Implementation of Parallel Algorithm Technology for Time Series Data Mining

Wei Wang\textsuperscript{1,*}, Xiaohui Hu\textsuperscript{2}, Mingye Wang\textsuperscript{1} and Yao Du\textsuperscript{1}

\textsuperscript{1}College of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China
\textsuperscript{2}Institute of Software Chinese Academy of Science, Beijing 100190, China

*Corresponding author e-mail: ww56050006ww@aliyun.com

Abstract. With the rapid development of computer technology, Internet technology and artificial intelligence technology, the amount of global data has exploded. However, the single-machine serial mode of traditional data mining cannot be directly transplanted to the cloud platform. Only by parallelizing and improving many classic data mining algorithms can the cloud computing platform and data mining be effectively combined. Therefore, it is of great significance to the research and implementation of parallel algorithm technology for time series data mining. The purpose of this paper is to study the research and implementation of parallel algorithm technology for time series data mining. This paper adopts the method of literature data, mathematical statistics, logic analysis and other research methods to study the parallel algorithm technology research and realization of time series data mining, mainly to make useful explorations of time series data mining and visualization technology. It embodies the design ideas of big data analysis tools, and finally reflects the power and market value of data analysis tools through the display of the platform. Research shows that running in the same data set and the same experimental environment, the improved parallel collaborative filtering algorithm ACF in this paper has higher time running efficiency than the parallel algorithm MCF based on the co-occurrence matrix, and in the case of larger data sets, the more obvious the time difference.

Keywords: Data Mining, Parallel Algorithms, Artificial Intelligence, Time Series

1. Introduction

With the rapid economic growth, urbanization and mobilization work further accelerate, the urban population continues to grow, and the scope of the city continues to expand [1-2]. Over the centuries, time series data mining technology has developed rapidly. From the initial similarity analysis to the current interdisciplinary research of artificial intelligence, time series data mining technology has had multiple research directions [3-4]. In order to further improve the efficiency of data mining, choose to run traditional data mining algorithms on the Hadoop cloud computing platform [5-6]. It is
advantageous to have a good scaling ratio and speed ratio, and is conducive to effective big data processing data mining algorithms [7-8].

In the research and application of parallel time data mining algorithm technology, many researchers have conducted research on it and achieved good results. For example, Pearce M et al. They divide time series data mining into four aspects, such as trend analysis, similarity search, sequential pattern mining and periodic pattern mining. There is no single standard for classifying time series data mining, and optical data mining is a hot spot in current data mining research. Visualizing the results of time series forecasting can provide great convenience to end users [9]. For example, Malyshkin. Victor and others. A visualization method of economic time series forecasting based on nuclear-SOM is proposed, which improves the effect of time series forecasting visualization and takes the forecast result as the target. By using appropriate visualization methods, various data can be visualized. Visualized data will enable users to intuitively discover the indirect dependencies between functions and data, and provide very good help for data analysts [10].

This article adopts the literature research method to search the literature and works related to "data mining", "time series", "parallel algorithm", etc., to study and learn from the new technology, and extract useful information from it; mathematical statistics: right The results of the experiment are drawn and then analyzed to obtain effective results.

2. Research and Implementation of Parallel Algorithm Technology for Time Series Data Mining

2.1 Timing Analysis Techniques
The following are three commonly used linear timing models.

(1) Moving average model (MA)
In MA, the time sequence \(x_t\) is a white noise sequence \(e_t\) and an N-order filter with coefficients \(b_0, b_1, \ldots, b_n\) to generate a convolution filter. The specific formula is as follows:

\[
x_t = \sum_{n=0}^{N} b_n e_{t-n} = b_0 e_t + b_1 e_{t-1} + \ldots + b_n e_{t-N}
\]

This formula shows that the current value of the sequence \(\{x_t\}\) is determined by the sequence \(\{e_t\}\), which is determined by the sequence value in the window of length \(N\) from the current value.

(2) Autoregressive model (AR)

\[
x_t = \sum_{m=1}^{M} a_m x_{t-m} + e_t
\]

In the AR model, the current value of the sequence \(\{x_t\}\) is determined by the current value of the sequence \(\{e_t\}\) and the sequence value in the previous window of length \(M\) of the sequence \(\{x_t\}\).

(3) Autoregressive moving average model (ARMA)
ARMA is composed of two parts, AR and MA, in the following form:

\[
x_t = \sum_{m=1}^{M} a_m x_{t-m} + \sum_{n=0}^{N} b_n e_{t-n}
\]

In the ARMA model, the current value of the sequence \(\{x_t\}\) is determined by the current value of the sequence \(\{e_t\}\) in the window of length \(N\) and the previous sequence value of the sequence \(\{x_t\}\) in the window of length \(M\).

2.2 Parallel Algorithm MapReduce and Spark
(1) MapReduce programming model
MapReduce is a parallel programming model applied to large-scale data processing. The bottom layer of MapReduce hides many details needed for distributed computing, and highly abstracts a
distributed computing task into two functions: map function and reduce function. The input and output of these two stages are based on KV key-value pairs. After the map phase is completed, the results of all map tasks are stored on the local hard disk, and Hadoop will automatically sort them according to the keys in the results, and then distribute them to various reducers. In the reduce phase, the results are merged according to the <key, value> key value output in the map phase, and the key-value pairs of the same key will be distributed to the same reducer. Similarly, the reduce task is also programmed by the user, and only needs to implement the reducer interface. Finally, the result of the reduce task will be stored in HDFS.

2) Spark
Spark is a distributed editing framework started by Apache, similar to MapReduce, for distributed big data computers. Spark has the advantages of MapReduce, but unlike MapReduce, Spark can store the intermediate output of a task in memory, and it can calculate the next task without rereading HDFS, thus speeding up the calculation. Therefore, Spark is more suitable for calculations that require multiple iterations, especially data mining and machine learning algorithms. Spark is based on in-memory computing, and its performance and speed are better than MapReduce. It is easier for developers to use, and it is more suitable for applications with long delays and big data forwarders. Therefore, Spark is expected to replace MapReduce and become the leader of the next generation of big data applications.

2.3 Parallelization Improvement of Collaborative Filtering Algorithm
(1) Algorithm parallelization ideas
It is easy to do it through MapReduce or Spark, with <user ID, item ID, rating> as input, user ID as key, and <item ID, rating> as value output. That is, the same user rating tuple as the rating matrix is obtained: {user ID, (itemID1, rating1), (itemID2, rating2),...}, (item IDk, ratingk)).

This tuple contains the rating information of all items of a user. In the next job processing, the map stage takes the tuple as input, uses a double loop to traverse all the user's rating item pairs, and generates an output with <item Idi, item IDj> as the key-value pair, indicating that item i and item j are common. Increase the number of occurrences by 1. The final result is the co-occurrence matrix we want. Taking the above tuple as a row in the co-occurrence matrix, the co-occurrence matrix is divided and stored on multiple different nodes.

(2) Problems with the algorithm
1) When there are many projects, the common appearance panel will be too large, and it will take a lot of time to build the common appearance panel. When calculating the target user prediction score vector, it takes a lot of time to multiply the co-occurrence table with the user score vector. In distributed computers, matrix multiplication is not efficient.

2) The collaborative filtering algorithm based on the co-occurrence matrix ignores the role of neighbor users. According to the number of times the items appear together, the more relevant the relationship between them may be, rather than the similarity calculation method, which affects to a certain extent Recommended accuracy.

(3) Improved parallel collaborative filtering algorithm ACF
1) Generate scoring matrix
When using MapReduce or Spark to calculate a set of score triples, in the map phase, each row of data <user ID, item ID, rating> is used as the value input, user ID is used as the key, and <item ID, rating> is output as the value. In the reduce phase, the <item ID, rating> corresponding to the same user ID will be distributed to the same reducer, and then the value will be merged according to the user ID to get this KV output: <user ID, list(item ID, rating)>. This is the same as each row of data in the scoring matrix. Since multiple nodes are calculated in parallel, the output results of the reducer on different nodes will be stored on different nodes, and finally the user's score matrix is generated.

2) Get neighbor users
In this step, the similarity calculation method is used to find K neighbor users of the target user. Use the K minimum heap algorithm to find the K maximum similarities.
3) Form a recommendation
In this step, the recommendation prediction algorithm is used to calculate the target user's expected value of the items rated by neighbor users, and the top N items with the largest expected value are selected as recommendations.

From the above three steps, it can be seen that the parallelization of the improved ACF algorithm is simple and easy to understand and easy to implement. It avoids building a huge co-occurrence matrix and complex matrix multiplication operations, and finds out the neighbor users of the target user through calculation, and finds the best recommendation item in the neighbor user rating items, which conforms to the idea and process of the traditional collaborative filtering algorithm.

3. Experimental Research on Parallel Algorithm Technology Research and Implementation of Time Series Data Mining

3.1 Research Content
This experiment uses Movie Lens datasets of 100W, 200W, 300W,...1000W score records, and applies these datasets to the implemented Spark-based collaborative filtering algorithm to calculate the calculation time of each input dataset, And then present the results through graphs.

3.2 Data Collection
The experimental data comes from the Movie Lens dataset provided by Group Lens. Movie lens was founded by the Group Lens laboratory, non-commercial, for the purpose of researching recommended technology. Movie Lens contains multiple data sets of different sizes, and you can estimate the quality of the recommendation algorithm by testing different data sets of Movie Lens.

4. Experimental Research Analysis of Parallel Algorithm Technology Research and Implementation for Time Series Data Mining

4.1 Improved ACF Algorithm Running Time Graph
The collaborative filtering algorithm implemented in this article is not aimed at the improvement of the algorithm itself, but to make the algorithm adapt to the distributed computing based on the cloud platform to make improvements, especially under the premise of big data. Therefore, the running time can be used to measure whether the improved parallel collaborative filtering algorithm has improved its operating efficiency. This experiment uses an improved parallel collaborative filtering algorithm based on a single machine and a distributed cluster composed of several nodes, and runs on data sets of different sizes. Run multiple times for each combination, and take the average as the final result. You can see the specific running time of each job on the Spark application web page, as shown in Table 1.

Table 1. Improved ACF Algorithm Running Time Table

| Data volume (Unit: ten thousand) | Node 1 | Node 3 | Node 5 |
|---------------------------------|--------|--------|--------|
| 100                             | 50     | 42     | 32     |
| 300                             | 65     | 52     | 41     |
| 500                             | 110    | 75     | 51     |
| 700                             | 142    | 79     | 59     |
| 900                             | 183    | 103    | 76     |
Figure 1. Improved ACF Algorithm Running Time Chart

It can be seen from Figure 1 that under the same size data set, the time required for the algorithm to run decreases as the number of nodes increases. In the case of a small data set, the time difference is small, and as the data set increases, the time difference is obvious. This reflects the ability of parallel computing.

4.2 Speedup Ratio

Speed-up, usually refers to the ratio of the time consumed by the same job running in a single-processor system to the time consumed by parallel processing by multiple processors. It is used to measure parallel system processing or program parallelization. The higher the speedup, the better the parallel performance.

In order to test the parallel processing performance of the enhanced parallel collaborative filtering algorithm, this experiment uses different data D1 (100W score records), D2 (400W), D3 (700W), D4 (1000W) and different data on a single computer. Run the cluster at the node number of, calculate the time required, and take the average of multiple paths to obtain the final result. Next, calculate the speedup ratio under different conditions, and the results are shown in Table 2:

| Data Set | D1   | D2   | D3   | D4   |
|----------|------|------|------|------|
| 1 node   | 1    | 1    | 1    | 1    |
| 2 node   | 1.22 | 1.2  | 1.3  | 1.55 |
| 3 node   | 1.3  | 1.48 | 1.65 | 2.2  |
| 4 node   | 1.6  | 1.58 | 2.1  | 2.6  |
| 5 node   | 1.75 | 1.8  | 2.3  | 2.7  |
| 6 node   | 1.9  | 2.1  | 2.55 | 3.1  |
Figure 2. Speed-up diagram

It can be seen from Figure 2 that the improved parallel collaborative filtering algorithm has a significant speedup effect. With the same number of nodes, the larger the data set, the larger the speedup, which reflects the ability of Spark to process big data. And as the number of nodes increases, the speedup ratio will increase accordingly, and finally tends to be slow. Because each node needs a network for communication, as the number of nodes increases, the network overhead in the calculation process also increases. Moreover, the experimental data itself is not large enough, and the cluster is still small (subject to experimental conditions), which is not sufficient. Reflects the actual ability to process big data in parallel. Therefore, the acceleration is slower than the last trend.

5. Conclusions
Now that we have entered the era of big data, data is growing at an alarming rate. How to effectively organize data, how to dig out effective information from massive data, and how to make data generate value are all problems we face to solve. Data mining algorithms and tools can effectively discover knowledge from data. However, when faced with large-scale data, traditional data mining algorithms have no effect. Therefore, various research institutions are studying the problem of parallelization of data mining algorithms. Combining the characteristics of cloud computing and using cluster parallel computing capabilities, the efficiency of data processing is greatly improved. Moreover, as the number of cluster nodes increases, the computing speed is also increased, and will speed up accordingly.

References
[1] Wang R, Guo S, Li Y, et al. Fisher Discriminative Dictionary Learning for Vehicle Classification in Acoustic Sensor Networks. Journal of Vlsi Signal Processing Systems, 2017, 86(1):99-107.
[2] Bahig H M. A new constant-time parallel algorithm for merging. Journal of supercomputing, 2019, 75(2):968-983.
[3] Oh B W. Parallel Algorithm for Spatial Data Mining Using CUDA. JOURNAL OF ADVANCED INFORMATION TECHNOLOGY AND CONVERGENCE, 2019, 9(2):89-97.
[4] Charan L S, Reddy M V, Reddy N P, et al. 5:3 compressor based neural network with LM Algorithm in Multiplier as Application. IOP Conference Series Materials Science and Engineering, 2021, 1042(1):012033.
[5] Fenner S, Gurjar R, Thierauf T. A Deterministic Parallel Algorithm for Bipartite Perfect
Matching. Communications of the ACM, 2019, 62(3):109-115.

[6] Dou W, Miao S. Performance evaluation of parallel re-computing algorithm in different data distribution modes. Journal of Algorithms & Computational Technology, 2018, 12(1):43-52.

[7] Phu V, Chau V T N, Tran V T N. SVM for English semantic classification in parallel environment. International Journal of Speech Technology, 2017, 20(3):487-508.

[8] Lavrinenko Y S, Morozov I V, Valuev I A. High performance wave packet molecular dynamics with density functional exchange-correlation term for non-ideal plasma simulations. Journal of Physics: Conference Series, 2021, 1787(1):012043 (13pp).

[9] Pearce M, Sparrow Z, Mabote T R, et al. stoBEST: an efficient methodology for increased spatial resolution in two-component molecular tagging velocimetry. Measurement Science and Technology, 2021, 32(3):035302 (14pp).

[10] Malyshkin, Victor. [Lecture Notes in Computer Science] Parallel Computing Technologies Volume 10421 || Parallel Algorithm for Solving Constrained Global Optimization Problems. 2017, 10.1007/978-3-319-62932-2(Chapter 38):396-404.