An Improved K-Means Parallel Algorithm Based on Cloud Computing

Xiaofeng Li[1](✉) and Dong Li[2]

[1] Department of Information Engineering, Heilongjiang International University,
Harbin 150080, China
mberse@126.com

[2] School of Computer Science and Technology, Harbin Institute of Technology,
Harbin 150001, China

Abstract. Through deeply analyzing of the problem in K-Means algorithm, this topic proposed an improved scheme based on Hadoop distributed platform. Using the proposed clustering analysis system to configure the experimental environment, the algorithm is optimized from three aspects: parallel random sampling, parallelization of sample distance computation and parallelization of data clustering process. At the same time, the improved K-Means parallel algorithm flow was described in detail. The experimental result shows that the cluster analysis system based on Hadoop distributed cloud computing platform can provide efficient, stable and configurable clustering analysis service. Improved K-Means parallel clustering algorithm can quickly deal with large scale calculation of cluster analysis.

Keywords: Cloud computing · Hadoop · K-Means · Clustering analysis

1 Introduction

As one of the oldest clustering algorithms, the K-Means algorithm has been invented for half a century. Due to its relatively simple and time-consuming features, the K-Means algorithm has been favored by many researchers [1, 2]. Up till now, the K-Means algorithm has also been active in the field of data mining. For K-Means algorithm, there are many factors affecting its clustering accuracy. However, the most intuitive and significant impact is its input parameter K, which refers to the number of final cluster centers specified by the user, that is, it is divided into several types of data [3]. The change of this value directly determines the accuracy of the algorithm's final clustering result. Therefore, there are many researches on how to initialize the cluster center by users [4, 5]. Among many clustering algorithms, K-Means algorithm is one of the most widely used algorithms, but the algorithm itself still has many problems. In this paper, the traditional serial K-Means algorithm will be used as a starting point to fully study the algorithm flow and characteristics, and conduct parallel optimization based on Hadoop cloud computing platform to solve the problem of its efficiency in the face of large-scale data sets.

Q. Zhou et al. (Eds.): ICPCSEE 2018, CCIS 901, pp. 394–402, 2018.
https://doi.org/10.1007/978-981-13-2203-7_30

2 Traditional K-Means Algorithm

The idea of the K-Means algorithm: First, the user needs to determine the number of clusters of the final clustering result, and then randomly selects the initial cluster center of number K in the original data set. Then, iteratively iterative process requires calculating the spacing of the full amount of data objects to the center of each cluster and merging them into their respective clusters according to the spacing. After all the data points are categorized, the average spacing of the objects in each cluster is calculated, and the original center is replaced with the new cluster center. This iterative process continues until the objective function converges. The convergence of the objective function is that after the end of a classification, the recalculation of the new cluster center does not change, and the algorithm ends.

2.1 Algorithm Equation

It is convenient to describe the improved algorithm of K-Means. This paper introduces the symbol $X = \{x_i \in R^n, i = 1, 2, \ldots, n\}$ to represent the original dataset, $M_1, M_2, \ldots, M_k$ represents the center of K class cluster, $L_1, L_2, \ldots, L_k$ represents a different class of K. The Euclidean distance Equation between two arbitrary data objects is in Eq. (1).

$$d(x_i, x_j)^2 = \sqrt{(x_{i1}, x_{j1})^2 + (x_{i2}, x_{j2})^2 + \ldots + (x_{in}, x_{jn})^2} \tag{1}$$

In the Eq. (1), x_i and x_j are data objects of dimension n. Define the center points of the same class cluster as shown in Eq. (2).

$$M_j = \frac{1}{n_j} \sum_{x \in w_j} x \tag{2}$$

In the Eq. (2), n_j is the number of data objects in the same class cluster. The definition of convergence is shown in Eq. (3).

$$J = \sum_{i=1}^{k} \sum_{j=1}^{n_i} d(x_j, z_j) \tag{3}$$

The target function requires the user to enter the specified parameters, R and z, when the number of data objects contained in the spherical cluster with radius R exceeds the value z, the current region is considered as a high-density area, whereas the other is the low-density region.

2.2 Problems Existing in K-Means Algorithm

1. The traditional K-Means algorithm is a stand-alone operation algorithm, which is limited by the hardware of a single machine, and the algorithm cannot adapt to the growing clustering of massive data.

2. The traditional K-Means algorithm uses a completely random selection strategy to initialize the clustering center point, which not only affects the accuracy of the algorithm, but also reduces the efficiency of the algorithm.
3. In order to ensure the accuracy of the replacement cluster center operation, the traditional K-Means algorithm uses the global sequence to replace the cluster center, but such coarse-grained operations increase the time complexity of the algorithm and thus affect the execution efficiency.

3 Improved Scheme of K-Means Algorithm

3.1 Parallel Random Sampling

The calculation of the traditional K-Means algorithm uses a full amount of data objects, which is very inefficient in the face of very large-scale data sets. In order to reduce the time consumption of the algorithm, this paper designs a preprocessing operation for initializing the clustering center, i.e., pre-sampling processing. In order to improve the efficiency of K-Means algorithm, a parallel random sampling process based on Top K processing is designed. And the parallel process is based on Hadoop distributed system. The parallel process algorithm is based on Hadoop distributed system is as follow:

Input: the random number range H, the sample data capacity N, and the number R_n of Reducer.

Output: N sample data samples.

1. In the Map phase, the total amount of data object is assigned, the value range is H, and the random value is key, the data value is value, and the key value is output.
2. The output results are sorted internally, each Reducer outputs a sorted previous N/R_n data.
3. The sample is preprocessed to get the initialization cluster center point. The preconditioning Eq. (4) is defined as follows:

$$V_j = \sum_{i=1}^{n}((\sum_{i=1}^{n} d_{i1}) - d_{ij}), j = 1, 2, \ldots, n \tag{4}$$

3.2 Parallelization of Sample Distance Calculation

The K-Means parallel algorithm is based on the independence of elements. The traditional K-Means algorithm calculates the distance of a full data object in a circular manner. Therefore, the distance calculation process is parallelized. In the Map Reduce parallel computing framework, Map plays a major role in mapping. Therefore, this paper considers the use of the mapping function of the Map stage to map the full amount of data in the form of <key, value> to different Reducers for parallel clustering calculations, and the Reducer in this case is different K clusters so as to make full use of the independence of the original data objects and parallel cluster analysis [6]. After parallelization at the Map stage, multiple nodes can simultaneously calculate the sample distance and speed up the algorithm operation efficiency.

3.3 Clustering and Parallelization of Data Object

After the mapping of the Mapper function, the data objects are mapped to the respective cluster Reducer according to the distance. Because each cluster corresponds to its own Reducer, the reducer parallelism is set to K. At the Reducer stage, it is necessary to iteratively calculate the center point of clusters, replacing the initial center point that was originally calculated based on parallel random sampling. The calculation rule here is the sum of the squares of the Euclidean distances of the full data objects in the cluster, and the minimum point is chosen as the new center point.

At the first stage of execution, each cluster corresponds to its own Reducer, and the parallel data strategy is performed sequentially on the entire data object in the cluster. First, all data objects are taken as input data sets, and then any data object is selected as the center point of the temporary clusters. The sum of the squares of the Euclidean distances from other elements in the class to the current center point is calculated, and the least squared point and the numerical minimum point are selected as the new center point.

In this paper, the minimum Euclidean distance is calculated, and the characteristics of kv structure are optimized by using the Map Reduce distributed computing framework. In the key value pair, key implements the compareTo() method of interface Writable Comparable. compareTo() can compare the numeric size between elements, so that it can be sorted [7]. Therefore, the iterative calculation of the comparison process of cluster center point steps can be realized by using the distributed sorting function of kv structure.

4 Implementation of Improved K-Means Parallel Algorithm

Through in-depth analysis of the characteristics of the traditional K-Means algorithm, this paper studies and implements an improved K-Means parallel algorithm based on the Hadoop clustering analysis system. The algorithm is optimized from the three directions: parallel random sampling, parallelization of Mapper, and Parallelization of Reducer. At the Mapper stage, the parallelization of the sample distance calculation is improved, and the data object clustering process and the Euclidean distance sorting are improved at the Reducer stage. The specific execution process of the algorithm is as follows:

1. The user enters the original data set with the final cluster number K and data size n. The output condition is that the objective function converges, i.e., the Euclidean distance at the center of each cluster is less than the threshold.
2. The original data set is processed by the Top K-based parallel random sampling. After the sample is preconditioned by Eq. (4), the center point of the cluster is initialized.
3. The data serial number is used as the key, and the distance calculation Eq. (1) is used to calculate the Euclidean distance for each data point.
4. Map the entire data object to its own classifier Reducer using the parallel mapping at the Map stage. This process requires the intermediate file storage of the HDFS distributed file system.

5. In the Reducer, the sum of the squares of the distances of each cluster is calculated in parallel to calculate the new cluster center.
6. Determine whether the Euclidean distance of the current cluster center is greater than the threshold. If yes, Replace the center point of the original cluster with the center point of the current cluster and return to step 3 to recalculate, otherwise the algorithm ends.

The full data set first undergoes parallel random sampling and preprocessing before performing clustering calculations. The sample distance calculation and data classification of cluster analysis are performed by MapReduce. Compared with the traditional K-Means serial algorithm, this improved algorithm parallelizes the cluster analysis process, which makes the efficiency of the algorithm greatly improved when running large-scale data.

5 Experimental Analysis and Results

In the environment of cluster analysis system, the design experiment of the improved k-means parallel algorithm is combined with the experimental results. Firstly, the experimental environment and data preparation of cluster analysis system are introduced. Then, the traditional k-means algorithm and k-means parallel algorithm are compared experimentally from the four directions of convergence speed, accuracy, initial sampling rate and acceleration ratio in the cluster environment. Finally, the improved algorithm is analyzed and summarized.

5.1 Experimental Environment and Data of Cluster Analysis

In order to simulate the distributed cluster environment in real situation, six PC computers were used in cluster analysis system experiment environment. The operating system is Cent OS6.4. Software Java_1.7.0_79, Zookeeper-3.4.5, Hadoop2.6.0 and HBase 0.96.2 were installed respectively.

Because the experiment needs the accuracy of the test and the speedup in the cluster environment, two data are prepared. One is the Iris open source dataset commonly used for cluster analysis, and the other is a large-scale dataset generated. There are three classic Iris datasets, each with a data capacity of 50, and each data object contains four different attributes. Due to the small capacity of the Iris dataset, it is impossible to test the improvement effect of the algorithm in large-scale clustering. Therefore, the attribute dimensions and the capacity of the Iris dataset are increased, and a large-scale random dataset is constructed with code. In this experiment, five sets of data sets with different sizes are generated. Each set of data is divided into three clusters, and the number of elements in each cluster is the same.

Table 1. Generated random data set

Data set file	Data set size	Total number of data elements	Data dimension	Cluster center point number
File A	0.2 M	8000	5	3
File B	150 M	8000000	5	3
File C	450 M	24000000	5	3
File D	1.3 G	64000000	5	3
File E	2.2 G	150000000	5	3

5.2 The Convergence Speed Comparison

The convergence speed comparison experiment is to compare the number of iterations required for running the algorithm when computing the same data set in a stand-alone environment, comparing the traditional K-Means algorithm and the improved K-Means parallel algorithm.

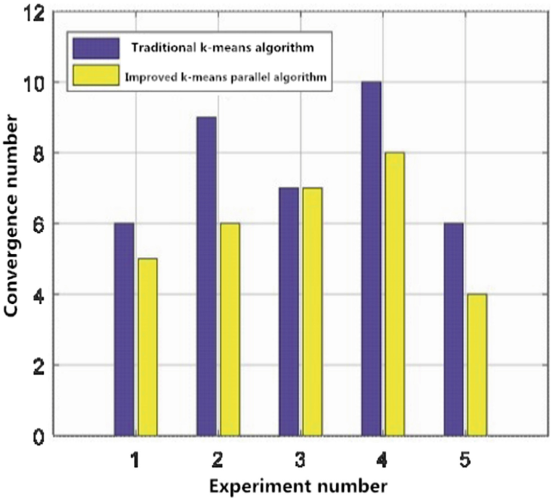

Fig. 1. Convergence performance comparison

In order to eliminate the interference of parallel computing, the traditional K-Means algorithm runs in the common stand-alone environment, and the improved K-Means parallel algorithm runs in the pseudo-distributed mode. The above two algorithms are run on 5 machine nodes with File A as the original data set. The test data is shown in Fig. 1.

The experimental results show that the improved k-means parallel algorithm has fewer average iteration times in the single-machine pseudo-distributed mode, so it has better convergence. The reason that the algorithm converges faster is that the pretreatment process makes the initial class cluster center more accurate than the traditional algorithm.

5.3 Accuracy Comparison

The purpose of the accuracy comparison experiment is to test the accuracy of traditional K-Means, mahout K-Means algorithm and improved K-Means parallel algorithm for standard Iris data clustering, where, the mahout K-Means algorithm is the K-Means parallel algorithm implemented by Hadoop platform. The clustering effects of the three algorithms are shown in Tables 2, 3 and 4.

Table 2. Traditional K-Means

	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	39	11
Virginica	0	11	39

Table 3. Traditional K-Means

	Setosa	Versicolor	Virginica
Setosa	49	1	0
Versicolor	1	37	12
Virginica	0	12	38

Table 4. K-Means parallel algorithm

	Setosa	Versicolor	Virginica
Setosa	50	0	0
Versicolor	0	43	7
Virginica	0	7	43

In Tables 2, 3 and 4, the total number of Iris data sets is 150, and the traditional K-Means calculation is accurate 128, and the accuracy rate is 85.3%. The mahout K-Means algorithm is accurate 124 and the accuracy rate is 82.7%. The improved K-Means parallel algorithm is accurate 136 and the accuracy is 90.7%. Therefore, the experimental results show that the improved k-means parallel algorithm has better accuracy. After analysis, this result is caused.

5.4 Initial Sampling Rate Comparison

This experiment is to compare the operation efficiency of several different random sampling methods. The sampling methods of the comparison are sequential traversal, byte offset, and parallel random sampling based on Top K improvement. The k-means parallel algorithm runs on 6 nodes. In this experiment, the File B File is the original data set, and the timeout period is 1 h. The sampling time of each method is shown in Table 5.

Table 5. Time comparison of different sampling methods

	The number of elements in the sample data set			
	90	900	900000	9000000
Line-by-line through	461.2 s	2605.1 s	Timeout	Timeout
Byte offset	1 s	9.6 s	624.1 s	Timeout
The parallel sampling	32.4 s	32.5 s	43.1 s	52.1 s

5.5 Cluster Environment Speedup Over Validation

Due to improved algorithm is a kind of parallel algorithm design in this paper, the speed ratio is a parallel algorithm is one of the most intuitive indicator of fine performance, so the improved k-means algorithm is used to speedup ratio the experiment. The speedup ratio is the ratio of the same task running in a different number of processors. The formula is defined as follows:

$$S_p = \frac{T_s}{T_p} \tag{5}$$

In Table 1, there are 5 orders of magnitude of different artificial data sets, including File A, File B, File C, File D and File E as the original input data The speedup ratio of parallel algorithm is calculated by using 1, 2, 3, 4 and 5 computing nodes respectively. The speedup ratio is shown in Fig. 2.

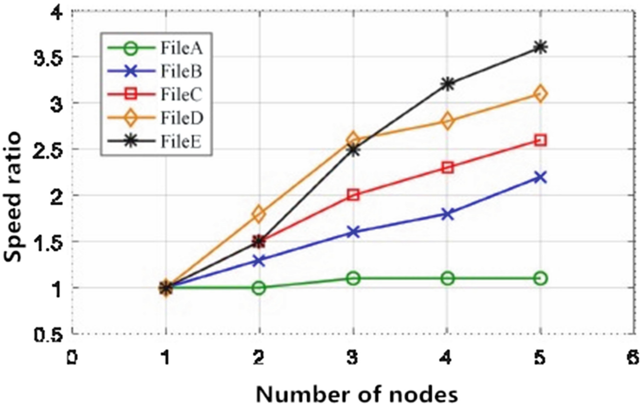

Fig. 2. Speedup ratio test

From the experiment result shows that all data set speedup ratio increases with the increase of computing nodes, the speedup increases with the increase of data amount. It shows that the improved K-Means algorithm in the distributed cluster parallel environment can significantly improve the operation efficiency, and can adapt to large-scale data set of cluster computing.

6 Conclusion

This paper first analyzes the process and existing problems of k-means algorithm. Then, the improvement scheme of k-means algorithm is studied, which mainly includes parallel random sampling, sample distance computation parallelization and data object clustering process. In this paper, the improved k-means parallel algorithm is tested in four directions from the convergence speed, accuracy, the initial sampling rate and the clustering environment speedup ratio. The experimental results show that the clustering analysis system was designed and implemented in this paper can efficient and stable distributed clustering services, improvement of K-Means parallel algorithm has good convergence and accuracy, initialization, sampling rate and the speedup ratio of the cluster environment.

References

1. Deng, Q., Yang, Y.: Research on improved parallel K-means algorithm based on Spark framework. Intell. Comput. Appl. **8**(01), 76–78 (2018)
2. Li, X., Yu, L., Lei, H., Tang, X.: A parallel implementation and application of K-means improved algorithm. J. Univ. Electron. Sci. Technol. China **46**(01), 61–68 (2017)
3. Li, H.: Improved K-means clustering method and its application, pp. 15–17. Northeast Agricultural University (2014)
4. Li, G.B., Han Qing, J.: An improved K-means clustering algorithm for MapReduce parallelization. Digit. Technol. Appl. (12), 134–136 (2016)
5. Lu, S., Wang, J., Zhang, X., Gao, J.: Optimization of K-means clustering algorithm based on Hadoop platform. J. Inner Mongolia Univ. Sci. Technol. **35**(03), 264–268 (2016)
6. Ran, J., Kou, C., Liu, R.: Efficient parallel spectral clustering algorithm design for large data sets under cloud computing environment. J. Cloud Comput. Adv. Syst. Appl. **2**(1), 1–10 (2013)
7. Fu, C., Zhou, G.: Improved parallel sorting algorithm based on Hadoop. Softw. Guide **15**(4), 68–70 (2016)