Spark-based Parallel Collaborative Filtering Recommendation Algorithm

Yongli Yang[1, a], Fei Xue[2, b], Yongquan Cai[1, c] and Zhenhu Ning[1, d,*]

[1]Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China;

[2]School of Information, Beijing Wuzi University, Beijing 101149, China.

[a]yyyyll1218@163.com, [b]xuefei2004@126.com, [c]cyq940218@163.com, [d]nzh41034@163.com

Keywords: Collaborative Filtering Recommendation Algorithm, RLPSO, K-means, Spark.

Abstract. The rapid development of Internet information technology makes the problem of information overload become more and more serious, and recommendation system is one of the effective ways to solve this problem which is favored by people. However, for the massive data information, the recommended algorithm faces the bottleneck problem of processing speed and computing resources, so this paper proposed a parallel collaborative filtering recommendation algorithm based on Spark. The RLPSO algorithm is used to optimize the clustering factor of the K-means clustering algorithm by associating users with similar interests into a cluster and using the recommended algorithm for users to recommend is implemented on the Spark platform. The experimental results show that the improved algorithm has a significant improvement in the prediction accuracy, and has a higher speedup and stability compared with the traditional collaborative filtering recommendation algorithm.

1. Introduction

The rapid development of Internet information technology makes the information overload problem become more and more serious. As a technical solution to Internet information overload, the recommendation system is drawing increasingly more attention from the industry [1]. The recommendation system can learn the behavior of the users, and understand their preferences, so as to help recommend what they may be interested in. Among many recommendation methods, collaborative filtering algorithm is currently the most widely used. Collaborative filtering algorithm is divided into two categories: memory-based and model-based [2], the memory-based collaborative filtering is divided into user-based and item-based collaborative filtering, which process is mainly based on the scoring matrix. While model-based collaborative filtering from the existing score matrix to learn a compact model and learn to predict from this model. At present, the commonly used models include Regression Model, Bayesian Model, Clustering Model, Markov Model, Latent Factor Model, Singular Value Decomposition Model, and Restricted Boltzmann Machine [3].

Although the collaborative recommendation algorithm has achieved great success, but there still exists some problems such as cold start, data sparse, and algorithm scalability and so on. In [4], the clustering algorithm is used to solve the problem of scalability of traditional collaborative filtering algorithm, while reducing the time complexity of the algorithm. In [5,6], a multi-clustering algorithm is proposed, which combines all users and items into several user-item clusters. Each cluster contains some user and item data. Experiments show that the accuracy of the recommended results has improved compared with the original algorithm. In [7], the multi-standard collaborative filtering algorithm extended horizontally and ran by adding the computing nodes. The experimental results show that the algorithm has not been improved obviously with the increase of nodes. In [8], the Tanimoto coefficient is used to calculate the similarity, and the recommendation based on the Spark platform is more efficient compared with the Hadoop. The above algorithm improves the recommendation result to a certain degree, but it uses the user rating information to calculate the similarity, ignoring the user group with similar interest to the target user group has better reference value than other users. In this paper, clustering algorithm is used to cluster the user information, and the Spark distributed platform is used to perform parallel computation while facing the problem of low efficiency of data processing. Based on the K-means clustering algorithm [10], the clustering analysis of the scoring data is based on the user group with similar interest to the target user, users with similar interests are clustered into one cluster, and recommend the product to the target user by using the recommendation algorithm, according to the user group with similar interest with the target user has higher reference value than other users. In this paper, the initialization factor of K-means clustering algorithm is optimized by RLPSO [11] algorithm for the uncertainty of K-means clustering algorithm initialization factor.

2. RLPSO_KM_CF Algorithm Based on Spark

In the RLPSO algorithm, the iterative process of each particle is independent of each other, and each particle iterates the same task, the larger the task granularity is, the longer the computation time will be. The same problem is also reflected in the K-means algorithm, the iterative calculation of the data object and the distance from the center of the cluster are then classified and the clustering center is updated, which challenges the time efficiency of the algorithm. In view of the above situation, this paper proposes the Spark-based RLPSO_KM_CF algorithm, which effectively supports the iterative operation and improves the efficiency of the algorithm. The algorithm is divided into three stages:

The first stage: Parallel design and implementation for RLPSO algorithm. In this paper, the process of initializing particle swarm information, constructing solution, local searching local solution and reverseing learning is independent as a parallel unit. The optimal solution of each particle swarm is solved by Spark parallel processing n particle swarm information, and finally the optimal solution of the whole particle swarm is updated and the optimal solution of particle swarm is printed.

The second stage: The optimal solution of particle swarm is decomposed as the initial clustering center of clustering algorithm. In the Spark, RDD is used to divide the data samples into n compute nodes. The cluster center is shared among the compute nodes by means of the broadcast method of SparkContext. The initialized clustering center is distributed to all nodes, and for each data slice, the Euclidean distance with each cluster center is calculated. The sum of the data belonging to the cluster is calculated according to each cluster center. Merging the sum of the data for each node that is to determine whether sum value is less than the threshold. Use the map and reduce functions to complete the operation of updating the cluster center until the algorithm completes the number of iterations or reaches convergence and prints the clustering center and the clustering results.

The third stage: Through the second stage of the output of the cluster center and clustering results. First of all, to determine whether the target user is a new user, if it is a new user, according to the item popularity formula, calculate the highest prevalence of items N items and then recommend item to the target user. Otherwise, sort the user's item rating information and persist these information through the map function; According to the cluster center to calculate the user belongs to the cluster, use the groupByKey function to obtain the target user rating of the item information as RDD1; according to the user similarity formula to calculate the similarity between users within the cluster and obtain the highest degree of similarity with the target user information as RDD2 through the filter function; obtain the target user's neighborhood user evaluation of the item information as RDD3 Through the groupByKey function; return to the target user has not evaluated the item information as RDD4 through the distinct funciton between RDD1 and RDD3;The score information of the item information in the RDD 4 is calculated by the score prediction formula, and finally the N item with the highest score is recommended to the target user.

3. Experiments

3.1 The Experimental Environment and Parameter Settings

According to the research, in the experimental environment we use centos7.0 equipment system server server with the bridge mode to deploy eight devices, including seven work nodes and one master node;the Spark version is 2.0, and Hadoop version 2.7. At the same time, in order to test the difference in the recommended quality between the collaborative filtering recommendation algorithm based on the multi-similarity between users and the traditional User-based collaborative filtering recommendation algorithm, this paper uses the MovieLens data set provided by the University of Minnesota's GroupLens Research laboratory. In this paper, three methods are selected as the contrast algorithm: the traditional UserCF collaborative filtering recommendation algorithm, the improved Top-N clustering collaborative filtering recommendation algorithm KCF, and the RLPSO_KM_CF collaborative filtering recommendation algorithm proposed in this paper.In order to reduce the complexity of the model calculation, the parameters set by the RLPSO algorithm are set by the parameters proposed in [12]. The RLPSO_KM clustering algorithm is used to divide the users into 2, 3, 4, 5, 6, 7 and 8 clusters, and the traditional user-based collaborative filtering recommendation algorithm is applied to each cluster in each time. Finally, calculate the average recall rate and MAE value.

3.2 Experimental Analysis

According to in Fig. 1, KCF and RLPSO_KM_CF are superior to the traditional User-CF algorithm, regardless of the influence of random factors. And the RLPSO_KM_CF algorithm is better than the KCF algorithm. At the same time, we can see from the figure, with the increase in the number of clusters, E-measure values of KCF and RLPSO_KM_CF algorithm are gradually decreasing and get the maximum value when the K value is 2 and K value is 4 respectively, which also shows that with the increase in the number of clustering factors, the target user neighborhood set gradually less recommendation will be reduced in accuracy. But on the whole, the RLPSO_KM_CF algorithm proposed in this paper has advantage on F-measure. At the same time, it improves the recommended accuracy rate.

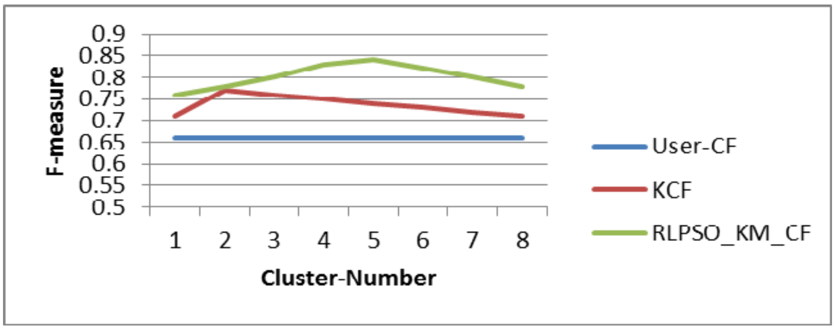

Figure 1. F-measure

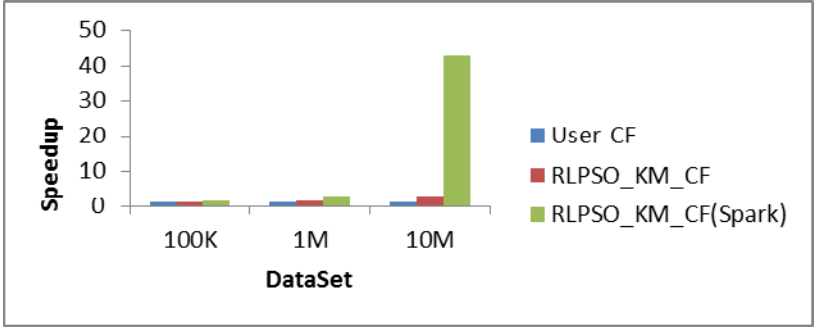

Figure 2. Speedup

According to the given three datasets, the traditional User-CF algorithm and RLPSO_KM_CF algorithm are executed in a stand-alone environment, and the RLPSO_KM_CF algorithm is implemented in the Spark environment to compare the acceleration ratio. The experimental results are shown in Figure2. As can be seen from the figure, in the data set for the 100K and 1M, the advantages of parallel algorithms cannot be well reflected. But with the increasing size of the data set, Spark parallelization gradually become obvious, which fully reflects the Spark-based memory computing model in the time overhead on the huge advantage. At the same time, the contrast speedup ratio between the traditional collaborative filtering recommendation algorithm and the RLPSO_KM_CF algorithm in the serial environment, the acceleration ratio reaches 3 times of the traditional collaborative filtering recommendation algorithm when the dataset is 10M.

4. Summary

The traditional collaborative filtering recommendation algorithm refers to the rating information of all users when recommending for a target user. However, when recommending for a target user, users who are more similar to it are clearly more valuable than other users. Based on the traditional cooperative filtering, this paper proposes a collaborative filtering recommendation algorithm based on RLPSO_KM_CF. The RLPSO_KM algorithm is used to cluster according to the distance between sample points by applying the user-based filtering algorithm to recommend for each other. In order to solve the problem of K-means algorithm itself and improve the clustering effect, this paper proposes to optimize the clustering factor by using RLPSO algorithm. In addition, this paper achieves the RLPSO_KM_CF collaborative filtering algorithm based on Spark platform. The experimental results show that the algorithm proposed in this paper has improved the accuracy rate and proved the correctness of the proposed algorithm. This paper chooses only a suitable clustering algorithm to study. In the future research, we can consider choosing some clustering algorithms suitable for sparse matrix with the traditional collaborative filtering algorithm faced the cold start problem and scarcity of scoring matrix.

References

[1]. Ricci F, Rokach L, Shapira B. Introduction to Recommender Systems Handbook [M]// Recommender Systems Handbook. Springer US, 2011:1-35.

[2]. HUANG Zhen-hua, ZHANG Jia-wen, TIAN Chun-qi, et al. A review of recommended algorithms based on sorting learning [J]. Journal of Software, 2016, 27(3):691-713.

[3]. Salakhutdinov R, Mnih A, Hinton G. Restricted Boltzmann machines for collaborative filtering[C]// International Conference on Machine Learning. ACM, 2007:791-798.

[4]. Sarwar B M,karypis G,Konstan J,et al.Recommender systems for large-scale e-commerce:Scalable neighborhood formation using clustering[C]//Proc of the 5th Conference on Computer and Information Technolofy.2002.

[5]. Xu B, Bu J, Chen C, et al. An exploration of improving collaborative recommender systems via user-item subgroups[C]// 2012:21-30.

[6]. Chen Z, Cai D, Han J, et al. Locally Discriminative Coclustering [J]. IEEE Transactions on Knowledge & Data Engineering, 2012, 24(6):1025-1035.

[7]. Wijayanto A, Winarko E. Implementation of multi-criteria collaborative filtering on cluster using Apache Spark[C]// International Conference on Science and Technology-Computer. IEEE, 2017.

[8]. Kupisz B, Unold O. Collaborative filtering recommendation algorithm based on Hadoop and Spark[C]// IEEE International Conference on Industrial Technology. IEEE, 2015:1510-1514.

[9]. Zaharia M, Chowdhury M, Das T, et al.Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computering[C]//Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation.2012:141-146.

[10]. JiaWei Han Micheline Kamber Jian Pei.Data Mining Concepts and Techniques Thrid Edition[M]. Machinery Industry Press. 2012:293-297.

[11]. XIA Xue-wen, LIU Jing-nan, GAO Ke-fu, et al.Particle swarm optimization with reverse learning and local learning ability [J]. Journal of Computer Science, 2015(7):1397-1407.