, by exploring the patterns of historical consuming records/ratings of the investigated users. There are two types of primary collaborative filtering algorithms: the user-based recommendation system, which recommends items to new users by ranking the similarity of the shared items between the history users and the new users, and the item-based collaborative filtering recommend items to new users by considering the rank of the similarity among all the history items of the training data. Although the collaborative filtering has been successfully applied to many commercial fields, several original drawbacks of collaborative filtering, especially the sparsity of the rating data raises a serious challenge to the accuracy and the universality of those algorithms. In particular, the most rating terms for each specific user are missing in many applications, and the performance of collaborative filtering will be degraded along with the increment of the number of items in training dataset. In this paper, we proposed a novel collaborative filtering method (CBE-CF) to extract the local dense rating modules to cope with the data sparsity and the computational efficiency of the traditional recommendation algorithms, by introducing the information entropy and bi-clustering into collaborative filtering. Here, both the rows and columns of the user-item-rating matrix are clustered together to identify the dense rating modules of the historical records (training) data, and then an information entropy metric is used to quantify the similarity between the new user and each dense modules, and the final prediction is optimized by the aggregative recommendations of the global generalization of item-based methods and the local similarity of the nearest modules. Experimental analysis presents the characters of the proposed CBE-CF, and the precision and the computational cost, etc., are better than state of the art on the benchmark dataset.

Keywords Collaborative filtering · Information entropy · Bi-clustering · Recommendation system

1 Introduction

The explosive growth of the diversity commercial data online has more and more given rise to the difficulty in extracting the interesting and valuable information from the huge redundant data warehouse, such as there are more than 17,000 movies in the Netflix Web site, and the kindle component of Amazon Web site already provide about 41,000 books to be selected [1, 2]. And a data-based

automatic personalized recommendation system is helpful to relief this information overload. Recently, many online shopping Web sites, such as Amazon and Alibaba have taken advantage of the recommendation services to improve the sales and comfort the user's experience. Generally, the collaborative filtering (CF) and the content-based recommending are the two main types of recommendation systems, with the great development of historical user data. The CF-based methods also are raised quickly, as CF-based methods predict the item preference of new users by ranking the similarities between the new users and historical users, and the performance of those methods are large depended on the quality and quantity of the historical users. Specifically, the user-based CF considers the item similarity among the historical users as the main reference of the recommendation of new users [3, 4],

✉ Zhili Pei
zhilipei@imn.edu.cn

¹ College of Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao 028000, China

while the reference standard of the item-based CF depends on the user similarities among the historical items [5].

In practice, the efficacy of CF models also is limited by the sparsity of the rating matrix of historical users and the cold start of new users [2, 6–8]. The data sparsity indicates that just a few items are rated records by the historical users, for example, there are less than 2% movies in the film Web site browsed by an audience on average and much less user produce comments (ratings). And the situation will be more serious with the increment of historical data. Data sparsity gives rise to severe reduction in accuracy and the high computational cost of CF-based methods. Cold start means it is difficult to predict the preference of new users who do not have any item records.

Researchers have proposed several advanced CF methods to overcome the above limitations and improve the performance of recommendation system. One class of wide range of concerns is taking advantage of the clustering or dimension reduction to eliminate the effect of sparsity of the historical user-item-rating data. Typical representative of this type methods are the bi-cluster algorithms, singular value decomposition, nonnegative matrix factorization, etc. [9–16], and the key idea of those methods is utilizing the local dense and low-dimension modules of rating matrix instead of the original sparse user-item-rating data to estimate the similarities of new user and historical users, and then give rise to recommendation by the improved similarity. Another strategy is utilizing some advanced similarity measures to improve the perception capability to the sparse data and complex information. The traditional similarity measures in CF are either the Pearson correlation or the cosine correlation. In order to consider the high-weight property of the shared common items within the K -nearest historical neighborhoods of the new user, the sigmoid function-based similarity measure is used to capture the contribution of shared common items. Furthermore, the cosine correlation is revised to adapt the real number of rating system [17], and some specific measures are able to integrate the generality and preference of rating, popularity, and influence of items into similarity metric to relieve the recommendation under the cold start. That is, Li et al. [18, 19] considered the gender and region as the basic components of the similarity metric, while Deng et al. [20] pre-filled the null rating terms of rating matrix before performing the recommended algorithm. In addition, some CF methods integrate the local and global similarity, user's credit level, and generalized K -nearest neighborhood into recommendation system to generalize the performance of CF [21, 22]. In recent years, the machine learning-derived CF models, such as probabilistic latent semantic analysis (PLSA), latent Dirichlet allocation (LDA), which often generate the multi-dimension probability density distribution of historical user-item-rating data to re-define the

similarity measures in a conditional probability way, are greatly developed as the high smooth and robustness for the heterogeneous training data, although suffering the high computational complexity [23, 24].

This paper presents a bi-clustering and information entropy-based collaborative filtering method (CBE-CF) to overcome the data sparsity and heterogeneity. Specifically, we take the advantage of the bi-clustering to identify the dense modules of rating matrix and then measure the similarity of new user and the dense modules based the information entropy measure, which characters both the global generalized consistency and the local maximal correlation of module ratings preference. Finally, a linear weighted combination of user-based CF with the improved similarity measure and item-based CF is used to perform the recommendation. In the experimental part, two benchmark datasets are used to evaluate the performance of our proposed method, compared to the state of the art.

2 Related method

2.1 User-based collaborative filtering

The basic idea of user-based CF is identifying the top n most similar neighbor users of the new user from the historical user set, and thus the item preference of the top n neighbor users is used to perform the recommendation for the new users; this procedure is simple and intuitive, and known as KNN. Specifically, for a given user-item-rating matrix U , user-based CF firstly transform the U to user-user similarity matrix R by a specific similarity measure S , that is $S: U \times U \rightarrow R$. The Pearson correlation coefficient and cosine correlation coefficient are the two widely used similarity measures.

$$S(u, v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)(r_{vi} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \bar{r}_u)^2} \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \bar{r}_v)^2}} \quad (1)$$

where the I_u and I_v indicate the index of common items of user u and v , while the r_{ui} and \bar{r}_u are the rating score of item i and the average rating of all common items of user u . In general, the common items should be more than 50 to avoid the high correlation effect derived by the few common items between the users of historical data.

Similarly, the cosine correlation can be represented as the following formula:

$$S(u, v) = \frac{r_u * r_v}{\|r_u\|_2 * \|r_v\|_2} = \frac{\sum_i r_{ui} * r_{vi}}{\sqrt{\sum_i r_{ui}^2} \sqrt{\sum_i r_{vi}^2}} \quad (2)$$

Based on the above similarity measures and the top N neighborhoods which have the highest similarity with the new user, the user-based CF can be implemented as the

weighted average of the neighborhoods u' of the new user u :

$$P_{ui} = \bar{r}_u + \frac{\sum_{u' \in N} S(u, u')(r_{u'i} - \bar{r}_{u'})}{\sum_{u' \in N} |S(u, u')|} \tag{3}$$

In practice, the selection of N have great influence on the performance of user-based CF, and $N = 20\text{--}50$ is an empirical parameter for the general application scenario.

2.2 Item based

Although the user-based CF has been widely used in various applications, the computational cost of similarity measure of users is sharply increased along with the increment of historical users; hence, the item-based CF is developed to adapt the requirement of fast response for large-scale user-item data, i.e., Amazon and Alibaba, which are the famous online shopping stores. Different with the user-based CF, the item-based CF firstly constructs the items similarity measure based on their shared users, as the number of items often is much less than the number of users in most applications; this strategy can effectively reduce the computational cost of identifying the K -nearest neighborhoods. Specifically, the item-based CF can be represented as:

$$p_{ui} = \frac{\sum_{j \in S} S(i, j)(r_{uj} - b_{ui})}{\sum_{j \in S} |S(i, j)|} + b_{ui} \tag{4}$$

where the b_{ui} is a baseline predictor.

2.3 Bi-clustering

In practice, some users often share the common preference to a specific group items, which patterns can well described by the local preference consistency among both the users(rows) and the items(columns) of the rating matrix and often are used to relieve the data sparsity in recommendation system. Unfortunately, those local patterns are explored neither by the user clustering nor by the item clustering. And we utilized the bi-clustering methods to identify the combining patterns composed of the local dense rating region for identified items with specific users. The general idea of bi-clustering is iteratively aggregating the rows and the columns of rating matrix until to convergence [25]. Specifically, for a rating matrix R , using X represents the users (rows) and Y represents the items (columns), then $I \in X$ and $J \in Y$ indicate the index subset of users and items within the same cluster. Then the clustering R_{IJ} can be derived by minimizing the average square loss function $H(I, J)$:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{iJ} + a_{IJ})^2 \tag{5}$$

where

$$a_{iJ} = \frac{1}{|J|} \sum_{j \in J} a_{ij}, \quad a_{iJ} = \frac{1}{|J|} \sum_{i \in I} a_{ij} \tag{6}$$

and

$$a_{IJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij} \tag{7}$$

here a_{iJ} , a_{iJ} , a_{IJ} represent the average value of rows, columns and cluster. The bi-clustering is a NP-hard problem and the approximate δ convergence region can be reached according to an iterative algorithm.

2.4 Information entropy

Information entropy is a measure of information distribution of random variable [26]; the large entropy means the trend of uniform distribution, and on the contrary, the small entropy indicates the sharp distribution of the random variable. Specifically, given a user i with m identified nonzero ratings for all corresponding items, the corresponding information entropy can be presented as:

$$H(x_i) = - \sum_{j=1}^m P(x_{ij}) \log P(x_{ij}) \tag{8}$$

where the x_{ij} indicates the j th specific rating value of user i , and the $P(x_{ij})$ represents the probability of user i taking the rating value according to all the item's ratings.

3 Method

Our proposed CBE-CF method can be described as the following four steps:

Step 1 A bi-clustering analysis is performed on the original user-item-rating matrix to identify its low-dimension and dense local modules. Theoretically, the ideal local clustering modules should be composed of the users with similar preference patterns and separated by the users with distinct preference patterns. In other words, users within each specific cluster have the identified item-rating patterns, and each pattern indicates specific cluster, as shown in Fig. 1.

Step 2 Calculate the information entropy for each cluster derived from step 1. In detail, we firstly count the number of items which are awarded the same rating score within the specific cluster and then estimate the probability of each rating score for the identified cluster to find out the cluster-specific information entropy, by the ratio of the

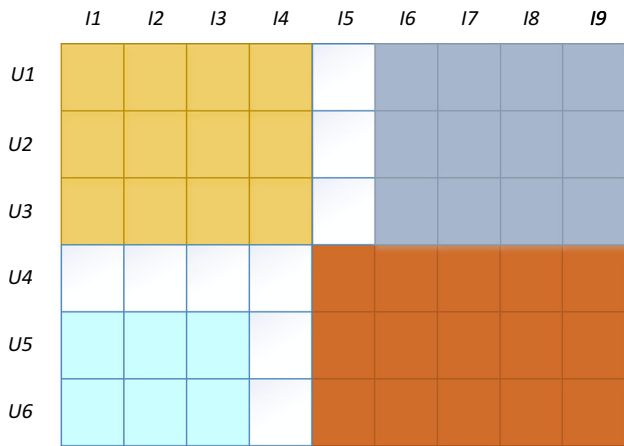


Fig. 1 An example of clustering result of user-item-rating matrix. Each color block indicate one identified cluster

number of specific rating score to the total number of items within the cluster. Then the information entropy of each cluster is calculated according to Eq. 8 and is considered as the representation patterns of each cluster. Theoretically, a new user is similar with an identified cluster when they give rise to the similar information entropy for the cluster-associated items, and hence the information entropy can be used to measure the local similarity between the new users and clusters.

Step 3 User-based CF implementation. We firstly ascending sort the $Ep_{diff_i}, i = 1, 2, \dots, C$ is the index of cluster, the difference of information entropy between all the clusters Ep_{clu_i} and the new user Ep_{new_i} with the measure $Ep_{diff_i} = |Ep_{new_i} - Ep_{clu_i}|$ and then select the top N smallest differences-associated clusters as the nearest neighborhoods to construct the recommendation system. This strategy can effectively reduce the computational cost of similarity estimation, as it just focuses on a few pre-defined clusters instead of the similarity between huge number of pairs of new users and historical users in a real-time computing way. We assume I_{new} , the items of new user can be divided into $I_k, k = 1, 2, \dots, N$ the item set associated with the top N neighbor clusters. And the similarity between a new user u_{new_k} and the cluster c_k can be defined as:

$$\text{Similarity}(u_{new_k}, c_k) = \max \frac{1}{\left(e^{|Ep_{new_k} - Ep_{clu_k}|} \left(\sum_{j=I_k^1}^{I_k^C} |r_{new}^j - r_{u_k^v}^j - b_{u_k^v}^{new}| + 1 \right) \right)} \tag{9}$$

where the IC indicates the number (capacity) of I_k , the U_k^v represents the v th user of cluster c_k , r is the rating value, and the $b_{u_k^v}^{new} = \bar{r}_{new} - \bar{r}_{u_k^v}$ indicates the base difference between average of ratings of new user and the historical

user v in cluster c_k . Finally, the recommendation for the new user u_{new_k} can be implemented by taking the weighted average of the top N nearest neighborhoods:

$$p_{u_{new_j}} = \bar{r}_{new} + \frac{\sum_{c_k \in C} \text{Similarity}(u_{new_k}, c_k)(r_{kj}^v - \bar{r}_{c_{kj}})}{\sum_{c_k \in C} |\text{Similarity}(u_{new_k}, c_k)|} \tag{10}$$

Step 4 Combinatorial collaborative filtering method (CBE-CF): the proposed bi-clustering and information entropy-based CF in step 3 primarily takes the advantage of local patterns of historical users and significantly reduces the computational cost for the large-scale training data; however, this method fails to consider the global patterns of the historical data. Hence, we present the combinatorial collaborative filtering by integrating the advantage of the bi-clustering and information entropy-based CF and the traditional item-based CF in a linear way; meanwhile, this model also maintains the low computational complexity. And this model can be described as:

$$\hat{p}_{u_{new_j}} = \lambda p_{u_{new_j}} + (1 - \lambda) p_{u_{new_j}}^{item} \tag{11}$$

Overall, the proposed method is named as CBE-CF, which is described as algorithm 1, and take $O(mn) + O(k)$ time in training phase, where the m, n, k are the user number, item number and cluster number, for the sparse matrix, the computational complexity of our method is approximated to $O(mk) + O(k)$ in training phase, and $O(1)$ in test phase.

Algorithm 1:

Parameter: cluster threshold δ , combination coefficient λ ;

While $H(I, J) > \delta$

Calculate the cluster loss: $H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iI} - a_{ij} + a_{iJ})^2$;

And $a_{iI} = \frac{1}{|I|} \sum_{j \in J} a_{ij}$, $a_{ij} = \frac{1}{|I|} \sum_{i \in I} a_{ij}$, $a_{iJ} = \frac{1}{|I||J|} \sum_{i \in I, j \in J} a_{ij}$

End;

Get the cluster set $C = [c_1, \dots, c_K]$;

For $k=1:K$

Count the number of items which are awarded the same rating score within c_k ;

Estimate the probability $P(c_k)$ of each rating score $i = 1, \dots, m$ for c_k ;

Calculate entropy EP_{c_k} of each cluster c_k by: $EP_{c_k} = -\sum_{j=1}^m P(c_{kj}) \log P(c_{kj})$;

End;

For a given u_{new} and $k=1:K$:

Calculate the $Ep_{diff_i} = |Ep_{new_i} - EP_{c_k}|$;

Calculate the similarity between u_{new} and c_k by:

$\text{Similarity}(u_{new_i}, c_k) = \frac{1}{\left(e^{|Ep_{new_i} - Ep_{clu_i}|} \left(\sum_{j=I_k^1}^{I_k^C} |r_{new}^j - r_{u_k^v}^j - b_{u_k^v}^{new}| + 1 \right) \right)}$

Calculate the recommendation score by:

$p_{u_{new_j}} = \bar{r}_{new} + \frac{\sum_{c_k \in C} \text{Similarity}(u_{new_i}, c_k)(r_{c_k^j} - \bar{r}_{c_{kj}})}{\sum_{c_k \in C} |\text{Similarity}(u_{new_i}, c_k)|}$

End;

The prediction score is calculated by:

$$\widehat{p}_{u_{new}j} = \lambda P_{u_{new}j} + (1 - \lambda) p_{u_{new}j}^{\text{item}}.$$

4 Experiment design

4.1 Data set and evaluation criteria

We construct the experiments on two benchmark dataset: One is MovieLens (HML), which was constructed by the GroupLens lab of the university of Minnesota and included 3952 users' votes on 6040 movies [27]. The other data set NF contains ratings of anonymous Netflix customers on movies. Although there are 480,189 users in NF, we employed 10,000 users sampled from differing density ranges. Each of the users in both two datasets has 20 rating scores at least, which guards the existence of the user-based similarity.

We utilize the mean absolute error (MAE) and root-mean-squared error (RMSE) to measure the precision of our CBE-CF method. Assume the predicted rating of user u_i , $i = 1, 2, \dots, n$ to film m_j , $j = 1, 2, \dots, m$ is p_{ij} , and the corresponding real rating in test dataset is r_{ij} , then the MAE can be represented as:

$$\text{MAE} = \frac{\sum_{i=1}^n \sum_{j=1}^m |p_{ij} - r_{ij}|}{nm} \quad (12)$$

And the RMSE is given in Eq. 13:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^n \sum_{j=1}^m (p_{ij} - r_{ij})^2}{nm}} \quad (13)$$

In addition, we record the computation cost of our CBE-CF method to explain how the time complexity is used and apply the t test to present the statistical significance of the results.

4.2 Experimentation methodology

In the experiments, we take the advantage of 10-fold cross-validation to test the performance of our method and the compared methods, thus, each of the two datasets is divided into 10 sub-datasets uniformly and in turn select the content of nine sub-datasets as training set and the rest one dataset as test set. The state-of-the-art, user-based CF, item-based CF, INMF, PLSA and NMF, are used to evaluate the advantages and the shortcoming of our CBE-CF method. In addition, the number of nearest neighborhoods is set to 50 for all the KNN-based DF. All experiments are run on the 1.7 version of JDK platform with an Intel i7 core processor and 8G RAM.

5 Result

5.1 The influence of cluster number

We take the threshold $\delta = 0.01\text{--}0.1$ as the range of bi-clustering convergence rate with an increment 0.01 to change the number of clusters, and then evaluate the performance of CBE-CF on the two test datasets by the MAE and RMSE. And the result is given in Table 1.

As we expect, the number of clusters is increased along with the reduction of δ ; furthermore, we can find the precision measures, both the MAE and the RMSE present a tendency of first decreasing and then increasing for the two test datasets, the minimal value is reached at $\delta = 0.09$ or 0.08, this observation also suggests the cluster number have a great influence on the result of recommendation system, the possible reason is that it is hard to capture the representative patterns for few large-size clusters derived from small δ and the global patterns will be broken on a large number of small-size local clusters derived from large δ . In the following analysis, we use the $\delta = 0.08$ as the optimal parameter.

5.2 The influence of nearest neighbor clusters and combinatorial coefficient

The performance of CBE-CF mainly depends on several pre-defined parameters: N is the number of selected nearest neighbor clusters; λ is the combinatorial coefficient of information entropy-based CF (the first term in Eq. 11) and K is the number of nearest neighborhoods of item-based CF (the second term in Eq. 11). In order to quantify the effect of parameter selection, we take the $N = 10\text{--}130$, with an increment of 10 for NF dataset, and the $N = 10\text{--}75$ with an increment of 5 for NML dataset, while set the $\lambda = 0\text{--}1$ with

Table 1 The precision of CBE-CF versus the number of clusters

Parameter	NF			HML		
	# cluster	MAE	RMSE	# cluster	MAE	RMSE
0.1	102	0.844	0.832	65	0.725	0.693
0.09	120	0.842	0.827	69	0.728	0.691
0.08	131	0.846	0.835	72	0.736	0.691
0.07	175	0.852	0.84	79	0.744	0.728
0.06	202	0.888	0.863	80	0.774	0.756
0.05	217	0.915	0.879	82	0.779	0.764
0.04	248	0.95	0.924	83	0.784	0.771
0.03	253	0.959	0.931	85	0.821	0.802
0.02	269	0.984	0.943	86	0.83	0.814
0.01	301	0.987	0.943	89	0.845	0.829

an increment of 0.1 in order, and $K = 50$, to estimate the optimal parameters by measuring the MAE and the RMSE. Specifically, Fig. 2a, b presents the trend of MAE and RMSE of the two test datasets along with the changing of parameter N , and Fig. 3c, d indicates the tendency of MAE and RMSE versus the changing of parameter λ .

Figure 2 indicates the optimal cluster number is reached at $N = 50$ for the two test datasets, while the optimal combinatorial coefficient $\lambda = 0.8$. Based on this observation, the optimal number of nearest neighborhoods suggesting a relative small value will contribute to the improvement in the accuracy of recommendation system and the reduction of the computational cost. The optimal λ with a relatively large value means the optimal structure of CBE-CF is composed of the high-weight component of the information entropy-based user CF and the low-weight component of item-based CF.

In order to explore the contribution of our proposed two components, the bi-clustering and the information entropy, to the performance of CBE-CF method, a density-based clustering method, DBSCAN, and a grid-based clustering

method, STING, are utilized to replace the bi-clustering of CBE-CF, and then the prediction accuracy of the three distinct clustering method-based CF is evaluated on the HML and NF dataset, by adopting an varying number of clusters (Fig. 3a, b). Obviously, the bi-clustering-derived CBE-CF has the predominant preference contrast to the two other methods on tested dataset. In addition, we highlight the contribution of information entropy on the aspect of capturing the local similarity by utilizing the Euclidean distance instead of the information entropy (named CB-CF) (Fig. 3c). All those comparisons validate the excellent performance of our CBE-CF.

5.3 Performance evaluation of CBE-CF

In this section, we evaluate the performance of our CBE-CF method and four other compared methods by comparing the precision and computational cost on the HML and NF dataset. We run the CBE-CF method based on the optimal parameters, and the result is given in Table 2 and Fig. 4.

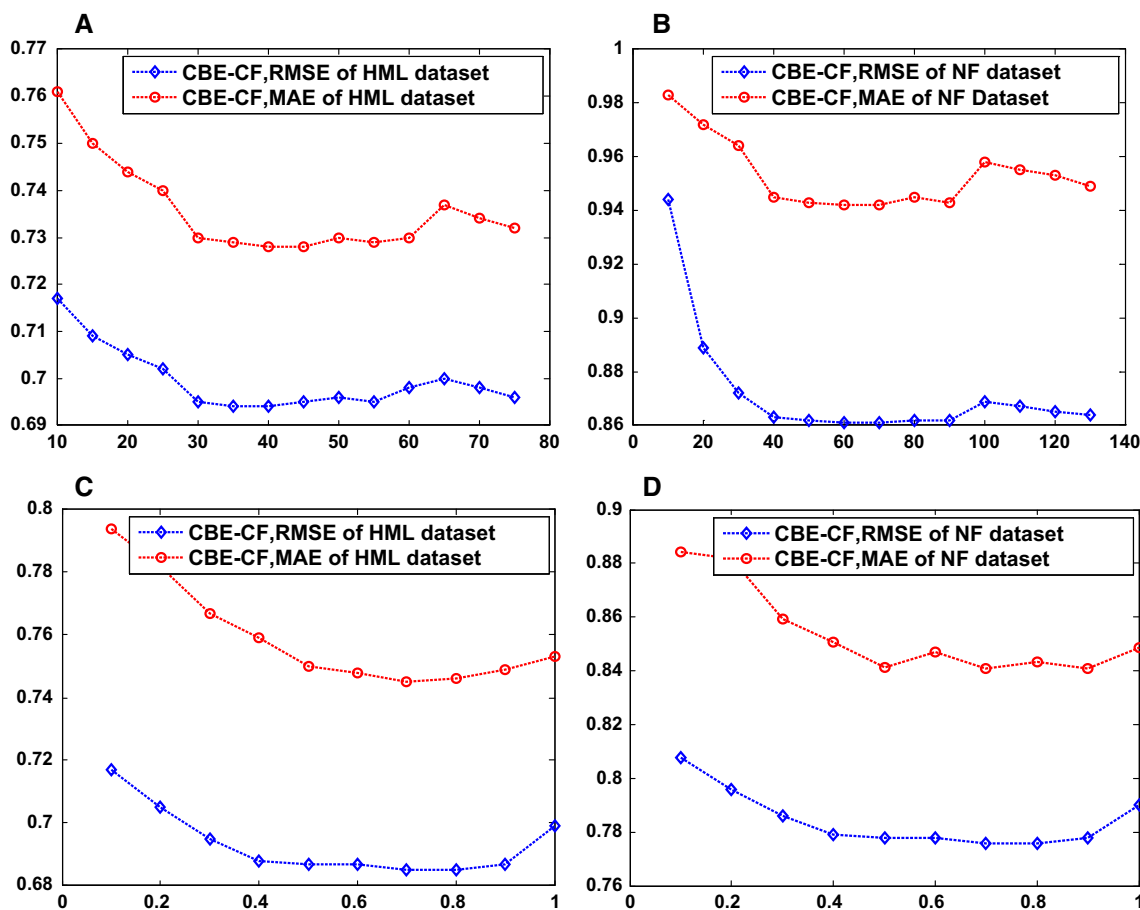


Fig. 2 The parameters estimation of CBE-CF. **a, b** The prediction accuracy of CBE-CF change along with the number of selected clusters which are traded as nearest neighborhoods for HML and NF

dataset, respectively. **c, d** The prediction accuracy of CBE-CF change along with the combinatorial coefficient λ for HML and NF dataset, respectively

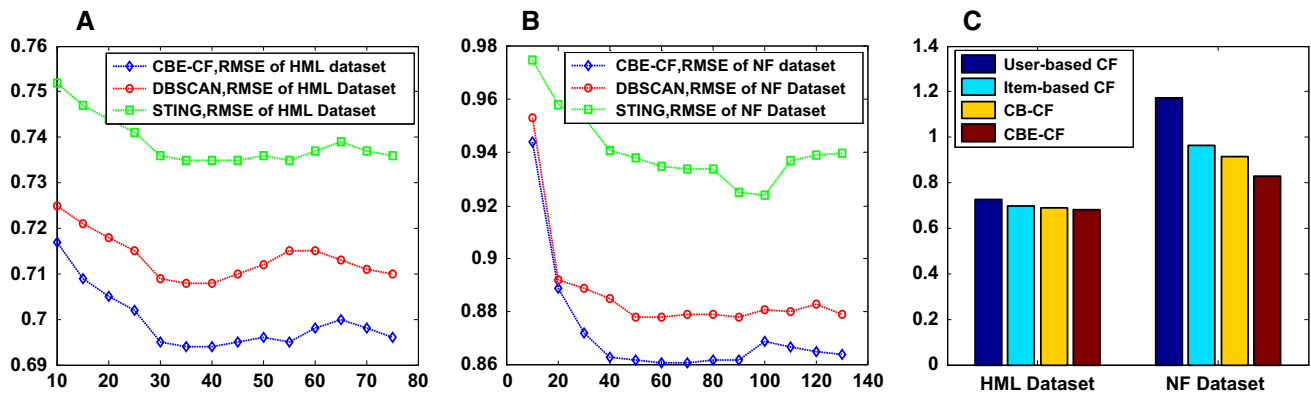
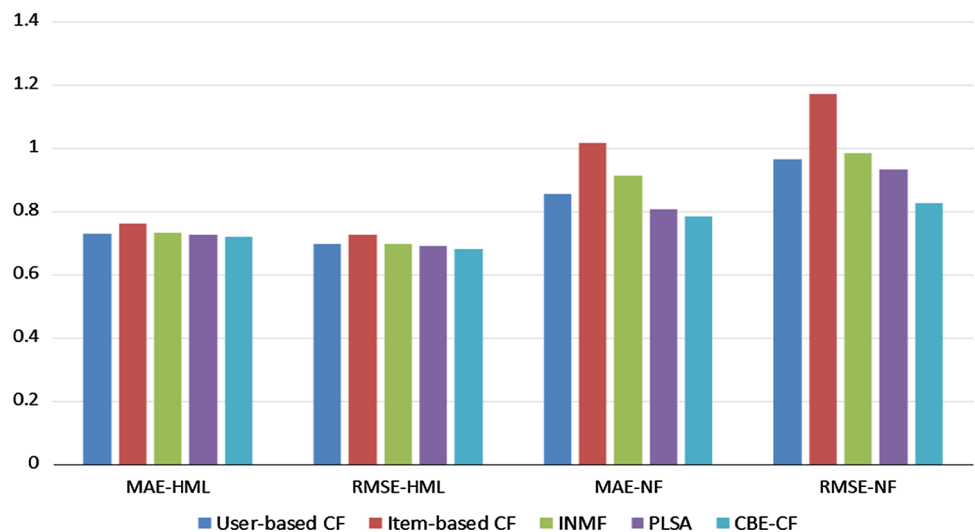


Fig. 3 The distribution evaluation of bi-clustering and information entropy. **a** The comparison of performance among three distinct clustering methods (bi-clustering CBE-CF, DBSCAN and STING)-based CF on HML dataset. **b** The comparison of performance among three distinct clustering method-based CF on NF dataset. **c** The comparison of performance among the information entropy-based (CBE-CF) and the Euclidean distance-based local similarity measures (CB-CF)

Table 2 The statistic of prediction precision and computational cost of 5 CF methods on two test dataset

Dataset	Variable	User-based CF	Item-based CF	INMF	PLSA	CBE-CF
HML	MAE	0.734	0.762	0.731	0.726	0.721
	RMSE	0.699	0.725	0.695	0.691	0.682
	P value	0.2E−5	0.1E−6	0.5E−5	0.7E−4	0.7E−6
	Time(s)	131	9.7	187	271	157
NF	MAE	0.857	1.017	0.914	0.807	0.783
	RMSE	0.964	1.172	0.985	0.932	0.827
	P value	0.8E−6	0.7E−6	0.4E−5	0.2E−6	0.3E−5
	Time(s)	3507	1025	3744	8428	2518

Fig. 4 The comparison of predicted accuracy of our CBE-CF methods against the 4 compared methods on HML and NF datasets. The accuracy is measured by the MAE and RMSE



According to Table 2 and Fig. 4, we can find that our method has the highest predicted accuracy and relatively low computational cost when compared to all five presented methods. In particular, the performance of our method is better than the probability model (PLSA) and the nonnegative matrix factorization model (INMF) with a relatively small cost. An obvious observation is that the

computational time of user-based CF fast increase along with the increment of the training data, while our CBE-CF method is not sensitive to the size of training data.

In order to validate the capability of dealing with the data sparsity of our CBE-CF, we split the NF dataset into 10 different-scale sub-datasets in a random way (such as each sub-dataset contains 1000, 2000, ..., 10,000 users in

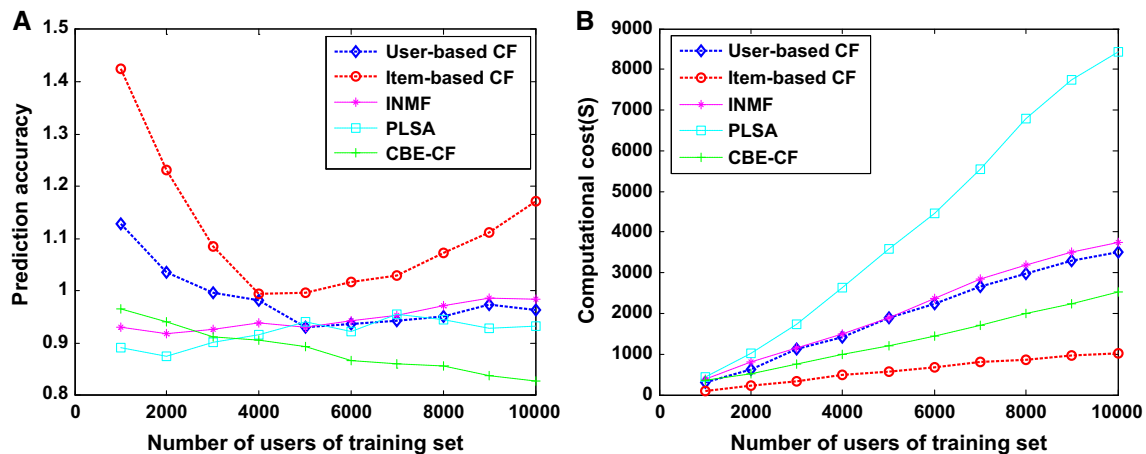


Fig. 5 The performance analysis of CBE-CF on the different-scale training set. **a** The comparison of accuracy measured by RMSE of the 5 compared algorithms versus the changing-size training set. **b** The

comparison of computational cost of the 5 compared algorithms versus the changing-size training set

order), and then we perform all the 5 CF methods on those different-scale sub-datasets and identify the tendency of the predicted accuracy and the computational cost along with the increment size of the training set. Then the result is presented in Fig. 5. Figure 5a shows the strong robustness of INMF, PLSA and CBE-CF, which maintain low RMSE against the changing size of training set. Interestingly, our CBE-CF method presents the improved accuracy along with the expanded training set, which indicates that our method can overcome the influence of sparse training data. However, except our CBE-CF, the other two robust methods (INMF and PLSA) are presented the high computational cost for the large training set.

6 Conclusion

In this paper, we propose a new collaborative filtering method: CBE-CF, which take the advantage of bi-clustering and information entropy to overcome original problems of CF methods, especially the data sparsity and high computational cost for the big training set. We replace the complex global similarity measure of the traditional user-based CF with the representation patterns (based on information entropy) of the local dense clusters. The feasibility of this strategy is validated on two benchmark datasets according to the four compared methods. It is noteworthy that the deep-learning-based CF methods also give rise to the excellent prediction capability although they are suffered from high computational cost and large size of training set [28–30]. Overall, we present a new CF method to improve the accuracy and the computational efficiency of recommendation system.

Acknowledgements This work was supported by the National Natural Science Foundation of China (61672301, 61662057), Program for Young Talents of Science and Technology in Universities of Inner Mongolia Autonomous Region (NJYT-19-B18), Industry Innovation Talent Team of Inner Mongolia Grassland Talent Engineering (2017), the Science and Technology Innovation Guide Project of Inner Mongolia Autonomous Region (2016, 2017), Research on the Information of Mongolian Medicine Based on Machine Learning Algorithm (MDXK004), Opening Foundation of Inner Mongolia Autonomous Region Engineering Technology Research Center of personalized medicine (MDK2018010), the science research project of Inner Mongolia University for the Nationalities (NMDYB17136).

Compliance with ethical standards

Conflict of interest No conflicts of interests.

References

- Zhou Y, Wilkinson D, Schreiber R, Pan R (2008) Large-scale parallel collaborative filtering for the Netflix prize. In: Fleischer R, Xu J (eds) *Algorithmic aspects in information and management*. Springer, Berlin, pp 337–348. https://doi.org/10.1007/978-3-540-68880-8_32
- Ekstrand MD, Riedl JT, Konstan JA (2011) Collaborative filtering recommender systems. *Found Trends Hum-Comput Interact* 4(2):81–173. <https://doi.org/10.1561/1100000009>
- Barragáns-Martínez AB, Costa-Montenegro E, Burguillo JC, Rey-López M, Mikic-Fonte FA, Peleteiro A (2010) A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Inf Sci* 180(22):4290–4311. <https://doi.org/10.1016/j.ins.2010.07.024>
- Breese JS, Heckerman D, Kadie C (2013) Empirical analysis of predictive algorithms for collaborative filtering. *CoRR* abs/1301.7363. arXiv: 1301.7363
- Sarwar B, Karypis G, Konstan J, Riedl J (2001) Item-based collaborative filtering recommendation algorithms. In: *Proceedings of the 10th international conference on World Wide Web*, pp 285–295. <https://doi.org/10.1145/371920.372071>

6. Xue G-R, Lin C, Yang Q, Xi W, Zeng H-J, Yu Y, Chen Z (2005) 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. <https://doi.org/10.1145/1076034.1076056>
7. Schein AI, Popescul A, Ungar LH, Pennock DM (2002) Methods and metrics for cold-start recommendations. In: Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval, pp 253–260. <https://doi.org/10.1145/564376.564421>
8. Ahn HJ (2008) A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Inf Sci* 178(1):37–51. <https://doi.org/10.1016/j.ins.2007.07.024>
9. Castro PADd, Franca FOd, Ferreira HM, Zuben FJV (2007) Evaluating the performance of a biclustering algorithm applied to collaborative filtering—a comparative analysis. In: Proceedings of the 7th international conference on hybrid intelligent systems, pp 65–70. <https://doi.org/10.1109/his.2007.55>
10. Symeonidis P, Nanopoulos A, Papadopoulos A, Manolopoulos Y (2007) Nearest-biclusters collaborative filtering with constant values. In: Proceedings of the 8th knowledge discovery on the web international conference on advances in web mining and web usage analysis, pp 36–55. https://doi.org/10.1007/978-3-540-77485-3_3
11. Paterek A (2007) Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD cup and workshop, pp 5–8. <https://doi.org/10.1145/1557019.1557072>
12. Zhang S, Wang W, Ford J, Makedon F, Pearlman J (2005) Using singular value decomposition approximation for collaborative filtering. In: Proceedings of the seventh IEEE international conference on E-commerce technology, pp 257–264. <https://doi.org/10.1109/iceect.2005.102>
13. Barragáns-Martínez AB, Costa-Montenegro E, Burguillo JC, Rey-López M, Mikic-Fonte FA, Peleteiro A (2010) A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Inf Sci* 180(22):4290–4311. <https://doi.org/10.1016/j.ins.2010.07.024>
14. Zhang S, Wang W, Ford J, Makedon F (2006) Learning from incomplete ratings using non-negative matrix factorization. In: Proceedings of the 2006 SIAM international conference on data mining, pp 549–553. <https://doi.org/10.1137/1.9781611972764.58>
15. Luo X, Zhou M, Xia Y, Zhu Q (2014) An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Trans Ind Inf* 10(2):1273–1284. <https://doi.org/10.1109/TII.2014.2308433>
16. Hernando A, Bobadilla J, Ortega F (2016) A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowl-Based Syst* 97:188–202. <https://doi.org/10.1016/j.knosys.2015.12.018>
17. Su X, Khoshgoftaar TM (2009) A survey of collaborative filtering techniques. *Adv Artif Intell*. <https://doi.org/10.1155/2009/421425>
18. Li G, Li L (2012) A new algorithm for solving cold start problem of collaborative filtering system. *J Shandong Univ (Engineering Edition)* 42(02):11–17
19. Zhang ZK, Liu C, Zhang YC, Zhou T (2010) Solving the cold-start problem in recommender systems with social tags. *EPL (Europhys Lett)* 92(2):28002–28007. <https://doi.org/10.1209/0295-5075/92/28002>
20. Deng XC, Jin C, Han QP (2013) Improved collaborative filtering modle based on context clustering and user ranking. *Syst Eng Theory Pract* 33:2945–2953
21. Jeong B, Lee J, Cho H (2009) User credit-based collaborative filtering. *Expert Syst Appl* 36(3):7309–7312. <https://doi.org/10.1016/j.eswa.2008.09.034>
22. Anand D, Bharadwaj KK (2011) Utilizing various sparsity measures for enhancing accuracy of collaborative recommender systems based on local and global similarities. *Expert Syst Appl* 38(5):5101–5109. <https://doi.org/10.1016/j.eswa.2008.09.034>
23. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. *J Mach Learn Res* 3:993–1022. <https://doi.org/10.1162/jmlr.2003.3.4-5.993>
24. Hofmann T (2004) Latent semantic models for collaborative filtering. *ACM Trans Inf Syst (TOIS)* 22(1):89–115. <https://doi.org/10.1145/963770.963774>
25. Cheng Y, Church GM (2000) Biclustering of expression data. In: Proceedings of the eighth international conference on intelligent systems for molecular biology, pp 93–103
26. Campbell J (1982) *Grammatical man: information, entropy, language, and life*. Simon and Schuster, New York
27. Kaleli C (2014) An entropy-based neighbor selection approach for collaborative filtering. *Knowl-Based Syst* 56:273–280. <https://doi.org/10.1016/j.knosys.2013.11.020>
28. Karatzoglou A, Balázs H (2017) Deep learning for recommender systems. In: Proceedings of the eleventh ACM conference on recommender systems, pp 396–397. <https://doi.org/10.1145/3109859.3109933>
29. Covington P, Adams J, Sargin E (2016) Deep neural networks for YouTube recommendations. In: Proceedings of the 10th ACM conference on recommender systems, pp 191–198. <https://doi.org/10.1145/2959100.2959190>
30. Wang H, Wang N, Yeung DY (2015) Collaborative deep learning for recommender systems. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining, pp 1235–1244. <https://doi.org/10.1145/2783258.2783273>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.