Research on the Application of Item-based Collaborative Filtering Algorithms in MOOC

Qirong Lu¹ and Jiwei Xia²

¹Qirong Lu; April 8, 1966; Doctor; Professor; Current Research Direction: Information Processing; email: Lqr0773@163.com.
²Jiwei Xia; July 9, 1993; Bachelor's Degree; Master's degree; Current research direction: Software Engineering; email: 836417829@qq.com
(Guilin University of Technology, College of Information Science and Engineering, Guilin, China, 541004)

Abstract. In the learning process, the recommendation system can recommend learning resources that are in line with the learning situation of the learning objects and help the learning objects to learn more easily and naturally. By analyzing the characteristics of item-based collaborative filtering algorithm, this paper applies it to the recommendation system of mooc learning resources to avoid the possible defects of this algorithm, such as low performance and offline processing in other scenarios, and gives full play to its good real-time performance and high recommendation efficiency. By analyzing the log files of imooc, this paper obtains the learned learning resources of the learning objects in imooc, takes the first 80% data of historical records, obtains the recommended list of learning resources through recommendation algorithm, and then compares the data of the last 20% of historical records. Finally, the algorithm is optimized according to the accuracy of the recommendation algorithm in the experiment and the actual learning scene.

1. Introduction

Collaborative filtering based on recommendation algorithm through to calculate similarity between items for target items and other items, the similarity of objects in the collection value, and then the history of the use of the current object choice item set to choose the highest similarity values are top-N items, to obtain all of the top-N items sorted and finally choose the final items collection.

The algorithm based on items collaborative filtering is a kind of offline processing algorithm, the algorithm calculate the similarities for all the items before users using it. If the item collection attribute is relatively sparse, and the numbers of items are more than the numbers of the users, these will lower the performance of the algorithm. And when the new items come in, it needs to calculate offline similarity values. In the learning scenario based on mooc, the numbers of learning objects are more than the numbers of mooc resources and the attributes of mooc learning resources are also relatively intensive. In this scenario, the use of item-based collaborative filtering algorithm can effectively avoid the problem of low performance caused by sparse data and large size. For most of the learning objects, the learning time is usually between 8:00 and 24:00. What’s more, the recommendation service can be stopped at other times to calculate the similarity of the new course, which can solve the offline processing problem of this algorithm.

The algorithm based on items collaborative filtering can bring good recommendation effect in the case that there are many long-tailed articles. Moreover, since the recommendation of articles comes from the historical behavior of users, this algorithm often has good user recognition. As long as the
user makes a new item selection, this algorithm will re-recommend according to the user’s new behavior, so the real-time performance of this algorithm is very good, and it can also well avoid the cold start problem of the recommendation system[8].

2. The Research Significance of Learning Platform Based on Recommendation System

Since the emergence of the Internet, the research on the classification and retrieval of information has never stopped, and the way to obtain information has been improving. By integrating the same and similar topics and information or by regional and other methods, portal websites complete the rapid classification of information, which makes the similar information easier to be obtained due to the close display distance. However, with the rapid growth of the amount of information, traditional portal sites are often unable to display in a limited number of pages. By using the reverse indexing technology, search engines can segment the content of information and extract keywords, so that users can quickly find the information they want to get in the massive data through keyword retrieval.

However, in real life, users don't have clear keywords for their own needs. This makes the search engine technology can't completely meet the daily needs. Amazon (Amazon.com)[9] proposed in 2003 to design the recommendation algorithm based on items by calculating the similarity between items, and successfully applied it to the actual business scenarios. This algorithm well completed the recommendation service for millions of commodities and tens of millions of users. The emergence of recommendation system enables users to acquire suitable articles more efficiently under the condition of their relatively vague goals.

In the learning scenes based on mooc, the learning objects usually only have the general direction of learning. For the learning objects with different personalities, the same learning resources often present different evaluations. Therefore, the learning effect of teaching students in accordance with their aptitude cannot be achieved through the scoring of learning resources by the learning objects. This paper adopts the algorithm based on item collaborative filtering and applies it to the recommendation of mooc by judging the similarity between learning resources to help learning objects realize the personalized learning.

3. Theoretical Basis of Algorithms

3.1. Algorithm Flow

In the learning scene oriented to mooc, the algorithm based on item collaborative filtering can be specifically divided into:

(1) Firstly, the similarity between learning resources can be calculated through the attribute vector of learning resources;
(2) For the learning resources of part M, a table of the nearest similarity of M X N can be established, where each learning resource most has N similar learning resources;
(3) Q resources of the learning object that he read the most recently are taken out, and the most N similar learning resources of Q resources are taken out from the M X N table;
(4) Order Q, X and N learning resources according to the similarity value, and then taking out the first N resources, then these make the learning resource recommendation table;
(5) When the learning object selects a new learning resource, repeat steps 3-4;
(6) Repeat steps 1-2 when new learning resources are added in your spare time.
3.2. Cosine Similarity

The basic idea of cosine similarity\cite{10}: setting the collection of judgment objects as a multi-dimensional space to represent a judgment object as a vector, and the cosine value of the Angle between two vectors is the similarity of two decision objects. The smaller the included Angle of the vector, the more similar the two are; otherwise, the less similar they are, the included Angle of two vectors is 180 degrees. This indicates that the two judgment objects are completely opposite.

Similarity range: $[-1, 1]$, the larger the value is, the smaller the included angle is. And the closer the two points are, the higher the similarity will be. The calculation formula is:

$$\text{sim}(A, B) = \cos \theta = \frac{x \cdot y}{\|x\| \times \|y\|} = \frac{\sum x_i y_i}{\sqrt{\sum x_i^2} \sqrt{\sum y_i^2}}$$

4. Practical Application of the Algorithm

4.1. Analysis of Log Data of Mucho Network

The data in the following format are obtained by parsing the log files of imooc.com:

![Figure 2. Data Display Chart](image)

Among them:
1: Represents the learning resource ID;
Search Engine Design With Distributed Reptiles On Python: Represents the name of learning resources;
Introduction | Practical | Algorithms: Represents attribute tags.

This paper divides the learning data records in the imooc log information into two parts. 80% of the records are used for training and 20% for evaluating the accuracy of the recommendation algorithm.

4.2. Data Information Modeling of Learning Resources

According to the attribute label of learning resources, a vector of learning resources is generated:

$$S^i = \{(t^1_r, w^1_r), (t^2_r, w^2_r), (t^3_r, w^3_r), ..., (t^n_r, w^n_r)\}$$

Among them:
$S^i$: Feature vectors representing learning resources;
Represents the nth attribute of learning resources.

Represents the weight of learning resources S for \( t_n \) in the learning resources model.

Then, for each learning resource, the cosine similarity formula is used to calculate the similarity of the label set between the two learning resources, which represents the similarity of the two learning resources.

4.3. Analysis of Experimental Data and Results

Since the recommended learning resources are selected from all collections, the data added in the later stage may exist in the last 20% of the learning resources, so the similarity hit ratio calculated from the experimental data will be higher than the actual situation.

Figure 3. Recommended results of Student 102

In the final experiment, we selected the top 10 learning resources with the highest recommendation similarity for testing. Finally, through the calculation of history records in the log file of 5267 users and the similarity of 312 learning resources, the recommendation success rate was 25.2%.

The experiment can be further improved in the following aspects:

1) Since the computing power at the server level cannot be obtained, the selected experimental data are relatively small, and the model established by the training set and obtained through log analysis is not reliable. If the model is actually used, the learning resources should be optimized.
2) This experiment evaluates the similarity between the learning resource finally selected by the learning object and the learning resource of the recommendation list. Since the learning resource is constantly updated, and the learning resource list used for evaluation directly selects the last 20% data of the learning object's historical learning data, the last two steps of the algorithm are not tested.

4.4. Improvement Based on Recommendation Algorithm of Users for Autonomous Learning Platform

According to the analysis of the experimental results, the user's recommendation algorithm will be simply optimized.

Optimization strategy:
1) After the learning resource is recommended to the learning object in the recommendation system, the learning object will score the effectiveness of the recommended learning resource;
2) Set a threshold for the score of learning objects. When the score is lower than the threshold, the association of two learning resources will be blacklisted;
3) If the score is higher than the threshold, the effective score will be combined with the original score, and the cosine similarity formula will be used again to calculate the similarity between learning resources.

5. Summary and outlook

Through experimental verification, the algorithm based on item collaborative filtering can better handle the current mooc learning mode and provide better recommendation services for learning objects. At the same time, this paper also adapts this recommendation algorithm to the actual application scene, and has achieved good results.

With the rise of the Internet courses, more and more schools will also be more public in teaching resources, and teaching resource property chamber pot becomes more complex. Learning resources, the amount of data of that will be larger and larger. These could make the sharp rise of the calculation of the similarity between the learning resources. Recommending through the ways like neural network and deep learning technology may make better recommendation results. We will further explore the application of recommendation algorithm in the learning field.

6. References

[1] Lü L, Medo M, Yeung CH, et al. Recommender systems. Phys Rep. 2012;519(1):1–49.
[2] Chen Y J J, Irey R, Cunningham A E. Word-level evidence of the role of phonological decoding during orthographic learning: A Direct Test of the Item-Based Assumption[J]. Scientific Studies of Reading, 2018, 22(6): 517-526.
[3] Gao M, Wu Z, Jiang F. Userrank for item-based collaborative filtering recommendation. Inform Process Lett. 2011;111(9):440–6.
[4] Shapira B. Recommender systems handbook. Berlin: Springer; 2011. p. 24–32.
[5] Lee H H, Samei E, Spronk N. Similarity degree of Fourier algebras[J]. Journal of Functional Analysis, 2016, 271(3): 593-609.
[6] Reich J. Rebooting MOOC research[J]. Science, 2015, 347(6217): 34-35.
[7] Deshpande M, Karypis G. Item-based top-n recommendation algorithms[J]. ACM Transactions on Information Systems (TOIS), 2004, 22(1): 143-177.
[8] Lika B, Kolomvatsos K, Hadjiefthymiades S. Facing the cold start problem in recommender systems[J]. Expert Systems with Applications, 2014, 41(4): 2065-2073.
[9] G. Linden, B. Smith, and J. York, “Amazon.com Recommendations: Item-to-Item Collaborative Filtering,” IEEE Internet Computing, vol. 7, no. 1, 2003, pp. 76–80.
[10] Muflikhah L, Baharudin B. Document clustering using concept space and cosine similarity measurement[C]//2009 International Conference on Computer Technology and Development. IEEE, 2009, 1: 58-62.