Research on Collaborative Filtering Recommendation Algorithm Based on Mahout

Hui CAO* and Li-yang YAN

Key Laboratory of China's Ethnic Languages and Information Technology of Ministry of Education, Northwest Minzu University, Lanzhou, Gansu, China

*Corresponding author

Keywords: Mahout machine learning, Collaborative filtering algorithms, Book recommendation.

Abstract: This paper studies the recommended algorithm for the Mahout machine learning platform. The principle analysis of the current mainstream recommendation algorithm is based on project-based collaborative filtering recommendation. The recommendation algorithm of Book-Crossing data set is implemented by using the collaborative filtering algorithm provided by Mahout. The similarity distance and other parameters in the general recommendation algorithm are used to compare and analyze the recommended results.

Introduction

With the rapid development of Internet technology, and with the progress of intelligent terminal equipment, mobile Internet has risen. It makes more convenient for people to publish and share information, at the same time, it brings a lot of data information to people. When we enjoy the convenience brought to us by the information age, we also produce various kinds of information. The application scenario of traditional search engine is that users can clearly know their needs through keywords and words to search. However, when users cannot express their needs or have no clear and effective search content, recommendation system emerges as an emerging technology to make up for the shortcomings of traditional information search engines. It uses different recommendation algorithms to model the user's preferences, and predicts the items or information that the user may be interested in according to the model to recommend a user.

Project-based collaborative filtering recommendation algorithm is one of the most widely used and effective recommendation algorithms. Gradually, recommendation systems have become the main functions of IT companies that rely on information and data, such as Taobao, Today's headlines and NetEase Cloud Music. The development of recommendation algorithms has developed rapidly from collaborative filtering algorithm to implicit semantic model, and then to deep learning model. The goal of recommendation system is to predict users' preferences through accurate calculation, to achieve the best recommendation effect by coordinating algorithms, system functions and user experience, and to enhance consumers' user experience with more intelligence and humanity. By analyzing the Mahout Recommendation algorithm and taking the book recommendation system as an example, the results of the recommendation algorithm under different parameters are analyzed and compared.

Recommendation Based on Collaborative Filtering

The recommendation algorithm based on collaborative filtering is one of the most mature algorithms in the recommendation system. The core idea of recommendation based on collaborative filtering: using user behavior data information to extract features from users, which finds new user-to-item correlations by calculating user-to-item correlation to recommend for current users.

Mainstream collaborative filtering algorithms include user-based filtering recommendation (User-Based CF) and Project-based Collaborative Filtering Recommendation (Item-Based CF) algorithm. These two collaborative filtering algorithms will be introduced below.
**User-Based CF:** Recommend to User A items that are of interest to User B and which User A has not browsed. When user A is recommended by the system, user item set B, which is similar to A preference, is found by calculating user history information, and the items that user A has not purchased in item set B are recommended to A. The algorithm is divided into two steps: first, only user B with similar preference to A is found, and recommendation A is not purchased from item B.

**Item-Based CF:** Recommend to User A the similar item B of the item A bought before. It does not calculate the similarity between items according to their content attribute characteristics. It calculates the similarity between the items to be recommended based on the user's historical information. If most people like item A and item B are the same, then item A and item B are similar. We will recommend item B to someone who likes item A but does not choose item B. The algorithm is divided into two steps: first, the similarity between items is calculated and a recommendation list is generated for users according to the similarity between users and items.

**Mahout Framework**

**Mahout Overview**

Mahout is an open source project under Apache. It integrates some commonly used machine learning algorithms in its framework, and improves developers' development efficiency by opening common algorithm interfaces. The main algorithms in the framework are divided into three categories: recommendation algorithm module, clustering algorithm module and classification algorithm module. Mahout framework has a large number of machine learning algorithms, such as some commonly used clustering algorithms and recommendation algorithms to support MapReduce computing. The recommended algorithm used in this paper is implemented by Taste interface in Mahout framework.

Taste is an implementation class that provides a large number of recommended computations. It is an open source extensible algorithm developed in Java language. Taste's recommendation algorithm includes collaborative filtering algorithm based on project, recommendation algorithm based on SlopeOne and recommendation algorithm based on singular value decomposition, so that developers can easily implement or rewrite their own recommendation algorithm. Taste mainly includes the following interfaces as Fig. 1:

![Figure 1. Taste working principle component diagram.](image)
(1) Data Model: It is the interface that receives the user's original data, and then transfers the data to the interface to convert it into [userId, itemId, pref] data format. It can support extracting user preference information from multiple file formats. User information can be read from traditional CSV or database.

(2) User Similarity and Item Similarity: User Similarity is used to calculate the similarity between two users (items), which is the core of Project-based Collaborative Filtering recommendation.

(3) User Neighborhood: User Neighborhood defines the user's (item's) neighbor through the similarity calculation above.

(4) Recommender: It is the final implementation interface in the recommendation algorithm. The results of the above steps (1), (2), (3) are transmitted to the interface and the user's recommendation list is finally obtained.

**Recommended Algorithm Comparison**

Mahout framework provides several collaborative filtering algorithms. By analyzing the characteristics and applicable scenarios of these algorithms, the comparison table of algorithm characteristics is obtained as follows as Table1.

<table>
<thead>
<tr>
<th>Number</th>
<th>Recommendation algorithm</th>
<th>Algorithmic characteristics</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>User-CF</td>
<td>Relatively fast when the number of users is small</td>
</tr>
<tr>
<td>2</td>
<td>Item-CF</td>
<td>Faster with fewer items</td>
</tr>
<tr>
<td>3</td>
<td>Slope-One</td>
<td>Recommend and update data quickly pre-calculation is needed and less suitable items are needed.</td>
</tr>
<tr>
<td>4</td>
<td>SVD</td>
<td>Good results, need to be calculated in advance</td>
</tr>
<tr>
<td>5</td>
<td>Linear Difference</td>
<td>Less articles, better results</td>
</tr>
<tr>
<td>6</td>
<td>Clustering method</td>
<td>Fast recommendation, need pre-calculation, less users, good results</td>
</tr>
</tbody>
</table>

**Recommended Algorithm Implementation**

**Recommended Algorithm Comparison**

Data Model implements storage and provides preferences, users, and item data for computing. We need to map the original data to a Mahout compatible format. In this process, we need to pre-process the original data, User Similarity, Item Similarity, and User Neighborhood. Recommender obtains the calculation results through the above steps and transmits them to the recommender to implement the recommendation algorithm. Meanwhile, Mahout also provides the method of recommendation evaluation. Similarity calculation is an important part of recommendation system. The specific algorithm steps are as follows:
(1) Collect relevant data such as user or item features, and preprocess the collected feature data to reduce redundancy and null value (i.e. reduce sparseness). Reduce the impact of information beyond the data itself on recommendation results.

(2) Tuning recommendation component similarity calculation component and neighborhood relation component. The calculation of similarity is between users or objects. For similarity measurement, there are many methods, such as logarithmic likelihood ratio similarity, Pearson coefficient similarity, Euclidean similarity and so on. Then the nearest neighbor set is selected by the nearest neighbor algorithm component.

(3) User's recommendation forecast. By calculating similarity and neighborhood, the parameters are passed to the recommender. Finally, the user's recommendation list is generated to complete the recommendation.

(4) After completing the recommendation calculation of all users, the performance of the recommendation results is evaluated by the score component.

Algorithm Implementation

(1) Data acquisition

The recommendation engine inputs preference data, the most basic abstraction of Preference, and generates a set of user preference data in a simple triple format <user_id, item_id, preference>. Recommendation engine Taste defines three different file formats to match different data sources through policy models. The first is Generic Data Model at memory level. The second method uses FileData Model to read data from the file system. The third way is to get data through JDBC Data Model. Our data source is stored in HDFS through log system processing. Here we will use FileData Model to obtain user data file formats as UserID, ItemID, and Preference.

(2) Similarity and Neighbor Computing

Similarity algorithm will affect the results of user and project similarity calculation. The more accurate the similarity calculation is, the higher the accuracy of recommendation results is. Later, the test section will test the recommended results through different combinations. Mahout framework provides a variety of similarity calculation methods. They include Euclidean distance similarity and Pearson similarity. Nearest Neighbor Computing generates the corresponding users' nearest neighbor from the result of similarity calculation. The commonly used nearest neighbor algorithms include the number of fixed size nearest neighbor specified N and the threshold-based nearest neighbor specified distance.

(3) Recommendation algorithm

The recommendation engine first analyses the input data source DataModel, calls the similarity component to calculate the similarity, and the nearest component to calculate the nearest neighbor. Finally, the recommender calculates the recommendation list.

(4) Result Scorer

Recommender Evaluator is a component used in recommendation engine to evaluate the performance of recommendation results. Usually we can calculate the average difference between estimated preferences and actual preferences to compare. The lower the value obtained, the better the evaluation method is MAE (average difference)

Result Analysis

The process of generating recommendation results by Mahout recommendation algorithm. It can be seen that the construction of recommendation engine can produce recommendation results through the combination of different similarity, nearest neighbor algorithm and recommendation algorithm. Comparing the recommendation results of the same test set, we choose UserCF, ItemCF and two recommendation algorithms to conduct a centralized combination test. The recommendation
performance was tested by scorer. The results include MAE value, Precision value and Recall value. The results of the code operation are as follows.

To facilitate the observation of differences between data results, visualize them as follows as Fig.4. The smaller the MAE value is, the better is. The more the average absolute difference is, the more stable the recommended results are. If the value is too large, it indicates that the recommended results are far from accurate. Comparisons between user-based recommendation combinations using Euclidean similarity based on histogram

Figure 3. Scorer results.

The closer Precision and Recall are to 1, the better is. By observing the scatter plot as Fig.5, it is found that only Euclidean similarity algorithm based on user's no-score has better evaluation score, while other algorithms are not ideal. Different recommendation results produced by different combination of recommendation algorithms are biased. If we use larger data sets, we can get more accurate evaluation results.

Figure 4. MAE Value.

Figure 5. Precision and Recall.
Summary and Review

Personalized recommendation, big data and so on are the research hotspots in various industries. This paper studies the common recommendation algorithm, analyses the common recommendation algorithm in the framework of Mahout machine learning, studies the implementation process of the recommendation algorithm, and finally compares the recommendation effect of the recommendation result by adjusting the relevant parameters through the implementation of collaborative filtering recommendation algorithm. Of course, the design of a complete recommendation system involves many fields, which is huge and complex system engineering. This paper is just a simple comparison of recommendation algorithms. In practice, there are still many problems to be solved, which affect the recommendation accuracy and timeliness of personalized recommendation system.

References


