An Improved K-means Algorithm based on Cloud Platform for Data Mining

Bin Xia*[1], Yan Liu[2]

[1.] *School of information and management science*, Henan Agricultural University, Zhengzhou, Henan 450002, P.R. China
[2.] College of Information Engineering, Zhengzhou Engineering and Technology College, Zhengzhou, Henan 450044, China

***Abstract* — Data mining allows users to make effective use of data in wide applications guided by their specific scientific research and business decisions. But with the enormous quantity of information embedded within data, traditional data mining algorithms face greater challenges in terms of ever increasing processing time or being unable to deal with massive data. Migration of traditional algorithms to cloud platforms for parallel processing is a first effective step to solve the problem. The K-Means algorithm of Hadoop platform in data mining is normally used with parallel processing ability to achieve time improvement. Prior to clustering in the K-Means algorithm, we sampled the data to determine the initial center point using neighborhood density and clustering. Based on the analysis of the limitation of the algorithm, we propose an improved K-Means version based on density and sampling (BSDK-Means). Determination of the initial K value and the center through sampling and density, we meet the users' needs to specify a K value and initial defects of center point in the initial stage. The improved K-Means algorithm MapReduce, uses Hadoop parallel processing ability to enhance the scalability of the algorithm. Our experiment shows that the algorithm has better scalability.**

Keywords - Hadoop; data mining; K-Means; MapReduce

I. INTRODUCTION

The emergence of data mining the user data to be used, and with the guidance of the scientific research and business decisions, but because of the enormous amount of information, traditional data mining algorithms have faced greater challenges: data mining algorithms cannot make traditional mass data processing or treatment to spend a lot of time[1]. How to data mining in massive data has become one of the hotspot of current research [2]. The use of high performance computer and parallel computing can be greatly solve the problem [3], for parallel computing can provide computing power needed to process mass data, and with the increase of data, can be used to increase the computing power by adopting cluster. So the research on parallel data mining algorithm is of practical significance.

The allocation of resources dynamic, parallel computing functions of cloud computing is unable to process mass data and provides a solution to the traditional data mining. Massive data can be stored through the cloud platform, users can access the data through different ways, and mining computation ability needed data through cloud platform on demand for. Cloud platform provides conditions for data mining, how will the cloud platform and traditional data mining are combined, the key is how to improve the traditional parallel algorithm, make use of cloud platform of mass data processing.

II. IDEA OF K-MEANS ALGORITHM

K-Means[4] algorithm is a clustering algorithm James proposed by MacQueen in 1967, the algorithm is simple, high efficiency, has been widely used in scientific research and industrial application.

The basic idea of the algorithm: K-Means algorithm is a cluster analysis algorithm, the n sample is divided into K clusters that objects within a cluster have high similarity, and between the clusters of elements with low similarity [9].

The user decides to cluster number k, and K were randomly selected points as the initial center point, every initial center point as a cluster; and then according to the distance formula or other similarity calculation formula of other points in the sample will be divided to the nearest cluster; and then calculate the average of all the objects in the cluster as the center point of the new[6,7]. It is repeating iteration until the objective function convergence. The characteristics of each iteration of K-Means algorithm will calculate the sample point which is assigned to the nearest cluster center, if the allocation error, you need to adjust to the corresponding cluster center, distribution right, there is no need to adjust.

A. Algorithm Definition

Definition 1: Clustered data set:

$$X = \{x_1, x_2, x_3, \cdots, x_n\} \tag{1}$$

Among them, X denotes the n data points, each data point is a dimensional data.

Definition 2: The similarity formula: here is a selection of Euclidean distance formula as the similarity calculation formula.

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

Among them, x_i 、 x_j are n dimensional data points.

Definition 3: cluster center

$$z_i = \frac{1}{m}\sum_{x\in\omega_i} x \qquad (3)$$

Among them, ω_i represents the i cluster, m represents the number of data points that belongs to ω_i cluster.

Definition 4: Convergence condition

$$E = \sum_{i=1}^{k}\sum_{p\in\omega_i}\left|p - m_i\right|^2 \qquad (4)$$

Among them, E is the sum of the square error of all objects, p is a space point, m_i is the average value of the cluster ω_i.

B. *Algorithm Process*

The K-Means algorithm uses partition criteria, such as the distance formula, the data is divided into k clusters. Data similarity within clusters is high, the lowest similarity between clusters. The main steps shown in Tab. 1:

TABLE I. THE MAIN PROCEDURES OF K-MEANS

Input: cluster number k, the initial center point and to divide the data
Output: K cluster members.
Step: 1) Randomly selected K objects as the cluster center; 2) Calculation of other objects and cluster center distance, the object is divided into the corresponding cluster center; 3) According to the cluster object, recalculate the cluster center; 4) Determine whether the cluster center change and iteration number is less than the threshold, such as changing jump to step 2; 5) Judge whether the number of iterations is less than the threshold, if less than the threshold then the output member of the K group, otherwise output cluster failure; 6) Execution.

C. *Aalgorithm defects*

The K-Means algorithm is very clear, the algorithm is simple and easy to realize. The complexity of the algorithm is similar to $O(tkn)$. Where t is the iteration number, n is the sum of the classification data, k is the number of packets. Typically $k < n$, $t < n$,so complexity of the algorithm is similar to $O(n)$.

Disadvantages:

- The K-Means algorithm depends on the K value set
- The K-Means depends on the initial cluster center
- Outliers sensitivity
- Scalability

Complexity of the K-Means algorithm is approximate to $O(n)$, but in the face of large amounts of data, computing the number of increase, the similarity calculation time has become very time-consuming. Therefore the use of parallel computing is essential in the case of a large amount of data.

III. IMPROVED K-MEANS ALGORITHM BASED ON DENSITY AND SAMPLING

In this paper, according to the former analysis of the defect of K-Means algorithm, this paper proposes an improved K-Means algorithm based on density and sampling (BSDK-Means）. Determination of the initial K value and the center through the sampling and density, to solve the user needs to specify a k value and initial center point defects in the initial stage.

A. *Concepts*

Definition 1(neighborhood): For any point in space P, radius, to the point of P is the center of a circle, which is the radius region, called the neighborhood of P.

Definition 2(density): For any point in space P, a number of points in the P neighborhood are called the density of P.

B. *Parallel improvement*

BSDK-Means algorithm mainly include 4 parts:

- multiple sampling of massive data;
- using density, find the center point of the sampling data;
- confirm global center;
- To cluster the data using the K-Means algorithm.

Multiple sampling is carried out through the acquisition of huge amounts of data, generating massive data form can reflect sample. For the sampling data, calculating between data points and data points to determine the data belonging to the neighborhood, and according to determine the sample center point neighborhood density, according to the global center of original data to determine the sample center point. Center point is determined by sampling and density, which solves the defects in the original K-Means algorithm, depends on the initial center points. Specify center point, the data clustering using K-Means algorithm. In the large amount of data, calculating the distance between the object and the computing center of the cluster is the most time-consuming operation, operation time and increases with the increasing of data size. So there will be BSDK-Means algorithm that is transplanted to Hadoop platform, operation ability to handle the most time-consuming calculation using parallel Hadoop platform.

The detail flow chart as shown in Fig. 1, from the diagram can be seen, for the center point and the clustering of the two steps using the characteristics of Hadoop, the realization of parallel.

1) *Determine the data center point based on sampling and density*

Sampling density and confirm the center point can be executed in sequence through serial mode, but for a large number of samples and sampling number of circumstances, the process is time-consuming; and sample confirmation center is of no relevance. Therefore, we use the Hadoop platform parallel processing of massive data capacity optimization; improve the speed of the sample center point.

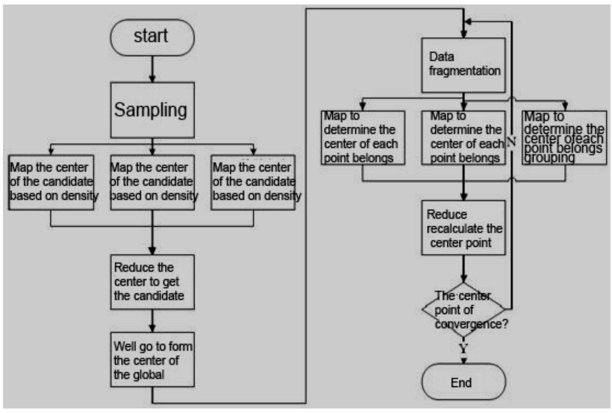
start
Sampling
Map the center of the candidate based on density
Map the center of the candidate based on density
Map the center of the candidate based on density
Reduce the center to get the candidate
Well go to form the center of the global
Data fragmentation
Map to determine the center of each point belongs
Map to determine the center of each point belongs
Map to determine the center ofeach point belongs grouping
Reduce recalculate the center point
The center point of convergence?
N
Y
End

Figure 1. BSDK-Means Flow Chart

The Hadoop platform assigned sample to perform different node, each node implementation of calling a custom Map function to calculate the generation of candidate points of the sample. Finally, the reduce operation is performed on the candidate points are generated, satisfy the condition (the neighborhood radius density >density) candidate center point. In accordance with the thought, design of class Sample Map, class Sample Reduce.

The Sample Map class is a concrete realization of Map operation. Map operation for default input on, here Key value for the current row offset to the start line, Value as a node of x coordinate information. In Map operation, the calculation of point X and the candidate point distance, if all distances are greater than radius, it will point x as a new candidate, otherwise it will point the x information is added to the x distance is less than radius and the candidate points. The final output of the candidate points of .The main steps shown in Tab. 2.

TABLE II. CLASS SAMPLEMAP MAJOR STEPS

Input: the starting offset key, node coordinate information for x value
Output: the candidate center point identifier key' , candidate center point $value'$
Step: 1) calculation of X between nodes and each candidate point distance; 2) if the distance is less than radius, then x will point you coordinate is accumulated to a candidate, and the candidate point density increase; 3) if all distances are greater than radius, point to X as a candidate new points; 4) generates all candidate center point, build string representation of candidate center point you coordinate, candidate center point hash value as key' , a string containing the candidate center neighborhood interior point each coordinate, cumulative and density as $value'$; 5) output $< key', value' >$, then end algorithm.

The Sample Reduce class is a concrete realization of Reduce operation. The Reduce operation is default input on, and the Key value is the identifier of candidate points, the V value is intermediate with the same Key value set. The Reduce function according to the density of set value judgment candidate is qualified (greater density value set), the center point of the output of qualified. The main steps are as shown in Tab. 3.

TABLE III. SAMPLEREDUCE MAJOR STEPS

Input: candidate center point hash value key, intermediate with the same key value V
Output: the candidate center point identifier key' , candidate center point $value'$
Step: 1) determine candidate center point density value whether is greater than the set density value 2) If greater than then calculate the new center in the field, the new center point identifier as key' , the new center point as the $value'$. Output $< key', value' >$; 3) If less than then discard the candidate center point; 4) The algorithm terminates.

The candidate center multiple samples Reduce output is generated, the candidate centers of different samples produced by point may belong to the same neighborhood, so merge of candidate center is necessary. Merging concept is also based on the global center point density of original data.

2) Using the K-Means algorithm to produce clustering

Use the K-Means algorithm to divide the recalculation clustering in large amount of data which mainly produces in between data and the center point and the distance calculation of the center point. Here the distance calculation operation is assigned to each of the Hadoop platform implementation of the node, the data point and the center point of the distance calculated by the implementation of the node, and the point into the minimum distance cluster. And the center point recalculation completed by the Reduce operation, in the Reduce implementation of the node re compute cluster center. According to the idea, design of class KMeansMap and class KMeansReduce.

The KMeansMap class is a concrete realization of Map operation. Calculated for each data point and the center point of the distance in the stage of Map, calculate the shortest distance, and the data points assigned to the distance from the center of the nearest point. The main steps are shown in Tab. 4.

TABLE IV. KMEANSMAP CLASS MAJOR STEPS

Input: the starting offset key, node coordinate information for x value
Output: the group number key' , node X coordinate information $value'$
Step: 1) the first implementation of the global center needs to read from HDFS, stored in the global variable space; 2) the calculation of x and the global center distance, find the minimum distance, determine the center point x belongs to; 3) index belong to the center point as the group number key' , node x coordinate information as $value'$; 4) Output $< key', value' >$; 5) The algorithm terminates.

The KMeansReduce class to get each cluster of data points to calculate each group center, its main steps is shown in Tab. 5.

TABLE V. KMEANSREDUCE CLASS MAJOR STEPS

Input: group index, nodes belonging to the group
Output: group index as key' , the new center point as the $value'$
Step: 1) nodes belong to the same group of cumulative index each coordinate, calculate the dimensional coordinate average value, average value as the new center point to sit standard; 2) group index as key' , the new center point as the $value'$; 3) Output $< key', value' >$; 4) The algorithm terminates.

New points Reduce production, if the new center and the center point of a wheel change is less than the threshold, then the algorithm ends. If more than the threshold, the new center point as the initial center point, for the next cluster.

The convergence condition of K-Means algorithm is usually a square error criterion, defined as formula (5):

$$E = \sum_{i=1}^{K} \sum_{p \in C_i} |p - m_i|^2 \qquad (5)$$

E is the sum of square error of all objects, p is space, m_i is the average value of group C_i . In large data, the computing time square error cannot be ignored; this criterion is not suitable for large data of convergence. In view of this situation, modify the convergence conditions for the similar two times the distance from the cluster center. Its definition as the formula (6).

$$E = \sqrt{\sum_{i=1}^{k} |p_i - p_i'|^2} \qquad (6)$$

p_i as the center point, the new center point p_i' to a corresponding p_i clustering.

C. Complexity analysis

The original K-Means is run on a single node, its complexity is $O(tkn)$. The proposed BSDK-Means algorithm based on Hadoop platform, using Hadoop parallel programming capability computing tasks will be assigned to the p node execution, its complexity is $O(t_1 + n \times t' \times k / p)$. t_1 said the center point of the sample to determine the cost of time, the amount of data in the case, the sampling to determine the center point of time is negligible. The t' is an improved iterative times, through the experimental test of $t' < t$.

IV. EXPERIMENT

The experiment mainly compares the K-Means parallel algorithm and BSDK-Means algorithm in the operation of multiple sets of data, from the clustering result, convergence time and speedup the three aspects of the analysis of the test results.

The experimental data consists of two parts, testing the clustering results using Edgar Anderson iris (Iris) data, and test the convergence time and speedup using artificial data, D0 (including 200000 data), D1 (including 500000 records), D2 (including 800000 records), D3 (including 1000000 records), D4 (including 1200000 records).

Because the K-Means algorithm relies on the K and initial center point, therefore using the scoring algorithm, each group data repeating the experiment 10 times, get rid of the worst and the best record, the remaining 8 records for the average. While the BSDK-Means algorithm depends on the density, the neighborhood radius, literature [8] concluded that $k < \sqrt{n}$, density $\geq \sqrt{n}$. The neighborhood radius depends on the specific data space. In the course of the experiment set density is $\sqrt{n}$, the neighborhood radius is $2\sqrt{n}$. After repeated experiments, the density and the neighborhood radius value with the experimental results very well.

A. Analysis of clustering results

The iris data set (Iris dataset) is iris Anderson research Canada Jasper peninsula on the geographic variation of data [5] presented flowers, which contains 150 samples. In the 150 samples, including three kinds of iris, respectively (Iris setosa) is a mountain of iris, iris versicolor (Iris versicolor) and Virginia (Irisvirginica) of iris. Each sample has four attributes, respectively is the length and width of sepals and petals, so the data sample matrix representation can be used 150 long 4.

The choice of the iris as a test of the original K-Means algorithm and BSDK-Means algorithm to data sets, the reason is that the 150 samples have been very determined and divided into three categories, and the clustering central points clear, central location point respectively (6.588, 2.974, 5.552, 2.026), (5.006, 3.418, 1.464, 0.244), (5.936, 2.77, 4.26, 1.326).

Tab.6 gives the original K-Means algorithm and BSDK-Means algorithm implementation results in the data set of iris flower. From the table we can see that the original K-Means algorithm misclassification sample number is 20, the success of the sample number is 130, while the BSDK-Means algorithm misclassification sample number is 14, the success of the sample number is 136. The improved algorithm is lower 4% classification error rate than the original algorithm, better clustering effect. The BSDK-Means algorithm to select the initial point is determined according to the sampling and density than the random initial point, confirm the more targeted, so it has more accuracy.

B. Running time

The running time of algorithm execution speed is used to judge. From the algorithm itself, the K-Means time is mainly consumed in the data packet, and the BSDK-Means time is produced by the two part centers and data packet composition. More in the original data, BSDK-Means algorithm is time consuming more in a data packet. In order to better illustrate the algorithm itself is time-consuming, ignored here Hadoop communication node time-consuming, ignore the different node performs the same time error data, using the iteration number to measure the algorithm execution speed.

TABLE VI. THE CONTRAST OF THE ORIGINAL K-MEANS ALGORITHM AND BSDK-MEANS ALGORITHM RESULTS

Clustering Algorithm	Error classifi-cation number	Error LV	Clustering center	Error
K-Means	20	13.3%	c1=(5.0038,3.4141, 1.4768,0.2545)	0.2680
			c2=(5.8799,2.7631, 4.3672,1.3873)	
			c3=(6.7820,3.0384, 5.6568,2.0435)	
DSDK	14	9.3%	c1=(5.0064,3.4020, 1.4962,0.25239)	0.184
			c2=(5.9362,2.8134, 4.3910,1.4082)	
			c3=(6.6359,3.0147, 5.501,2.0132)	

From the analysis of the algorithm, under the same conditions of parallel K-Means algorithm and BSDK-Means algorithm execution time depends only on the initial center point. Set the number of Hadoop platform node 4 in this experiment, and the test of D0, D1, respectively D2, D3, D4 data set, obtained the results as shown in Fig. 2. As you can see in Fig. 2, the BSDK-Means algorithm and the parallel algorithm of K-Means iteration is generally increased with the increasing amount of data, but the BSDK-Means algorithm is better than the parallel K-Means iteration number. Because the BSDK-Means algorithm based on sampling and density to confirm the initial point, than the random selection of more targeted, so can be faster convergence.

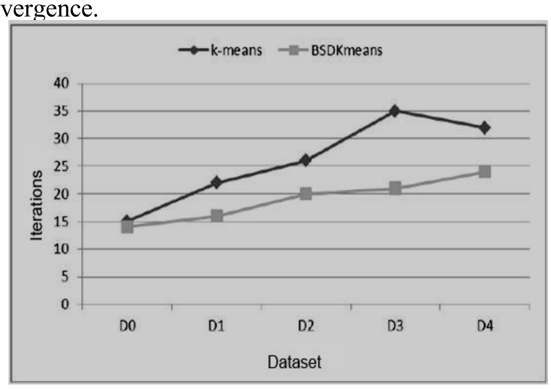

Figure 2. Figure Algorithm Running Times

C. Speedup

The speedup ratio refers to the ratio of the execution time of task execution time in a single treatment with multi-processor, commonly used to measure the performance of parallel programs and effect, definition as formula (7).

$$s_n = \frac{T_1}{T_n} \tag{7}$$

T_1 is for the task execution time on a single processor, T_n is for task execution time in n processor. Experiments were to test the D0, D1, D2, D3, D4 data sets in different nodes on the execution time. Fig. 3 is a parallel K-Means algorithm speedup, Fig. 4 BSDK-Means algorithm speedup. As can be seen from the graph on the platform of Hadoop BSDK-Means algorithm and K-Means algorithm have good speedup, but more data, the speedup is greater. But as the number of nodes increases, the speedup increases flattening algorithm. Since the nodes increases, the inter node communication consumption increased, resulting in accelerated than incremental gentle.

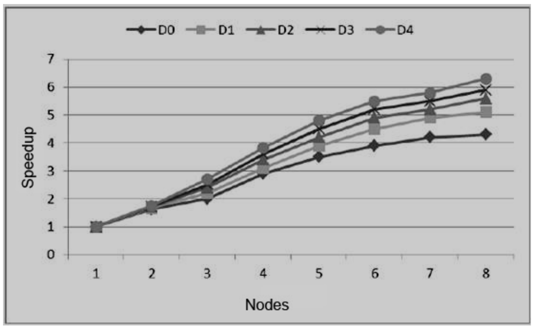

Figure 3. parallel K-Means algorithm speedup

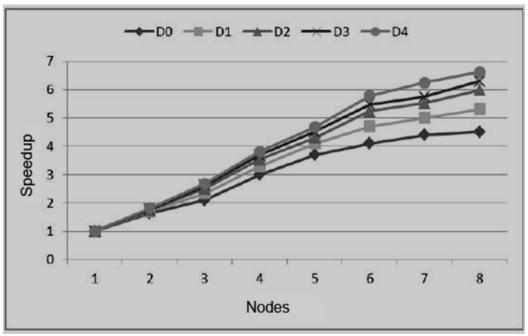

Figure 4. BSDK-Means algorithm speedup

V. CONCLUSIONS

This chapter from the start with the K-Means algorithm, according to the clustering result depends on the K value of defects and initial center point, put forward the improved clustering algorithm BSDK-Means sampling, density and based on Hadoop platform[10]. The BSDK-Means algorithm keeps the advantage of the original K-Means algorithm, to select the initial center point by density; the algorithm does not depend on the K value and the initial center point. Finally, through different sets of data carries on the experiment to the algorithm, we conclude that BSDK-Means algorithm has better convergence and acceleration.

CONFLICT OF INTEREST

The author confirms that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work was financially supported by the Henan Science and Technology Key Project Foundation (122102210507).

REFERENCES

[1] Ma S and Wang TJ and Tang SW, "A fast clustering algorithm based on reference and density. "Journal of Software, pp.1089-1095,Jul, 2003.

[2] Jeffrey Dean,,"Simplified Data Processing on LargeClusters", Google.Inc.

[3] MJ Ruo.and Y Ge, "Member. Shared Memory Parallelization of DataMining Algorithms Techniques. Programming Interface. And Performance,"IEEE Transactions on Knowledge and Data Engineering,pp.71-89,Jan,2005.

[4] J.MacQueen,Some methods for classification and analysis of multivariateobservations,1967,pp.28 1-297.

[5] Edgar Anderson,The irises of the Gasp Bulletin of the American Iris Peninsula,Society, pp. 2-5,1935.

[6] ShL Yang and YS Li, "K-Means algorithm of K value optimization problem of," systems engineering theory & practice, pp.97-102,Feb,2006.

[7] Edgar Anderson,The irises of the Gaspé Peninsula. Bulletin of the American Iris Society,pp. 2-5, 1935.

[8] XF Lei and KQ Xie and F Lin, "An emcient clustering aIgorlthm based on locaI optimalityof K-Means," Journal of Software, pp.1683-1692,Jul,2008.

[9] ZhP Zhang and AJ Wang, "Method for initializing K-Means clustering algorithm based On breadth first search," Computer Engineering and Applications, pp.159-161,2008.

[10] Apache Software Foundation. ambari, http://incubator.apache.org/ambari/, 2013.