ATLANTIS PRESS
Advances in Engineering Research, volume 136
7th International Conference on Advanced Design and Manufacturing Engineering (ICADME 2017)

A Distributed K - means Clustering Algorithm

JIANG Guo-song [1, a], He Xiao-ling [2,b]

[1]School of Computer Science, Huanggang Normal University, Huanggang, 438000, P. R. China;

[2]School of Journalism and Communication, Huanggang Normal University, Huanggang, 438000, P. R. China

[a]hustjgs@126.com, [b]hustjgs673521@sina.com

Keywords: K-means clustering algorithm; distributed environment; large data set; complexity.

Abstract. This paper presents a distributed clustering algorithm for large data sets. The algorithm is based on the traditional K-means algorithm to make reasonable improvements, make it more suitable for distributed environment, and analysis algorithm from complexity to compare the algorithm with the traditional centralized K-means algorithm and other distributed algorithms. Experiments show that the algorithm improves the data processing speed while keeping all the necessary features of the centralized K-means algorithm.

Introduction

One of the prerequisites for traditional clustering is that data is centralized at one site and needs to be loaded into memory at once. However, in many environments, LANs, WANs, and Internet networks connect multiple data sources into a large, distributed heterogeneous database. Users need to deal with large, multi-compute nodes, geographically distributed data, and need to protect data privacy and Safe [1]. The centralized clustering algorithm cannot be well applied to a distributed environment, even if a large amount of data is allowed to be executed centrally. Otherwise, the algorithm collapses or the execution efficiency is too low, and the execution time is too long for the user to accept it. The change of data storage method puts forward the requirements of parallelism and distribution of clustering algorithm. Distributed clustering is an effective way to solve this problem [2-3].

Distributed clustering based on the distributed data source and computing resources to clustering analysis for large-scale, distributed data, is the result of further evolution of clustering analysis, reflecting growing trend of the parallel computing, distributed computing and communication. Its idea is: first in the individual site data to perform local clustering analysis, and then part of the clustering results as output to other sites, and gathered into the final clustering results.

Based on the idea of distributed clustering, this paper proposes a distributed K-means algorithm based on the centralized K-means algorithm. The experimental results show that the proposed algorithm has higher efficiency and lower time complexity than the centralized K-means algorithm for large-scale database.

The K-means algorithm is a clustering-based clustering algorithm whose task is to divide the data set into disjoint point sets to make each point set as homogeneous as possible [4], that is, given set $P\{p_1, p_2, \dots p_N,\}$ with N data points, the goal of clustering is to find K clusters $C\{c_1, c_2, \dots c_K,\}$, so that each dot p_i is assigned to a unique cluster C_j . Where $C_i \neq \emptyset, i = 1,2, \dots K$; $C_i \cap C_j \neq \emptyset, i = 1,2, \dots K,\ j = 1,2, \dots K$, and $i \neq j$; $\bigcup_{i=1}^{k} C_i = S$.

The basic idea of the algorithm [5] is: given a database containing N data objects and the number K of clusters to be generated, randomly selected K objects, each object initially represents the average or center of a cluster, Then calculate the distance from the other samples to each cluster center, return the sample to the cluster where the cluster center is closest to it, and use the averaging method to calculate the new cluster center for the adjusted new cluster. There is no change in the cluster center of the two times, indicating that the sample adjustment is over and the clustering square error criterion function E converges, and finally all the data objects are stored in the corresponding class C_j.

The square error criterion is defined as follows:

$$E = \sum_{i=1}^{k} \sum_{p \in c_k} |p - m_i|^2 \quad (1)$$

Where:

E is the sum of the squared errors of all the objects in the database;

p is the point in the space, representing the data object;

m_i is the average of the cluster C_j (p and m_i are multidimensional).

The goal of clustering is to use Eq. (1) to minimize the value of E.

Algorithm 1: Centralized K-means

Enter: the database containing the N objects and the K value (K is an integer);

Output: K clusters, making the square criterion minimum.

Method:

(1) Randomly select K objects $p_j \in S$ as the initial clustering center;

(2) When E is unstable, the distance $d_{ij} = |p - m_i|^2 (1 \leq i \leq K \; and \; 1 \leq j \leq N)$ is calculated for each K;

(3) (re-)Assign each object to the most similar cluster according to the minimum distance from point to m_i;

(4) Calculate the new mean m_i ($1 \leq i \leq K$);

(5) Calculate E until the value of E is stable.

The complexity of the K-means algorithm is represented by O (TK N), where K is the number of expected clusters, T is the number of iterations, and N is the number of data objects.

Distributed K-means clustering algorithm

If you carefully observe the process of K-means algorithm, it is not difficult to find that K-means algorithm itself contains a distributed idea, the process is started form a set of data and a set of random clustering center, in each iterative process to assign each object to its nearest cluster. This is done only by a single processor executing the K-means algorithm. The processor's memory must contain the structure of all clusters and repeat the algorithm steps until the final clustering center m_i is estimated. However, in a distributed environment, several processors (sites) are connected over the network to execute the K-means algorithm, which means that the data sets are distributed across several sites in the network, and the processes of these sites interact with each other. The key problem to solve the distributed K-means algorithm is the global central calculation, which is the biggest difference with the centralized K-means. This paper presents a distributed K-means central algorithm for computing global centers. The implementation of the improved distributed clustering algorithm will be described in detail below.

The process of distributed clustering algorithm

For convenience of description, it is assumed that the initial data distribution is absolutely random and independent. Each site S_i arbitrarily initializes a set of central vectors $M_i = \{m_k | k = 1,2, \ldots, K\}$. After completion of the initialization, each site calculates the respective center points in parallel, and during each iteration of the distributed K-means algorithm, the local site S_i broadcasts the respective local cluster centers to other sites. After clustering at the local site, all the local data that has been clustered and the center point vector that has been estimated are denoted as $\{m_k^{(old)}\}$, and a new center point vector is calculated as $\{m_k^{(new)}\}$. In the process of calculating the new center point, in order to avoid any one site vacancy situation, the estimated center point for the data item.

The central computing is the most important feature of the distributed clustering algorithm. It is also the main difference between the distributed K-means algorithm and the centralized K-means algorithm, which can be expressed by the following mathematical formula.

K-means is:

$$m_k^{(new)} \leftarrow \frac{1}{n_k} \left\{ \sum_{p_j \in c_K} (p_j) \right\} \quad (2)$$

The central formula of the distributed K-means in this paper is:

$$m_k^{(new)} \leftarrow \frac{1}{n_k+1}\left\{\sum_{m_j \in c_K}(m_j) + m_k^{(old)}\right\} \tag{3}$$

The new center vector is distributed at all sites and operates in a broadcast manner. Each site S_i calculates the average by its own center and the center received from other sites, and uses these new averages instead of $\{m_k^{(old)}\}$. Obviously, in addition to the first step, every step in the future of all the sites have a clear center. Repeat the above process until the center vector is stable. The new central computing strategy makes the distributed K-means algorithm different from the traditional K-means algorithm. The following is a description of the distributed K-means algorithm.

Algorithm 2: Distributed K-means
Input: Integer K and data set
Output: K clusters

begin
  for each site S_i do in parallel
  initialize a set of center vectors m_k for $k = 1$ to K
  // Initialize a group of centers, which are called old centers $\{m_k^{(old)}\}$
  repeat
  for each site S_i do in parallel
  begin
    distribute local data in Si into k classes acco rding to minimum distance from m_i for $k = 1$ to K
    compute new center vectors $\{m_k^{(new)}\}$ for $k = 1$ to K considering old centers $\{m_k^{(old)}\}$ as data items
  end
  for each site S_i broadcast $\{m_k^{(new)}\}$, for k =1 to K to all other sites
  for each site S_i do in parallel
  begin
    for $k = 1$ to K do
    compute average of $\{m_k^{(new)}\}$ from self and those received other sites and replace $\{m_k^{(new)}\}$ with this average
  end
  until center vectors are stable
end

Complexity analysis of Distributed K-means algorithm

For any parallel and distributed clustering algorithm has two aspects of complexity [6], that is, time complexity T_{time} and communication complexity T_{comm}. In the calculation process, the main calculation step is to calculate the distance of each data point to the corresponding center vector; in the communication process, from a site to other sites to send data, the central vector and other related information [7]. First, the complexity of the distributed clustering algorithm in one repetition step is analyzed. Let T_{data} be the actual communication time for one data item; T_{start} is the time required to establish the connection. Since it is executed in parallel, only one data is transferred, so the complexity of each step is:

$$T_{time} = T_{start} + KT_{data}$$

Similarly, the complex of computational distance is:

$$T_{time} = KnT_{dist}$$

Where: T_{dist} is the time to calculate a single data point distance;

$n = N/P$.

Now let T be the number of cycles required for the K-means algorithm, the complexity of the whole algorithm is:

$$T_{time} = T\{T_{start} + KT_{data}\}$$

$$T_{comm} = TKnT_{dist}$$

As the network developed, the time to establish the connection can be ignored. Therefore, the complexity expression of this algorithm can be written as follows:

$T_{time} = TKT_{data}$
$T_{comm} = TKnT_{dist}$

In order to reflect the superiority of this algorithm, the complexity of Dhillon's distributed clustering algorithm is compared. In the distributed algorithm proposed by Dhillon et al. [8], the time complexity in addition to $T_{time} = TKT_{data}$, plus the transfer time of calculate the number of all vectors, local site elements, and the Euclidean minimum squares error of all local sites, but also with the calculation of all the local site of all vector time and.

And communication complexity in addition to $T_{comm} = TKnT_{dist}$, but also with the calculation of all local sites all the vector time.

Obviously, in Dhillon's scheme, the time complexity and communication complexity are somewhat higher than those described in this paper, and the empty clusters in [9] are treated as a tuple for communication, there is no empty cluster in the improved method.

Experimental results and performance analysis

The performance of the proposed algorithm was tested by two groups of experiments. The experimental platform is configured for 100 Mb / s LAN, 4 PCs, configured as Pentium IV / Intel 1 .66 GHz / 512 MB, Window s XP (Server version), 80 GB hard drive. The algorithm into a specific source code, in New Zealand Waikato University open source platform WEKA on the algorithm to verify. The experiment consists of two parts: the first part of the 6 group is the artificial two-dimensional data set, the size of 5 KB, 10 KB, 50 KB, 100 KB, 300 KB, 500 KB; Part 2 from the UCI machine learning database [8] The Iris plant dataset, which has four attributes, three categories, a total of 150 samples.

In the first experiment, six sets of different data sets were used to compare the efficiency of different clustering algorithms. In order to understand these sets of data, this paper adopts two-dimensional data sets, and is divided into three categories. Experimental results shown in Figure 1, time unit of y-axis is the clock cycle. With the increasing size of the data set, the running time of distributed clustering algorithm is obviously smaller than that of centralized clustering algorithm, and the operation efficiency is improved compared with Dhillon's distributed clustering algorithm. Growth rate if the distributed clustering algorithm proposed in this paper is also smaller than the centralized K-means algorithm.

The second experiment aims to prove the correctness of the proposed algorithm. The experimental data is the famous Iri s dataset. The distribution of the original data set is shown in Figure 2. Sites 1, 2, 3 reflect a certain part of the global space, as shown in Figure 3 ~ 5. It can see from the final clustering results (see Figure 6), the global center positioning of the distributed clustering algorithm proposed in this paper is quite accurate.

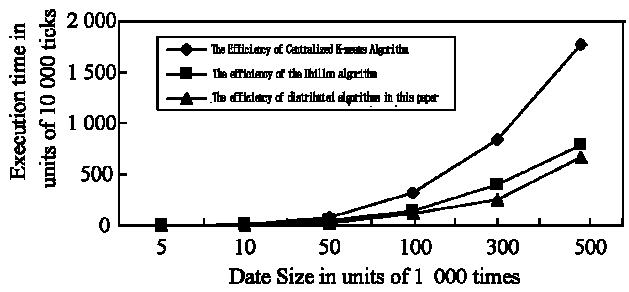

Figure 1 Implementation efficiency comparison between the algorithm proposed in this paper with the centralized K-means algorithm and Dhillon algorithm

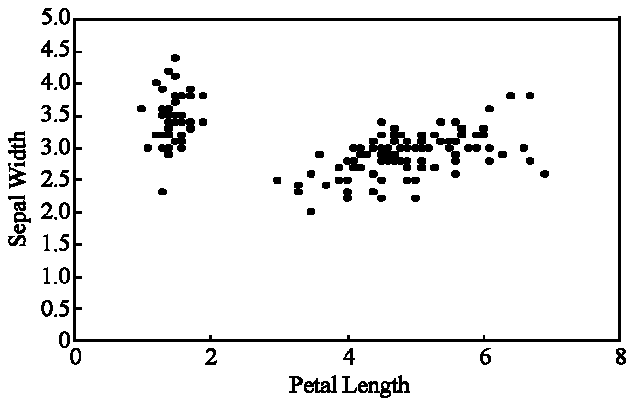

Figure 2 Iris raw data distribution

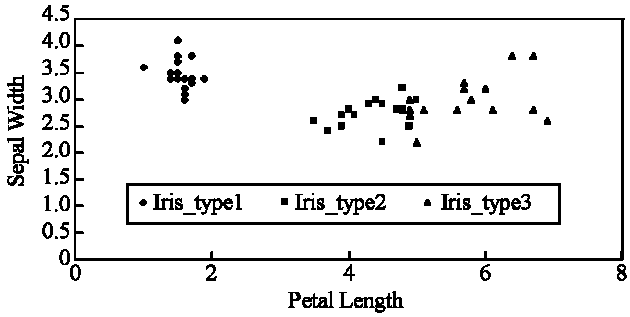

Figure 3 Site 1 data distribution

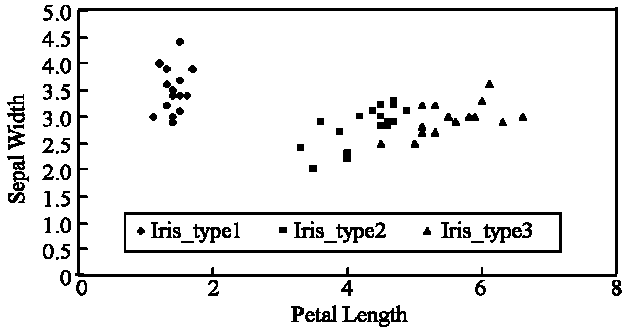

Figure 4 Site 2 data distribution

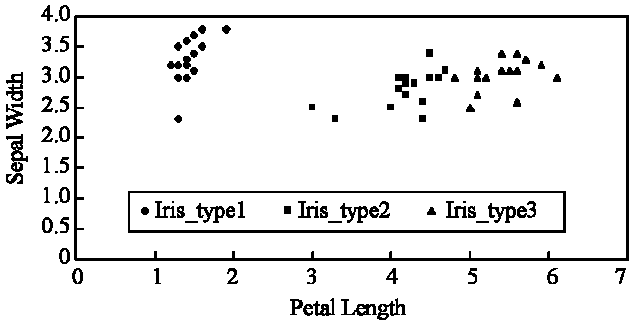

Figure 5 Site 3 data distribution

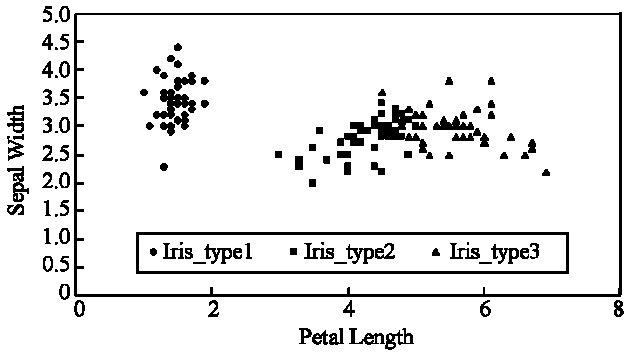

Figure 6 Distributed clustering of K-means algorithm

Conclusion

Based on the intensive research of centralized K-means clustering algorithm, this paper proposes a new distributed K-means clustering algorithm, analyzes the complexity of the new algorithm, and proves that the new algorithm greatly improve the performance of the algorithm while keeping all the features of centralized K-means clustering algorithm at the same time. The experiment also shows that the distributed algorithm proposed in this paper reduces the complexity of the algorithm and improves the efficiency of the algorithm compared with the algorithm reported in the reference.

ACKNOWLEDGMENT

This work is supported by Research project of Hubei Provincial Education Department (No.D20152903,15Y159), Foundation of Huanggang normal university(No. xfg2015A11, 2014015103, 201616303), Thanks to the reviewers for the valuable comments helping to improve the quality of the manuscript.

References

[1] LI Cheng-an. New methods for cluster analysis in distributed environments [D] :Hangzhou :Zhejiang University, 2006.

[2] ANKERST M, BREUNING M M, KRIEGEL H P, et al. Ordering points to identify the clustering structure [C] //Proc. of ACM SIGMOD International Conference on Management of Data .US A:ACM Press , 2008 :213-216.

[3] BRECHEISEN S, KRIEGEL H P, KROGER P, et al. Visually mining through cluster hierarchies[C] .Proc. of SIAM Int'l Conf. on Data Mining Orlando. USA: [s. n.], 2006.

[4] HAN Jia-wei, KAMBER Micheline. Data mining: concepts and techniques (3) [M]. Beijing: China Machine Press, 2008.

[5] KRIEGEL H P, Kr¨oger P, PRYAKH IN A , et al. Effective and efficient distributed model-based clustering [C] // Proceeding s o f 5th IEEE International Conference on Data Mining. USA: [s. n.], 2005 :258-265.

[6] CHEN Jian-mei, ZHU Yu-quan, NI Wei-wei, et al. An efficient algorithm for updating global frequent close item sets [J] .Mini-Micro Systems, 2008 , 29(7):1237-1240 .

[7] ZHAO Da-wei, XIAO Zhou-fang .Improved K-means clustering algorithm based density and sample size [J] . Science &Technology information, 2008, 28:171-172.

[8] DHILLON I S , MODHA D S .A data-clustering algorithm on distributed memory multiprocessor [J]. Proceedings of KDD-WS on High Performance Data Mining, 2009, 23(9):123-127.

[9] SANTOS D S.A biologically-indpired distributed clustering algorithm[C] // Pro c .of ACM SIGMOD International Conference on Management of Da ta .USA :ACM Press , 2009:132-137 .