

A Hybrid Recommendation Model Based on Estimation of Distribution Algorithms

Tianyi LIANG*, Yongquan LIANG, Jiancong FAN, Jianli ZHAO

*School of Information Science and Engineering, Shandong University of Science and Technology,
Qingdao 266510, China*

Abstract

To overcome the fundamental problems of collaborative filtering: sparsity and cold-start, in this paper, a novel hybrid recommendation model which takes advantages of collaborative filtering and content-based mechanism is proposed. The model employs Estimation of Distribution Algorithms to learn the preferences of users and combines them into user interest profiles which are used to accurately describe users' interest features. With the profiles, the content-based mechanism in the model is able to recommend new items to users, and the collaborative filtering suffers less from the sparsity problem because the similar users are determined by the profiles rather than the user-item matrix. Empirical studies on MovieLens data set prove the validity of the proposed model.

Keywords: Collaborative Filtering; Sparsity; Cold-start; Estimation of Distribution Algorithms; Hybrid Recommendation

1 Introduction

Recommender system is a powerful tool for addressing information overload problem [1], it provides personalized service for us according to our demand features and interest features. In recent years, recommender system attracted great attention from industry and academia because it has shown huge application potential in many fields.

In a variety of recommendation technologies, collaborative filtering (CF) [1, 4] is the most successful one in the past years. The mechanism of classic collaborative filtering is very simple: for user u , finding a group of users which are called “nearest-neighbors”, and recommend their favorite items to user u . The term “nearest-neighbors” represents a type of users who are similar with the target user. In the classic CF (or called User-based CF [1, 4]), if two users are “similar”, that means their ratings or preferences are similar. Compared with other recommendation technologies, such as content-based [1, 2] and ruled-based recommendation which must work with

*Project supported by the National Nature Science Foundation of China (No. 61203305), 973 Program (No. 2012CB724106), Natural Science Foundation of Shandong Province (No. ZR2010FQ021, ZR2012FM003).

*Corresponding author.

Email address: liangtee@126.com (Tianyi LIANG).

specific type items, CF can be applied to any form of items. Additionally, CF has the ability to explore users' new interests. This feature makes CF can continually recommend novel and exciting items to users, rather than always recommend similar ones, like content-based recommender does.

However, some challenges still come along with CF. Firstly, CF's performance will decline badly if the user-item matrix is sparse. To solve this problem, many approaches have been proposed. Item-based CF [1, 3, 4], another classic Nearest Neighbor-based CF (User-based CF and Item-based CF are often called Nearest Neighbor-based CF because they generate recommendations based on Nearest Neighbors. They are also called Memory-based CF), calculates the similarity according to items instead of users. It performs better than User-based CF, especially when the rating data is sparse, but its scalability is limited with the large and increasing number of item data.

Model-based CF, which tries to improve CF's performance by using some data mining or machine learning techniques, has been researched for many years. Among various of Model-based CF, Dimensionality Reduction-based CF, especially SVD-based CF [6, 7, 17] and its variations [6, 16], became research focus in recent years because of their good performance in some famous contests such as Netflix Prize and KDD cup. The philosophy of Dimensionality Reduction-based is using some techniques, for example, Singular Value Decomposition [6, 7, 17] and PCA [4], to extract the feature values from the original user-item matrix and then build a smaller matrix with the feature values ("smaller" means lower dimensionality). The final recommendation is generated based on the new user-item matrix. This kind of algorithm performs very well on sparse data, but its some drawbacks are difficult to overcome: Dimensionality Reduction will cause information-loss that may decrease the recommendation accuracy. And like most of Model-based CF, the model training process will cost a lot of time. Other typical Model-based CF includes Latent-Semantic based CF, MDP-based CF, Regression-based and Clustering-based CF etc [1, 4, 8]. The Model-based CF usually has some attractive features and can alleviate the sparsity problem in some way, but they cannot tackle another severe challenge for CF, that is the cold-start problem.

Cold-start problem refers to the system cannot generate any recommendations about new items because there are no ratings on them. And for new users, because they do not have any ratings stored in the system, therefore they cannot receive any recommendations from the system. Some recommender systems, such as Fab [9], combine content-based and CF to address the cold-start and sparsity problems, their solutions are called hybrid recommendation (HR) [1, 4, 9, 10, 14, 15]. Hybrid recommendation usually has better performance than single algorithm, hence it is a brighter way to deal with the challenges for CF.

1.1 Proposed approach

The content of the rated items factually reflect users' preferences, thus, it is very meaningful to mine the users' interest features in the content by using some machine learning techniques.

Inspired by above idea, we develop a novel recommendation model that mines the users' interest features by Estimation of Distribution Algorithms (EDAs) [11, 12, 13] and combines these features into user interest profiles. In the model, users are represented by their profiles, rather than the user-item matrix. Compared with the user-item matrix, the advantages of the user interest profile are, firstly, it can describe user's interest features more precisely because it is refined from the

content of rated items. Furthermore, the process of building user interest profiles is less sensitive to data sparsity, this feature will help us to solve the sparsity problem.

At the same time, the work of building user interest profiles can help us integrate the content-based mechanism into our solution seamlessly, therefore we get a hybrid recommendation model. In this model, the content-based recommendation mechanism will recommend the new item to users if the item's features are similar with user interest profiles.

The advantages of our approach has been proven by the experiments in section 4. Therefore, the main contribution of this paper is to present a novel hybrid recommendation model that is able to solve the new item cold-start problem and performs very well on sparse data.

2 Introduction to EDAs

Estimation of Distribution Algorithms (EDAs) derives from Genetic Algorithm (GA) but is different from it. There is no crossover or mutation operating in EDAs, they are replaced by modeling and random sampling (shown in Fig. 1). The search strategy of EDAs is establishing probability model for the entire population and random sampling according to the model. In EDAs, since the superior individual is being assigned to higher probability than normal and inferior one, therefore the superior individual is more likely to be selected as the member of new population, and this will boost the population evolution.

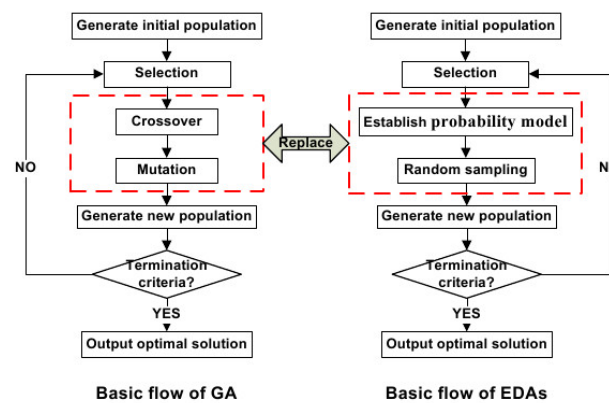


Fig. 1: Comparison of GA and EDAs

Compared with GA, EDAs has more powerful global search capacity and faster convergence rate, thus, it is a more powerful tool for solving hard optimization problem.

3 EDAs-based Hybrid Recommendation

As is described in section 1.1, the core of our approach is using EDAs to learn user interest profiles hence our first task is implementing this idea. The implementation details of task 1 is introduced in section 3.1. Once the profiles learnt, the following tasks are using it to generate CF recommendations and new item recommendations. Implementation details of these two tasks are introduced in section 3.2 and section 3.3.

3.1 Learning user interest profile by EDAs

3.1.1 Definitions

Denote the rating on item n made by user u as $r_{u,i}$. The item n is represented by a weighted-term vector $t_n = ((k_1, w_1), (k_2, w_2), \dots, (k_n, w_n))$. The keywords are extracted from the text information of item and the weights are calculated by TF-IDF [2].

Define S_u is a item set and its members are “not disgusted” by user u : $S_u = \{(t_1, r_{u,1}), (t_2, r_{u,2}), \dots, (t_n, r_{u,n}) | r_{u,i} > \bar{r}_u\}$. All of the keywords in S_u are put into set D_u .

The user interest profile of user u is to describe his preferences on different keywords, it is also represented by a weighted-term vector. Let function (1) be the fitness function, hence the goal of EDAs is to learn a weighted-term vector $profile_u$ to minimize the fitness function.

$$Fitness(profile_u) = \sum_{i \in S_u} \log(r_{u,i} \times sim(profile_u, t_i)), \quad (1)$$

where

$$sim(profile_u, t_i) = \cos(profile_u, t_i) = \frac{profile_u \cdot t_i}{\|profile_u\| \times \|t_i\|}$$

3.1.2 Procedures of EDAs learning

Step 1: Randomly select n keywords from D_u and assign value 0.1, 0.3, 0.5, 0.7, 0.9 to each keyword as weight, put these weighted keywords into set K_u :

$$K_u = \left\{ \begin{array}{cccc} (k_1, 0.1), & (k_1, 0.3), & \dots & (k_1, 0.9) \\ \dots & \dots & \dots & \dots \\ (k_n, 0.1), & (k_n, 0.3), & \dots & (k_n, 0.9) \end{array} \right\}$$

Step 2: Generate initial population B_0 by Monte Carlo method: $B_0 = \{profile_1, profile_2, \dots, profile_n\}$. Here $profile_n = ((k_i, w_i), (k_j, w_j), \dots, (k_k, w_k))$ and each gene (k_i, w_i) is random sampled from K_u according to probability model p : $p = (((k_1, 0.1), c_{1,1}), \dots, ((k_1, 0.9), c_{1,5}), \dots, ((k_n, 0.9), c_{n,5}))$. In the probability model p , $c_{n,i}$ represents the probability of gene (k_n, w_{ni}) (e.g., $((k_1, 0.3), 0.45)$ represents the probability of gene $(k_1, 0.3)$ is 0.45). At the beginning, all the $c_{n,i} = 1/N$. N is the total number of items in set K_u .

Step 3: If the evolutionary generation is less than **MAXGEN**:

(1) Compute the fitness value of each $profile_u$ by function (1) and select the top-M individuals (ranked by fitness value and $M < N$) to update the probability model p . For a $((k_n, w_{n,i}), c_{n,i})$ in p , it will be updated by function (2):

$$((k_n, w_{n,i}), c_{n,i}) = P(((k_n, w_{n,i}), c_{n,i}) | X_S), \quad (2)$$

where $P(((k_n, w_{n,i}), c_{n,i}) | X_S) = \frac{\sum_{j=1}^M \delta_j((k_n, w_{n,i}) | X_S)}{M}$.

X_S is a set that consists of the top-M individuals, $\sum_{j=1}^M \delta_j((k_n, w_{n,i}) | X_S)$ computes the total number of $(k_n, w_{n,i})$ in X_S . The result of the computation will be assigned to $c_{n,i}$.

(2) Generate N new individuals by random sampling according to the new probability model p , which is updated in previous step (1). Go to **Step 3**.

Step 4: Return the optimal individual as $profile_u$.

3.2 Collaborative filtering based on user interest profile

Compared with the classic CF, the most significant improvement in our approach is the similarity between users is determined by their user interest profiles, instead of the user-item matrix. The similarity is calculated by weighted pearson [3]:

$$sim(u_i, u_j) = \begin{cases} S_{pearson}(u_i, u_j) \cdot \frac{|I_i \cap I_j|}{10} & |I_i \cap I_j| < 10 \\ S_{pearson}(u_i, u_j) & \text{otherwise} \end{cases} \quad (3)$$

$$S_{pearson}(u_i, u_j) = \frac{\sum_{c \in I_i \cap I_j} (w_{i,c} - \bar{w}_i)(w_{j,c} - \bar{w}_j)}{\sqrt{\sum_{c \in I_i \cap I_j} (w_{i,c} - \bar{w}_i)^2} \sqrt{\sum_{c \in I_i \cap I_j} (w_{j,c} - \bar{w}_j)^2}}$$

Here I_i and I_j are keyword sets of u_i and u_j . $w_{i,c}$ denotes the weight of keyword c in the user interest profile of u_i , \bar{w}_i denotes the mean weight of user u_i . Weighted pearson is developed for solving the **overlapping problem** of classic Pearson, 10 is the best parameter for our approach (In practical we set each user interest profile is consist of 30 weighted keywords).

Predictions are computed as the weighted mean deviations from the neighbors' mean:

$$PR_{i,n} = \bar{r}_i + \frac{\sum_{u_j \in Neighbors_i} sim(u_i, u_j)(r_{j,n} - \bar{r}_j)}{\sum_{u_j \in Neighbors_i} |sim(u_i, u_j)|}. \quad (4)$$

3.3 Cold-start of new item

CF is unable to recommend new items to users because nobody rated it, while content-based mechanism has no such drawback. Content-based mechanism in our model computes the similarity between the new item's weighted term-vector and the user interest profile, and recommend the item if the similarity exceeds the threshold. The similarity is computed by function (3).

4 Experiments

4.1 Data set and evaluation metrics

The experiments are performed on MovieLens 1M data set provided by GroupLens lab. This data set contains 1000209 ratings from 6040 users on 3883 movies. To evaluate the algorithms, we take MAE (Mean Absolute Error), Precision, Recall and F1 measure as evaluation metrics.

Since the data set only contains little text description of movies, thus we crawl directors, writers, actors, genres and synopsis information of each movie from **IMDb**.

4.2 Methodology

According to the principles of cross-validation, the data set is split into ten subsets (for each user his ratings is split into ten parts) and each experiment is iterated for ten times. In each iteration

we randomly select N subsets as test set (e.g. $N=2$) and merge remaining ones as training set. In experiment 1 we set $N=2$, while in experiment 2 the value of N is from 9 to 1, that means the percentage of data used as training set is from 10% to 90%. The final result is the mean of ten times experiments.

4.3 Results and discussions

4.3.1 Experiment 1: Cold-start of new item

In this experiment the items in test set are treated as new items and algorithms tries to recommend them. Two typical content-based recommender algorithms: kNN [2] and Naive Bayes [2] are taken as comparison. Results are presented in Table 1.

Table 1: Experiment 1 results

Algorithm	Precision	Recall	F1	Precision(5)	Precision(10)
kNN(k=30)	0.691	0.687	0.692	0.944	0.850
Naive Bayes	0.710	0.696	0.703	0.947	0.858
EDAs(threshold=0.3)	0.736	0.732	0.734	0.955	0.873
EDAs(threshold=0.5)	0.762	0.703	0.731	0.955	0.873
EDAs(threshold=0.7)	0.772	0.686	0.726	0.955	0.873

Precision (N) (e.g. $N=5$) is the precision of top- N recommendation. The results demonstrate our model is able to recommend new items to users and its performance is better than the other two algorithms, that is largely because the user interest profile learnt by EDAs can describe user's interest features more precisely. On this data set, for our model, as the threshold rises, precision will rise but recall will decline. The reason behind this phenomenon is a higher threshold will make our model classify new items as user desired more strictly and more precisely, but fewer new items are beyond the threshold.

4.3.2 Experiment 2 : Accuracy and sparsity

We study the accuracy changes of different algorithms when the percentage of training set is changed. As shown in Fig. 2, all the algorithms will perform better accuracy as the training percentage increase, and finally their accuracy differences are not very large because the training set becomes more denser as the percentage increase.

But when the percentage is relatively low (percentage $\leq 60\%$), the data in training set is sparse, algorithms' accuracy differences are very clearly. This phenomenon is caused by the differences in the abilities of different algorithms in extracting information from ratings. SVD++ [17] performs well on sparse data mainly because it uses feature value extraction technique to learn latent information about users from ratings and exploit these information to make predictions. Our model, the only one in these can compete with SVD++, uses EDAs to learn user's preferences from the content of rated items and make prediction based on the preferences, hence it also performs very well. Other three kinds of algorithms just simply use the rating values therefore when the ratings are scarce their performances decline badly.

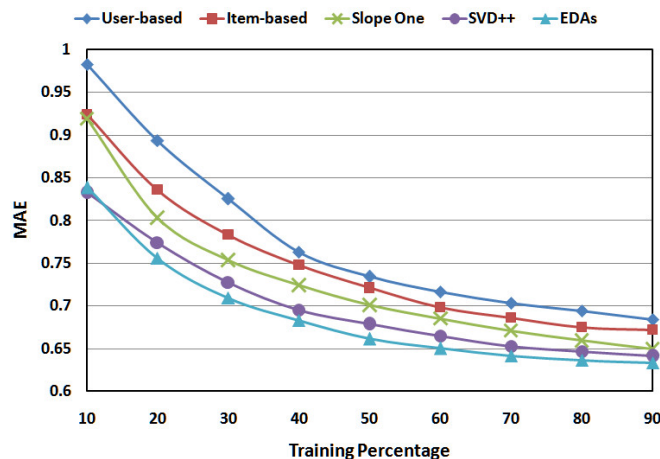


Fig. 2: MAE-training percentage graph

5 Conclusions and Future Work

In this paper, we propose a novel hybrid recommendation model, which takes advantage of collaborative filtering and content-based filtering, and employs Estimation of Distribution Algorithms to improve the user modeling. Experiments demonstrate that our EDAs-based HR can solve the cold-start of new item problem and performs well on sparse data. On this basis, improving the dynamic updating mechanism of user interest profile will be a very meaningful work in the future.

Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61203305), 973 Program (No. 2012CB724106) and Natural Science Foundation of Shandong Province (No. ZR2010FQ021, ZR2012FM003).

References

- [1] X Su, TM Khoshgoftaar, A survey of collaborative filtering techniques, *Advances in Artificial Intelligence*, Vol. 2009, pp. 1-19, 2009.
- [2] Pazzani M J, Billsus D, Content-based recommendation systems, *The adaptive web*. Springer Berlin Heidelberg, pp. 325-341, 2007.
- [3] Sarwar, Badrul, et al, Item-based collaborative filtering recommendation algorithms, *Proceedings of the 10th international conference on World Wide Web*, ACM, 2001.
- [4] Cacheda F, Carneiro V, Fernandez D, et al, Comparison of collaborative filtering algorithms: Limitations of current techniques and proposals for scalable, high-performance recommender systems, *ACM Transactions on the Web (TWEB)*, Vol. 5, pp. 2, 2012.
- [5] Daniel Lemire, Scale and translation invariant collaborative filtering systems, *Information Retrieval*, Vol. 8, pp. 129-150, 2005.
- [6] Ma, Chih-Chao, *A Guide to Singular Value Decomposition for Collaborative Filtering*, csientue-dutw, 2008.

- [7] Sarwar, Badrul, et al, Application of dimensionality reduction in recommender system-a case study, No. TR-00-043, Minnesota Univ Minneapolis Dept of Computer Science, 2000.
- [8] Lemire, Daniel, and Anna Maclachlan, Slope one predictors for online rating-based collaborative filtering, *Society for Industrial Mathematics*, Vol. 5, pp. 471-480, 2005.
- [9] Balabanovic, Marko, and Yoav Shoham, Fab: content-based, collaborative recommendation, *Communications of the ACM*, Vol. 40, pp. 66-72, 1997.
- [10] Shahabi, Cyrus, and Yi-Shin Chen, An adaptive recommendation system without explicit acquisition of user relevance feedback, *Distributed and Parallel Databases*, Vol. 14, pp. 173-192, 2003.
- [11] Zhou, Shude, and Zengqi Sun, A survey on estimation of distribution algorithms, *Acta Automatica Sinica*, Vol. 33, pp. 113, 2007.
- [12] Muhlenbein, Heinz, Jurgen Bendisch, and H-M. Voigt, From recombination of genes to the estimation of distributions II. Continuous parameters, *Parallel Problem Solving from NaturePPSN IV*, Springer Berlin Heidelberg, pp. 188-197, 1996.
- [13] Muhlenbein, Heinz, The equation for response to selection and its use for prediction, *Evolutionary Computation*, Vol. 5, pp. 303-346, 1997.
- [14] Barragns-Martnez, Ana Beln, et al, A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition, *Information Sciences*, Vol. 180, pp. 4290-4311, 2010.
- [15] de Campos, Luis M, et al, Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks, *International journal of approximate reasoning*, Vol. 51, pp. 785-799, 2010.
- [16] Ziming, Z. E. N. G. A Context-aware Personalized Commodity Recommendation for Ubiquitous Commerce Based on Click Streams and Collaborative Filtering. *Journal of Computational Information Systems*, Vol. 8 (8), pp. 3489-3496, 2012.
- [17] Koren, Yehuda, Factorization meets the neighborhood: a multifaceted collaborative filtering model, *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 426-434, 2008.