An Improved Parallelization of K-means Algorithm based on HADOOP

Yizhuo Guo
Department of Information Engineering, Heilongjiang International University, Harbin 150025, china
610984970@qq.com

ABSTRACT In order to improve the problem that the single-machine serial programming model is not ideal for mass data clustering, we combine big data technology with text clustering related technologies. Implement distributed storage and calculations for text data, parallelization of text vectors and parallel clustering using clustering algorithms based on the Map Reduce programming model. The traditional k-means clustering algorithm is a typical algorithm for solving clustering problems. It has better with good scalability and scalability for processing large data sets, but the initial center of the algorithm is chosen randomly, and the algorithm is unstable every time. To solve the above problems, firstly, based on the idea of density segmentation and sampling thought, the initial clustering center is selected and optimized. Secondly, parallel sampling of the data set to find the best candidate cluster center by referring to the sample maximum and minimum method search and consolidate data objects in parallel. Finally, the selected initial cluster center is replaced by the central point randomly selected by the k-means algorithm, and the clustering algorithm is parallelized. Experiments show that the improved k-means algorithm can effectively reduce the number of iterations.

1. INTRODUCTION
Cloud computing provides an effective solution to deal with massive data, in which the multiple computers can cooperatively work, and then share virtualized resources by Internet. Hadoop which is the basic cloud computing platform can process distributed large data with reliable and scalable. Compared with the existing distributed computing framework, the service components in Hadoop have their own advantages that have parallel computing capability and high availability in storage capacity. Map Reduce is completely open source which provides a concise programming interface. Based on Map Reduce, the relevant algorithm can optimize in various research areas. The data storage mechanism with HDFS is a multi-copy storage strategy. According to the size of the cluster, the number of copies of text datasets are specifics. When the number of nodes storing a copy of the data is reduced by half, the integrity of the stored data can be maintained, the integrity of the stored data can be maintained. The redundant copy strategy ensures high availability of the file system. In this paper, based on Map Reduce parallel sampling and parallel merging data objects, the algorithm is improved for the disadvantage of randomly selecting the center of K-means algorithm. The initial centroid selected by this optimization method is replaced by the original centroid randomly selected by the traditional K-means algorithm. This proposed method can be processed in parallel, effectively reducing the number of iterations and running time of the algorithm. The proposed parallel random sampling is more suitable for large-scale data sets, which can improve the stability of the algorithm and reduce the selection time of initial clustering centers for large data sets. Compared with the effect
and quality of traditional K-means algorithm, improved k-means algorithm and canopy-k-means algorithm have improved the effect and quality of text clustering. The experimental results show that the proposed method has superiority in clustering efficiency, clustering quality and parallel performance.

2. The Traditional K-means Algorithm

2.1 The Main Idea of Clustering Algorithm

The main idea of K-means clustering algorithm is to randomly select K objects from the data set as the initial clustering center at first. The second step is to measure the distance from the remaining vectors to the clustering center by distance and merge the remaining vectors with the nearest clustering center into the same cluster and generate K clusters. Next step, the distance formula is used to calculate the distance of each class. With K cluster centers (cluster mean) as the new clustering center, recalculates the distance between the remaining objects in the data set and K cluster centers, and repeats the above process until the clustering center and the clustering center generated by the previous iteration does not change (the objective function converges). K-means algorithm is easy to understand. Although the algorithm iteration increases the complexity of the algorithm, the whole implementation process is easier and can make the clustering results to be optimal.

2.2 K-means Algorithm Analysis

Compared with other clustering algorithms, K-means clustering algorithm has the following advantages. The high efficiency of large-scale data processing, a wide range of applications, suitable for image processing, information retrieval and other fields, deal with image features and text types of data sets, optimize the clustering effect and get closer to the real clustering effect. Based on the advantages of the above K-means clustering algorithm, this paper selects the algorithm for text clustering. Of course, K-means clustering algorithm also has imperfections whose defects will affect the clustering effect to a certain extent and steps need to be improved in the traditional clustering algorithm. The initial center of traditional K-means clustering algorithm is randomly selected, which will lead to unsatisfactory clustering results. At the present, the following methods are selected for improving the initial cluster centers.

(1) A method based on the maximum and minimum distance. The distance between the initial clustering centers is as far as possible for the data set selection, and two or more objects with similar distances in the space are avoided.

(2) Based on density weighted method. According to density distribution, the objects in the dataset are assigned and compute the average of data objects. This method has good convergence, but the algorithm is of high complexity and low efficiency.

(3) K-means algorithm is executed many times. This method is inefficient.

In this paper, several classical improved methods are fused. We select candidate clustering centers for sampling data by the maximum and minimum distance algorithm, and then candidate clustering centers which are close to each other are merged. The accuracy of selecting the initial clustering centers are guaranteed at the expense of the efficiency of selecting the centers. Therefore, in order to satisfy the K-means clustering algorithm has better efficiency and accuracy, this paper mainly improves the algorithm from three aspects. The parallel random sampling process, parallel merging data object process, data object clustering parallelization process. The improved k-means parallel algorithm is designed and implemented based on Map Reduce.

3. Improved K-means Parallel Algorithm

3.1 Improved K-means Algorithm Parallel Design

The overall process of the improved k-means algorithm based on Map Reduce parallel processing is shown in Figure 1.
Figure 1. The overall process of the improved k-means algorithm based on Map Reduce parallel processing

The improved k-means algorithm is parallelized. Firstly, by two Map Reduce tasks the initial clustering center is optimized based on the parallel design of Map Reduce. Based on Map Reduce, the specific implementation framework with the selection strategy of the initial clustering center is shown in Figure 2. Because the process of computes the cluster center in iteration is independent by K-means algorithm, the locality of K-means clustering algorithm and the distributed computing of Map Reduce distributed are combined. Based on Map Reduce and the initial centroid, the mean of clustering cluster and the computed cluster are divided to several nodes of the cluster and execute at the same time. The specific parallel framework is shown in Figure 3.
3.2 Parallel Sampling

Based on Map Reduce, the design of parallel sampling is divided into two processes. One task of Map Reduce extracts text vectors by several times in parallel. The set of text vectors is divided into samples...
which containing the same number of vectors and selects a set of text vectors as candidate clustering centers by the maximum and minimum distance algorithm. Another task of Map Reduce is to evaluate whether candidate clustering centers are adjacent according to density parameters, and vectors which meet adjacent conditions are merged. The distributed dense vectors with points contained in a certain neighborhood are merged. The mean value of each merged class is calculated as the final selection of the initial clustering center.

1. Sample extraction

The most common random sampling methods are traverse sampling and byte offset sampling. The traverse sampling can preserve the original data format of the data set, and it is inefficient to sample large-scale data sets. The byte offset sampling is also inefficient to large-scale data sets sample. In order to improve the efficiency of the above two random sampling methods, we present a random sampling method based on Map Reduce in this paper.

In the first random sampling process Map Reduce Job1, the main task is to sample many times in parallel, the initial cluster center for each sample is selected in the sample set, and the selected text vectors form the candidate cluster center set. In the Mapper stage, the input is in the form of <doc, docVector>. The map function is mainly responsible for extracting \( S (S = n/m, n \text{ is the total number of text in the data set, } m \text{ is the number of map tasks}) \) from the data set. First, the input text vector is counted from 1 to select the first \( S \) vector. When the count value \( i \) exceeds \( S \), the selected vector is replaced by the vector used \( S/i \) probability. The sample extraction is finished until all the vectors have been accessed. Then, the selected \( S \) vectors whose form is <null, case Vector>output. In the Reducer stage, the output of the above step is used as input. The selected vectors are merged into the same Reduce because the all keys of the input are empty. The reduce function takes the selected sample vectors as the input of the maximum and minimum distance algorithm. After the algorithm is executed, the clustering center of the sample is selected to <null, tempVector> form output. In addition, the calculation of radius \( R \) is needed to calculate the density parameter of the point when the point is merged near in the second stage. Therefore, in the Reducer stage, the average distance averageDis of candidate cluster centers is calculated and output by<-1, averageDis>. Because the random sampling method used in this paper, the samples are distributed concentration. In this paper, sampled execute many times by above sampling method, that is, Map Reduce Job1 is executed many times.

2. Candidate clustering centers selection

For each sample in the sample set filtrate again to obtain candidate clustering centers. The filtrated method used by the maximum and minimum distance algorithm. The initial clustering center is selected in the sample set which the distance of the data objects is as far as possible. The number of iterations of the algorithm is increased to avoid the near distance of selected objects. This method can be used to firstly filtering of text vector sets.

The idea of the maximum and minimum distance algorithm is to select an object which marks as \( C_i \) from the data set as the first object of the candidate clustering center set. Compared with the distance between the remaining objects in the data set and \( C_i \), using Euclidean distance formula, the object who farthest from \( C_i \) which marks as \( C_1, C_2 \) is used as the second object in the candidate clustering center. The remaining objects which are the minimum distance between \( C_1 \) and \( C_2 \) in the data set marks \( 12\min(d_{1},d_{2}) \). When the object with \( \max(\min(d_{1},d_{2})) \) satisfy formula (1), the object is the third candidate clustering center.

\[
\max(\min(d_{1},d_{2})) \geq t(C_2 - C_1) \quad (1)
\]

3.3 Consolidated Data Object Parallelization

Firstly, the distance matrix \( T \) is obtained by computing the distance between all objects in candidate clustering centers. The mean distance \( R \) of all objects in candidate clustering centers is calculated according to the distance matrix. The density value of all objects in the candidate clustering center is calculated. The density value of the object is the number of objects which is the ball with the radius of \( R \). The density value of all objects is sorted from large to small, and the object with the large density
parameter value is selected. The object and the ball with the radius of Rare merged into clusters which are marked as visited points. The objects with the highest density value are selected from the objects that have never been visited, and the merging process of the above data objects is repeated until all the objects have been visited. The mutually disjoint clusters calculate the cluster center. When the candidate cluster center merges operation finish, the final initial cluster center generates.

In the second processing Map Reduce Job2, the candidate clustering centers are merged in parallel. In the Mapper stage, the output of the previous step is input and <null, tempVector > and <-1, averageDis > are input. A branch condition is whether the map function takes -1. At First, it reads the value of the key -1 and calculates the average value R of all candidate clustering centers. The candidate clustering centers whose key is null are read. The cosine distance between candidates clustering centers are calculated to obtain similarity matrix T which will be initialized. For each component of the corresponding matrix T, if its value is greater than the radius R, the corresponding component reset 1, otherwise the matrix reset component is 0. After the initialization matrix is finished, the density parameters of candidate clustering centers are computed. The number of vectors contained in each vector radius R is counted according to the distance matrix T, and the density parameters are sorted from large to small. The above operations are repeated until all vectors are divided into different clusters. Finally, the center of each cluster is calculated as initial vector. The corresponding text number of the cluster center is Cluster ID, which is the initial center of K-means clustering algorithm.

3.4 Cluster Parallelization
The initial centroid selected in parallel from the text vector set replaces the K centers randomly selected by the traditional K-means algorithm. The distance between the remaining vectors in the text data set and the selected cluster centers are computed. The remaining vectors are classified into clusters represented by the initial cluster centers. The average distance of the clusters is recalculated as the cluster center which is the new cluster centers. The above operations are repeated. The improved clustering algorithm finish when the objective function converges.

Map Reduce Job3 in Figure3 mainly calculates the distance between the remaining text vectors in the text dataset and the selected clustering centers. The remaining text vectors are classified into clustering centers which have the smallest distance. The average distance of the cluster clusters are calculated and generate a new cluster center. Map Reduce Job4 mainly generates clustering results based on the final clustering center.

Map Reduce Job1 is the third task processing procedure which can be summarized as follows. In the Mapper stage, the map function is input as <clusterID, initialVector>. The cosine distance from the text vector to the clustering center is calculated and merges the text vector into the cluster represented by the nearest clustering center. The vectors with small degrees become clusters. The each vector label as <clusterID, (1, docVector)> is output. In the Reducer phase, the main task is to calculate the average value of all the vectors in the cluster. First, the number of vectors in each cluster is counted by using the Combine function, and then the average value of all the vectors in the cluster is calculated by using the distance formula. Then the output is in the form of <clusterID, (num, averageVector)>. Num represents the vectors in each cluster. The average Vector represents the cluster mean. Then, reduce function aggregates the local results of each cluster to determine whether the objective function converges or not. When the algorithm finish, the output is the form of <clusterID, centerVector>. The centerVector is a new clustering center generated after iteration by K-means algorithm. If the objective function does not converge, each cluster center is used as a new initial cluster center point. The above process is repeated.

Map Reduce Job4 is the fourth task processing procedure which only needs to perform key-value conversion in the Mapper phase and output as <cluster ID, doc>. The data processing is not request by Reducer.
3.5 The Parallel Implementation of Improved K-means Algorithm

In this paper, the principle of selecting the initial center points is that the distance between the selected initial center points is as far as possible in multidimensional space. To avoid the local optimal solutions caused by the distance between the center points randomly selected by k-means and eliminate the interference of clustering results. The improved K-means algorithm can be summarized as follows.

(1) Sample size and number of samples are estimated in the text vector set \( D = \{d_i \mid d_i \in R^n, i = 1,2,...,n\} \) generated by MapReduce Job4. Using the method of equal probability sampling, \( D \) is sampled many times to get the sample set \( S_1, S_2,..., S_n \).

(2) A text vector \( d_j \) is randomly extracted from the sample \( S_i \) and used as the first clustering center \( e_1 \). The Euclidean distance from the remaining text vector in set \( D \) to \( d_j \) is computed, and the text vector \( d_k \) farthest \( d \) is selected as the second clustering center \( e_2 \).

(3) For each remaining vector \( d_i \) in the text vector set \( D \), the distance \( d_{i1} \) and \( d_{i2} \) of each vector \( d_i \) and clustering center \( e_1 \) and \( e_2 \) are calculated separately. The minimum values \( \min(d_{i1}, d_{i2}) \) of distance \( e_1 \) and \( e_2 \) are found.

(4) The larger part of \( d_{i1} \) and \( d_{i2} \) mark \( \max(\min(d_{i1}, d_{i2})) \), and the corresponding text vector is \( d_m \).

(5) When the \( n+1 \) vector in the sample set \( S_i \) is selected as the clustering center, the distance between the residual vector in \( S_i \) and the selected clustering center is calculated to find the \( \max(\min(d_{i1}, d_{i2},..., d_m)) \). The corresponding text vector is selected as the \( n+1 \) clustering center if the condition of formula (1) is satisfied.

(6) For each sample in the sample set, step (2)-(5) will repeat until it generates a number of candidate clustering centers.

(7) The density parameters of candidate clustering centers are calculated and sorted in ascending order.

(8) The text vector with the largest density parameter is selected. The vectors in the vector radius merge into a cluster. The visited vectors are marked and select the vectors with the largest density parameter from the vector set that has never been visited. Repeat the above steps until all text vectors are accessed. The mean value of clusters with several disjoint clusters is used as the final initial clustering center.

(9) The selected initial clustering center is considered as the initial center. K-means clustering algorithm is executed.

4. Experimental Results and Analysis

The experimental data in this paper are selected from Chinese text corpus. The corpus is collected from news websites including entertainment, education, military, sports, wealth, tourism, finance and so on. Each category in the text data set contains 2000 texts. Before the experiment, the text in the data set is sorted and numbered again from small to large. The storage form is a key which is customized web text numbering. Web text content is the value of key. It is stored in the distributed file system (HDFS) by value of key. In this experiment, two sets of datasets which recorded as \( S_0 \) and \( S_1 \) respectively from the corpus are used. Five thousand and ten thousand texts of six different categories were extracted.

4.1 Comparison of Iterations Number

The algorithms are applied using Mahout Subproject in Hadoop ecosystem, Canopy-K-means algorithm and the improved K-means parallel algorithm on multi-nodes.

The parameters involved in the calculation process of the improved algorithm are as follows. The threshold frequency of the feature extraction stage is \( (0.1-0.65) \). The K value of the K-means algorithm is set to \( k=6 \). The sample size is \( n = 500, n = 700 \). The data set is extracted five times. The
density check parameter $t=0.5$. Two sets of data sets $S_0$ and $S_1$ are constructed based on Map Reduce. The text vector set $DS_0$ and $DS_1$ obtained by vectorization of the two sets. The time consumption of clustering and the numbers of iterations of each algorithm are recorded. It is shown in Table1 and Table2.

| Algorithm       | Select cluster center | Number of iterations | Total time (s) |
|-----------------|------------------------|----------------------|----------------|
| K-means         | 0                      | 12                   | 678.3          |
| Canopy-k-means  | 77.3                   | 9                    | 529.3          |
| Improved k-means| 98.1                   | 8                    | 478.5          |

Table2. Execution of each clustering algorithm in $DS_1$

| Algorithm       | Select cluster center | Number of iterations | Total time (s) |
|-----------------|------------------------|----------------------|----------------|
| K-means         | 0                      | 25                   | 1518.3         |
| Canopy-k-means  | 181.5                  | 17                   | 1278.4         |
| Improved k-means| 159.2                  | 15                   | 998.5          |

(1) The execution with test datasets of different sizes is shown in Table1 using three algorithms which are traditional K-means algorithm, improved k-means parallel algorithm, Canopy-k-means clustering algorithm. The experimental results show that the number of iterations is most using K-means algorithm. The number of iterations is significantly reduced using other two algorithms because they optimize the selection of the initial clustering center. From the experimental data, we can see that the improved k-means algorithm proposed in this paper has the least number of iterations and total time consumption. It is verified that the improved k-means clustering algorithm based on the initial center optimization selection is more efficient than the traditional K-means algorithm and Canopy-k-means clustering algorithm.

(2) Because more cosine distances between text vectors need to be computed, more text vectors need to be partitioned multiple times. Using three clustering algorithms on the same cluster, when the number of text in the data increases, the number of iterations of each algorithm increases. When K-means clustering algorithm clustering test data sets are containing 5000 and 10000 texts, the changes number of execution cycles of the algorithm are most. Because the selection of the center point of the algorithm is random, the results of each execution of the algorithm are not consistent and the algorithm is more unstable. Using Canopy-K-means algorithm and the improved K-means algorithm, the number of iterations varies slightly and avoid the situation that the selected center point is not ideal because algorithms use optimization scheme to select the center point. Although the selection of the center point also consumes time, it can effectively reduce the number of iterations and the total time consumption of the algorithm.

4.2 Initial Sampling Rate Comparison

The initial sampling rate compares the operation efficiency of several different sampling methods. Sampling methods for comparison are row-by-row traversal, byte offset, and parallel sampling based on Map Reduce. For multi-group data sets, the sampling time is recorded as timeout when the sampling time is more than one hour. The comparison of sampling time between different random sampling methods for different size data sets is shown in Table 3.

As shown in Table 3, with the increase of the size of the sampled data set, the time consumed by the method of traversing the data line by line is very long. This sampling rate is the lowest. When sampling a small data set, the byte offset sampling method consumes the least time. The sampling time
increases linearly with the increase of the amount of data because of limitation of the time complexity. When processing large-scale data sets, this method is not applicable. When small data sets are processed based on Map Reduce parallel random sampling method, the sampling rate is lower than byte offset sampling. Because the start-up of Map Reduce tasks and intermediate steps affect the sampling rate, large-scale data, task processing parallelization and cluster nodes parallel mining, the time consumption tends to be stable and the variation amplitude is small, which verifies that the parallel sampling method designed in this paper can adapt to large-scale data sampling processing. Therefore, parallel random sampling based on Map Reduce in this paper can effectively reduce the selection time of initial clustering centers for large data sets.

5. Conclusions
Defects randomly selected for the k-means algorithm center point, further improving the algorithm, parallel sampling and parallel merging data objects based on Map Reduce, and replacing the initial centroid selected by the optimization method with the initial center of random selection of the traditional k-means algorithm. The improved k-means clustering algorithm is based on Map Reduce parallel processing to effectively reduce the number of algorithm iterations and improve the running time of the algorithm. The initial sampling rate of parallel random sampling is compared with other random sampling methods. It is verified that the parallel sampling proposed in this paper is more suitable for large-scale data sets, effectively reducing the selection time consumption of the initial cluster center of large data sets and improving the stability of the algorithm.

REFERENCES
[1] LI Xiaoyu, YU Liying, LEI Hang, TANG Xuefei. Implementation and application of a parallelization of K-means improved algorithm. Journal of University of Electronic Science and Technology of China, 46(01), 61-68 (2017)
[2] Li Guobao, Han Qingjiu. A MapReduce Parallelization Implementation Based on Improved K-means Clustering Algorithm. Digital Technology and Application, (12),134-136(2016)
[3] Mengya Shao. Transition-based Soft Phase Partition Algorithm of Multiphase Batch Processes Based on K-means Optimal Clustering Proceedings of the 37th China Control Conference, 2018: 6.
[4] Yu Hualong, Han Xuefeng. Bank customer classification algorithm based on improved K-means clustering. Natural Science Journal of Xiangtan University, 40(03),125-128(2018)
[5] Wu Dechao, Liu Xiaohong, Qu Zhijian. Research on Distributed Clustering Algorithm Based on Hadoop. Journal of Shandong University of Technology (Natural Science), 32(04), 25-29(2018)