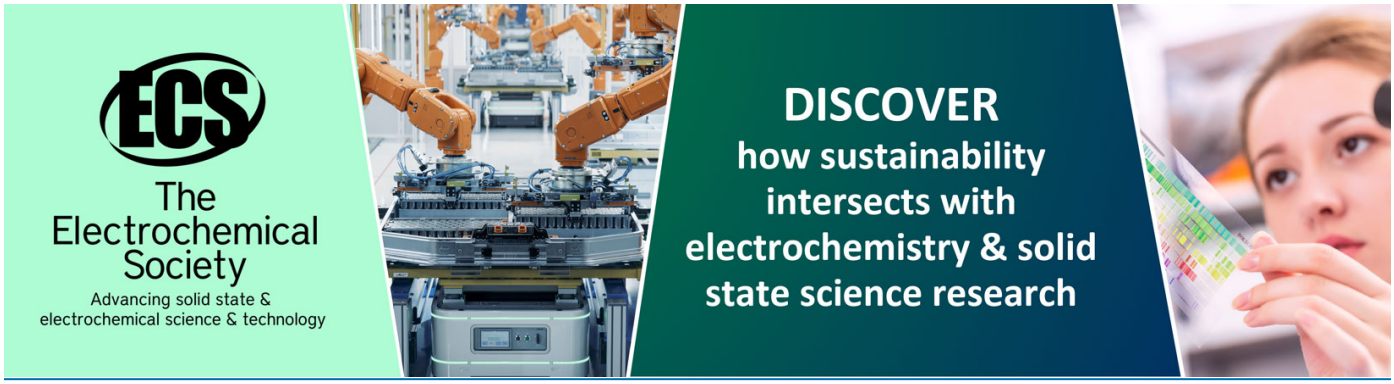
A Recommendation Algorithm Based on Item Genres Preference and GBRT



IOP Conf. Series: Journal of Physics: Conf. Series **1229** (2019) 012053 doi:10.1088/1742-6596/1229/1/012053

A Recommendation Algorithm Based on Item Genres Preference and GBRT

Yuan Wang[1, a] and Yan Tang[1, b]

[1]School of Computer and Information Science, Southwest University, Chongqing; 400715, China

[a]wangyuan6921@163.com; [b]ytang@swu.edu.cn

Abstract. The most mature and widely used collaborative filtering algorithm is facing the problem of data sparsity, which is not conducive to the acquisition of user preferences, thus affecting the recommendation effect. Introducing item genres into recommendation algorithm can reduce the impact of data sparsity on recommendation effect. The personalized preferences of users can be extracted more effectively from the preference information of item genres, and the recommendation accuracy can be further improved. On this basis, this paper proposes a recommendation algorithm based on item genres preference and GBRT, which divides similar users by K-means clustering algorithm, extracts user preferences of item genres and auxiliary features, and establishes a rating prediction model combined with GBRT algorithm. The experiments on the common datasets of Movie Lens 100K and Movie Lens 1M show that the proposed algorithm achieves 0.8%-7% optimization on the evaluation index MAE, which indicates that the impact of data sparsity is reduced to a certain extent and the recommendation efficiency is better than the existing recommendation algorithm.

1. Introduction

Collaborative filtering algorithm is the most widely used and researched recommendation technology [1], including memory-based collaborative filtering algorithm and model-based collaborative filtering algorithm. The memory-based collaborative filtering algorithm uses the implicit or explicit behaviors of users to get the item rating matrix, and then calculates the similarity between users or items for collaborative filtering prediction. However, it is difficult to effectively improve recommendation performance due to the problem of data sparsity and diversity. For improving the accuracy of recommendation, the model-based collaborative filtering algorithm predicts the user's rating on the unrated items by modeling existing rating data to make recommendations. Matrix factorization model [2-4] can reduce dimension, computational complexity and storage space, however, the factorization algorithm not only loses the original rating information, but also easily produces the phenomenon of over-fitting. The model based on clustering technology [5-7] fills the rating matrix after narrowing the similarity range by clustering users or items, optimizing the problem of sparsity but ignoring the diversity of users' interests, which makes the recommendation accuracy difficult to guarantee.

Although the above methods improves the prediction accuracy of the model and reduces the impact of data sparsity to a certain extent, it can't effectively extract user preferences of items, which limits the accuracy of the prediction. In this paper, a recommendation algorithm based on item genres preference and GBRT is proposed. Integrating item genres and auxiliary features into user rating information can effectively mitigate the impact of data sparsity. Comparison of experiment with

compared algorithms shows that the MAE of proposed algorithm is smaller and the recommendation accuracy is better.

2. GBRT

GBRT (Gradient Boosting Regression Tree) [8] is an integrated learning algorithm based on gradient boosting proposed by Friedman. It can also be referred to as MART (Multiple Additive Regression Tree) or GBDT (Gradient Boosting Decision Trees). The principle is to achieve accurate classification and regression effect through iterative calculation of weak classifiers. This algorithm was originally designed for yahoo's CTR prediction but now has been widely used in traffic passenger flow prediction, pharmaceutical prediction and many other aspects because of its high prediction accuracy, the suitability for low-dimensional data and the ability to process non-linear data and other advantages.

For regression problems, it is assumed that the input is the training sample set T, m is the total of samples, T is the maximum of iterations, L is the loss function. The core steps of GBRT algorithm are as follows:

1) Initialize $f_0(x) = \arg\min_c \sum_{i=1}^{m} L(y_i, c)$.
2) For t = 1 to T:
   a) For $i = 1, 2, \cdots, m$ compute $r_{\text{it}} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=f_{t-1}(x)}$.
   b) Fit a regression tree to targets r_{it} giving terminal regions $R_{jt}\ (j = 1, 2, \cdots, J_{\text{t}})$.
   c) For $j = 1, 2, \cdots, J_{\text{t}}$ compute $c_{jt} = \arg\min_c \sum_{x_i \in R_{jt}} \mathrm{L}(y_i, f_{t-1}(x_i) + c)$.
   d) Update $f_t(x) = f_{t-1}(x) + \sum_{j=1}^{J} c_{jt} I(x \in R_{jt})$.
   e) Output $\hat{f}(x) = f_T(x)$.

The selection of loss function of GBRT model is very important. In this paper, huber is selected as the loss function of regression model because of its better noise resistance. In addition, we prevent over-fitting by controlling the learning rate and subsample, which can effectively improve the generalization ability of the model.

3. Recommendation Algorithm Based on Item Genres Preference and GBRT

The proposed recommendation algorithm based on item genres preference and GBRT divides similar users by K-means clustering algorithm, constructing sub-datasets based on user clusters for extracting user genres preferences. Considering user rating habits, item global evaluation and user genres preferences, we extract three auxiliary features of user average rating, item average rating and project genres average rating. The GBRT regression model is used to train the samples and predict user's ratings on the non-rated items.

3.1. Extracting user genres preferences

3.1.1. Labeling User. Assuming that users or items in the same cluster have more similar characteristics, clustering method is used to divide users or items into different clusters to distinguish the nearest neighbors. For example, Kim et al. [9] utilize dynamic K-means clustering algorithm to cluster users' music lists and realizes users personalized recommendation based on clustering results. In this paper, K-means is used to cluster the user-item rating matrix R to obtain different user clusters. Supposing that R is clustered into k clusters and then each user is labeled. The labeled user-item rating matrix R' is shown in Table 1.

Table 1. Labeled user-item rating matrix R'

	i_1	i_2	i_3	⋯	i_m	Cluster
u_i	5	0	3	⋯	2	C_1
u_j	0	5	4	⋯	0	C_3
⋯	⋯	⋯	⋯	⋯	⋯	⋯
u_m	0	3	0	⋯	2	C_2

Table 2. Initial DataSet D

user	item	rating	G_1	G_2	G_3	⋯	G_t
u_i	33	4	1	0	0	⋯	1
u_j	17	3	0	1	0	⋯	0
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
u_m	289	5	0	0	1	⋯	1

3.1.2. Constructing Sub-datasets. Obviously, an item can have one or more genres attributes. For example, a movie can be labeled romantic, while the other is marked as adventure, comedy, action, and west. So the construction of item-genres matrix G is as follows.

$$G_{ij} = \begin{cases} 1, & \textit{Project i contains genre j} \\ 0, & \textit{Project i does not contain genre j} \end{cases} \tag{1}$$

The initial dataset D is formed by combining item-genres matrix G with user-item rating data, as shown in Table 2. According to the K clusters, all information of users belonging to the same cluster is divided into a sub-dataset, that is, the initial dataset D is divided into k sub-datasets $\{D_i \in D \mid i \in 1, \cdots, k\}$, The next step is to calculate user preferences for item genres within each sub-dataset D_i.

3.1.3. Building a user genres preference matrix. The frequency of each genre attribute appearing and being graded is different for the market reason. For movies, theatre, comedy and action have a higher frequency of appearing and have more rating records, while fantasy, darkness and other genres have a lower frequency of appearing and relatively lower scoring data. For Movie Lens 1M, one of the experimental datasets in this paper, the genres frequency of items is shown in Figure 1.

Considering the difference of genres frequency, the user personalized preference for genres is obtained by counting the number of genres that users have evaluated and the number of genres of sub-dataset that users belong to. In addition, the user rating for specific item also reflects the user preference on genres to a certain extent. Combining the above two preferences, we define user preference for item genres as follows:

$$preference(u, g) = \frac{1}{num(I_{ug}) + 1} \sum_{i \in I_{ug}} \frac{f_g}{f_{C,g}} * \frac{r_{ui}}{\overline{r_u}} \tag{2}$$

Where preference(u,g) represents the preference score of user u for genre g, I_{ug} represents the set of items containing the genre g evaluated by the user u, f_g represents the frequency of g in the item sets evaluated by u, $f_{C,g}$ indicates the frequency of g in the item sets evaluated by all users in the cluster C that u belongs to, r_{ui} represents the rating of the u for the i, and $\overline{r_u}$ represents the rating average value of u. Num() denotes the quantity of elements in a set.

Most rated items do not cover all genres. To alleviate the cold-start problem, we define the user preference on unrelated genres as the average of the genres preference of other users in the same cluster, as follows:

$$preference(u, g') = \frac{1}{num(C_{g'})} \sum\nolimits_{v \in C_{g'}} preference(v, g') \tag{3}$$

Where $C_{g'}$ represents the set of users with rating for the genre g' in the cluster C to which the u belongs. According to formula (2) and formula (3), the user-genres preference matrix P as shown in Table 3.

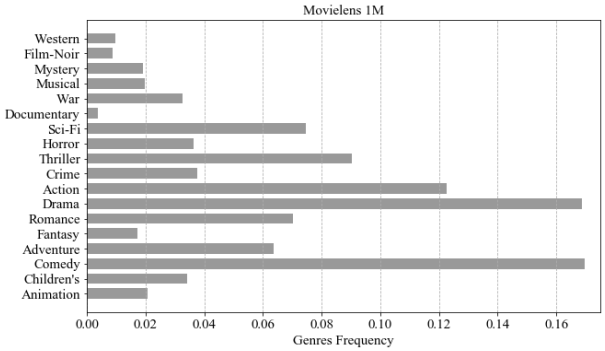


Figure 1 Item genres frequency

Table 3. User-genres preference matrix P

	G_1	G_2	G_3	⋯	G_t
u_i	1.1429	1.0315	0.3500	⋯	0.2592
u_j	0.7463	0.9888	0.5052	⋯	1.2215
⋯	⋯	⋯	⋯	⋯	⋯
u_m	1.4118	1.2215	0.0885	⋯	0.2423

3.2. Extracting auxiliary features

We know that some users are very critical and always give very low ratings, while some users are more tolerant and tend to rate highly. On the other hand, some items are of poor quality and bad reviews, while some items are very good and get high ratings. The rating may be higher when users are faced with the genres they like, but when the opposite is true, the score may be lower. Based on the above considerations, we extracted three kinds of auxiliary information, namely mean user rating, mean item rating and mean user item genres rating. User mean rating is the average value of all user ratings $\overline{r_u}$, which will not be described here. The mean item rating is defined as follows:

$$\overline{r_{C,i}} = \frac{\sum_{u \in C} r_{ui}}{num(C)} \tag{4}$$

Where $\overline{r_{C,i}}$ denotes the mean rating of item I in the sub-dataset corresponding to user cluster C.

The mean user item genres ratings is as follows:

$$\overline{r_{ug,i}} = \frac{\sum_{g \in (G_i \neq 0)} r_{ug}}{num(G_i \neq 0)} \tag{5}$$

Where $\overline{r_{ug,i}}$ denotes the mean genres rating of user u on item i, and $num(G_i \neq 0)$ indicates the number of genres of I. In the above formula, r_{ug} represents the mean rating of the all scores of user u on the genre g, which is as follows:

$$r_{ug} = \frac{\sum_{i \in (I_{ug})} r_{ui}}{num(I_{ug})} \tag{6}$$

3.3. Regression prediction based on item genres preference and GBRT

The user-genres preference matrix, three auxiliary features and the initial dataset are combined to construct the final regression training sample set, as shown in Table 4. The GBRT model is used to fit the training set. After training the model parameters, the model is used to predict the user's rating on the item.

Table 4. Regression training sample set

userId	itemId	rating	G_1	G_2	G_3	⋯	G_4	$\overline{r_u}$	$\overline{r_{C,i}}$	$\overline{r_{ug,i}}$
1	33	4	0.9544	0.0000	0.0000	⋯	0.0000	3.6036	3.4430	4.2142

25	196	5	1.1653	0.0000	0.0000	⋯	1.6227	3.4122	3.4122	3.1250
⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯	⋯
938	17	3	0.0000	1.0848	2.0723	⋯	0.0000	3.6036	4.0425	3.8200

4. Experiments

4.1. Experimental environment and dataset

The experimental environment of the algorithm is: 4 GHz processor PC, 8 GB RAM and Pycharm 3.6 used under 64-bit Microsoft Windows 10. The Movie Lens 100K and Movie Lens 1M datasets provided by the GroupLens [10] project team at Minnesota University were used as experimental datasets. Among them, Movie Lens 100K contains 943 users, 1682 items total 100,000 score records; Movie Lens 1M contains 6,040 users, 3706 items total 1,000,209 score records. Item scores range from 1 to 5. During the experiment, user ratings data and movie genres data were used. Regardless of unknown genre, movies are classified into 18 categories according to genre, such as comedy, music, the West and action. 80% of the initial dataset D was randomly selected as the training set to train the regression model, and the remaining 20% as the testset to predict the score.

4.2. Evaluation index and comparison methods

MAE (mean absolute error) is the average of the absolute errors, which can well reflect the actual situation of the prediction error. The smaller the value of MAE, the closer the predicted value is to the actual value. The concrete implementation is as follows:

$$MAE(pred, act) = \sum_{i=1}^{N} | \frac{pred_{ui} - act_{ui}}{N} | \tag{7}$$

Where $pred_{ui}$ is the predicted score, act_{ui} is the actual score, and N is the number of testsets.

To evaluate the performance of the proposed method, we compared it with KmeansLeader(2018) proposed by Kant et al. [11], DLCAutoRec(2018) proposed by Shantanu et al. [12], DSMMF and DSTNMF(2017) proposed by Li et al. [13], and DMF(2018) proposed by Hu et al. [14].

(1) KmeansLeader is a collaborative filtering algorithm based on the K-means clustering initialized by LeaderRank.
(2) DLCAutoRec is an automatic coder recommendation method based on user attributes and item genres metadata.
(3) DSMMF model and DSTNMF model can make better use of user and item information for collaborative filtering recommendation.
(4) DMF is a collaborative filtering algorithm based on ordinal regression.

4.3. Experimental results

Different parameters in the experiment will affect the experimental results of the algorithm. In this paper, a series of parameters are determined by fitting the data set to obtain the optimal prediction accuracy. The influence of clustering parameters is analyzed by experiments. When the clustering center is 3, the experiment results are the best. The parameters of GBRT model are determined by grid search method.We finally got the best combination, max_depth=8, min_samples_split=8, subsample=0.8, loss=huber, n_estimators=500(ML 100K), n_estimators=800(ML 1M), learning_rate=0.07(ML 100K), learning_rate(ML 1M)=0.11.

The experimental results are evaluated on two datasets. Figures 2 and 3 show the MAE bars of the proposed algorithm on Movie Lens 100K dataset and Movie Lens 1M dataset, respectively. As shown in the figure, the prediction accuracy of the proposed algorithm is better than that of the other four comparison algorithms. The MAE values of Figure 2 and Figure 3 are reduced by more than 2.67% and 0.8% respectively.

In summary, when the algorithm is evaluated on the Movie Lens 100K and Movie Lens 1M data sets, the prediction accuracy is improved to some extent. In general, the algorithm achieves the

predetermined goal of reducing the impact of data sparsity by combining item genres and GBRT models, and improving prediction accuracy.

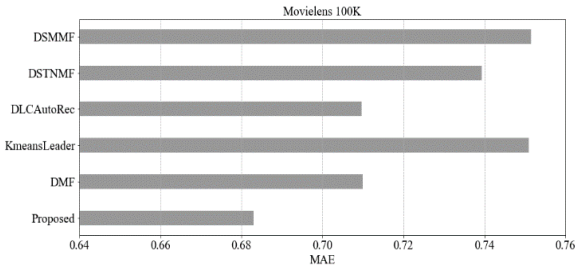


Figure 2. Performance in Movie Lens 100K.

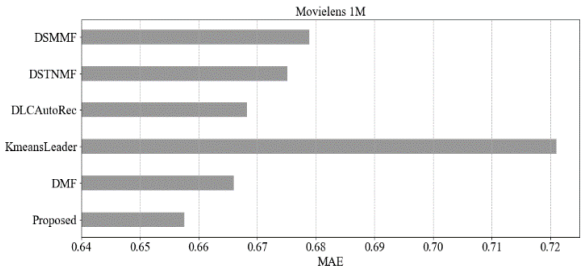


Figure 3. Performance in Movie Lens 1M.

5. Conclusion

Considering the user preference for different item genres, a recommendation algorithm based on item genres preference and GBRT is proposed to reduce the impact of data sparsity on the prediction accuracy of the recommendation algorithm. Compared with the four latest excellent algorithms, the experimental results show that the prediction accuracy of the proposed algorithm is better than that of the comparison algorithm. The items that the number of genres less than 2 in the dataset is more than 80% in this paper, which is still too small compared with the total number of items, and has a great impact on the experimental accuracy. So in the next step, we will consider how to extract more personalized features of users or items to get better experimental results.

Reference

[1] Polattdis N, Georgiadis C K. A multi-level collaborative filtering method that improves recommendations [J]. *Expert Systems with Applications*, 2018, 48:100-110.
[2] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model [C]. *In Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008:426-434.
[3] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems [J]. *Computer, IEEE*, 2009, 42(8):30-37.
[4] Ocepek U, Rugelj J, Bosni Z. Improving matrix factorization recommendations for examples in cold start [J]. *Expert Systems with Applications*, 2015, 42(19): 6784-6794.
[5] Gong S. A collaborative filtering recommendation algorithm based on user clustering and item clustering [J]. *Journal of Software*, 2010, 5(7): 745-752.
[6] Li W, He W. An improved collaborative filtering approach based on user ranking and item clustering [M]. *Internet and Distributed Computing Systems*, 2013: 134-144.
[7] Zhang J, Lin Y, Lin M, et al. An effective collaborative filtering algorithm based on user preference clustering [J]. *Applied Intelligence*, 2016, 45(2): 230-240.
[8] Friedman, J H. Greedy function approximation: a gradient boosting machine. *Ann. Stat.*, 2001.
[9] Kim D, Kim K S, Park K H, et al. A Music Recommendation System with a Dynamic K-means Clustering Algorithm. *In Proceedings of the International Conference on Machine Learning and Applications*, 2008: 399-403.
[10] Harper F M, Konstan J A. The movielens datasets:History and context[C]. *ACM Transactions on Interactive Intelligent Systems*, 2015.
[11] Kant, Surya, et al. LeaderRank based k-means clustering initialization method for collaborative filtering [J]. *Computers & Electrical Engineering*, 2018, 69: 598-609.
[12] Jain S, et al. Doubly Label Consistent Autoencoder: Accounting User and Item Metadata in Recommender Systems. *International Joint Conference on Neural Networks (IJCNN)*, 2018.
[13] Li Y, Wang D, He H, et al. Mining intrinsic information by matrix factorization-based approaches for collaborative filtering in recommender systems. *Neurocomputing*, 2017, 249:48-63.
[14] Hu J, Li P. Collaborative filtering via additive ordinal regression. *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining(WSDM)*, 2018:243-251.