An Improved Parallel Scalable K-means++ Massive Data Clustering Algorithm Based on Cloud Computing

Shuzhi Nie

***Abstract*—**Clustering is one of the most effective algorithms in data analysis and management. It has been widely used in related fields. However, with the rapid development of mass data, the traditional clustering algorithms have disadvantages of poor scalability and low efficiency. How to effectively cluster mass data has become a hot area in the field of data mining. According to the characteristics of massive data with the large amount of data and the variety of data types, used MapReduce distributed parallelization of the data processing model. For the high requirements of real-time analysis and processing, proposed an improved parallel k-means++ clustering method based on MapReduce, implemented the weighted k-means++ initialization method, improved the slow convergence speed and often converges to local optimum, reduced the MapReduce job iteration, economized a lot of network and I/O overhead etc., to improve the scalability of the algorithm, upgrade the efficiency, ensure the clustering results of the algorithm. The proposed optimization strategy can avoid a lot of distance calculation in real datasets and synthetic datasets. More importantly, as K becomes larger, the pruning ability will become more and more obvious. The result is almost perfect sequence of linear complexity of the optimal clustering results almost perfect nonlinear function approximation, while the almost perfect nonlinear function has the very good difference property, is the best function of the difference evenness. The experiment results proved the validity and superiority of the algorithm.

***Keywords*—**cloud computing, mass data, improved k-means++ clustering algorithm.

I. INTRODUCTION

Analysis and processing of massive data in the Internet era is an inevitable trend, from the current study, big data more accurate positioning of massive data development, big data is large volume and complicated structure, it’s difficult to deal with the traditional method. Big data is generally considered to have five attributes: large size, diversity, timeliness, authenticity and value, the emergence of big data has made the traditional parallel database system be challenged greatly in scalability, which makes it unable to do the task of large data analysis. The cloud computing platform and the method of data processing based on MapReduce can carry out dynamic resource scheduling and allocation, with a high degree of virtualization, high availability and high reliability, so it can meet the needs of big data analysis and processing efficiency. At present, MapReduce technology has been applied to data mining, machine learning, information retrieval, computer simulation and other fields.

MapReduce is one of the core technologies of cloud computing, which uses a "divide and rule", the problem difficult to solve directly divided into several sub problems one by one, to get the final results and the integration of each sub problem solution. MapReduce gives full play to the advantages of dealing with massive amounts of data, its programming model is simple, and it has good scalability, fault tolerance, a reasonable load balancing mechanism and task scheduling mechanism. Compared to the traditional parallel programming model, MapReduce doesn’t require users to consider complex implementation details, can develop their own parallel applications. Hadoop is the open source implementation of the Google cloud computing system, which mimics and implements the major technologies of Google cloud computing, MapReduce, GFS, and BigTable. Hadoop is a distributed computing framework, the core idea is in the cheap hardware equipment to deploy cloud computing environment, to provide a stable, reliable, and simple interface for users and applications, build a highly reliable, scalable distributed system.

As k-means clustering algorithm has the characteristics of gradient descent, causes the quality of convergence to the local optimum and the clustering result can’t be guaranteed, the fundamental reason is that the k-means clustering algorithm is very sensitive to the selection of the initial point, the improper selection of the initial point will lead to the slow convergence and low efficiency of k-means. K-means++ initialization algorithm is an outstanding work to solve the above problems. It not only improves the efficiency of the k-means++ clustering algorithm, more importantly, which gives the approximate guarantee of clustering results of the K-means algorithm, but when k-means++ initialization algorithms deal with massive data, its inefficiency is emerging. This thesis mainly studies the improved k-means ++ clustering algorithm based on MapReduce in the massive data environment of cloud computing.

II. ANALYSIS AND DESIGN OF IMPROVED METHOD

A. Analysis of Classical K-means Clustering Algorithm

K-means clustering algorithm is proposed by MacQueen in

This work was supported in part by Project 51375178 of the National Science Foundation of China.

Shuzhi Nie is with the Department of Electronics and Information Technology, Jiangmen Polytechnic, Jiangmen 529000, Guangdong, China (e-mail: sznie778@163.com)

1967, is one of the ten classic data mining algorithm, K-means clustering is a method of dividing type, divide n data objects into k clusters, make the sum of Square errors is least for each data point to the cluster centroid. The clustering algorithm is defined as follows:

Let $\mathrm{X} = \{x_1, x_2, \ldots, x_n\}$ is a data set containing n points, each point of which expressed by D vector dimension, K-means clustering algorithm divide X into k disjoint clusters $\mathrm{Y} = \{Y_1, Y_{2,\ldots,}Y_n\}$, for any $1 \leq \mathrm{i} \neq \mathrm{j} \leq \mathrm{k}$, as a cluster Y_i, its center point can be shown as follows:

$$c_i = \frac{1}{|Y_i|}\sum_{y \in Y_i} y \tag{1}$$

Set $\mathrm{C} = \{c_1, c_2, \ldots, c_n\}$ is the center of k clusters, and $\|x_i - x_j\|$ is the Euclidean distance of points x_i and x_j, then

$$\mathrm{SSE(C)} = \sum_k \binom{n}{k} \tag{2}$$

In other words, the goal of the k-means clustering algorithm is to find an optimal partition C, which minimizes SSE(C). By dividing the data set which containing n objects into k non-intersecting nonempty sets, for the large number of division methods, can calculate by the second type of Stirling number, that is:

$$\mathrm{S}(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^{k-i}\binom{k}{i} i^n \tag{3}$$

This value is approximately equal to $k^n/k!$. From this can see, the computation of the global optimum is too large by enumerating all possible clusters and finding the global optimum. In fact, this non-convex optimization problem has been proved to be the NP-hard problem; so many heuristic algorithms are proposed to obtain an approximate optimal solution.

As the initial k-centric points of the k-means clustering algorithm are randomly selected, the different central point will make the clustering results of k-means different, so the evaluation of k-means clustering is also a very important problem. As the k-means clustering algorithm has many advantages, it is widely used in reality. On the other hand, the k-mean has some main shortcomings are shown as follows: k-means clustering algorithm need to specify the number of clusters k in advance; can detect the compact, super-spherical, separated from each other better cluster; use the square of the Euclidean distance as a measure of similarity, which is very sensitive to the noise and outliers, and even a very small number of points can significantly affect the mean of the cluster; SSE has a gradient of the characteristics, so it is usually has local convergence, that is, the local optimal solution, but the global optimal is difficult to achieve; select the initial point is sensitive, so that the accuracy of k-means clustering results can't be guaranteed, different initial points will have different clustering results, Inappropriate initial points lead to empty clusters, which converge slowly and are likely to fall into local optimum.

B. Improved K-means++ Clustering Algorithm

To improve the quality of clustering, K-means clustering algorithm involves iterative operation, while the MapReduce model leads to lack of iterative support, need start a MapReduce job for each iteration operation, which caused a large number of I/O and network overhead. MapReduce implementation of K-means clustering algorithm is relatively simple; the algorithm process is similar to the traditional K-means clustering algorithm, K-means clustering algorithm aiming at the existing problems, the biggest defect is that it is sensitive to the choice of the initial point, thus selecting the initial points has become an important research field. Scalable k-means++ method is the parallelization of k-means++, and also the synthesis of k-means++ initialization method and clustering algorithm. In this thesis, proposed an improved parallel scalable k-means++ initialization algorithm, whose Map processing and Reduce processing are shown as follows.

In the stage of Map, for the standard k-means++ initialization algorithm, pre specified the number of clusters k, contains the n object data set C; let $\mathrm{C} \leftarrow \emptyset$, the center point X implementation of random sampling is x, then:

$$\mathrm{C} = \mathrm{C} \cup \{\mathrm{x}\} \tag{4}$$

Loop until $\mathrm{num[i]} = 0$; when meeting $|C| \leq k$, calculate the distance interval $d^2(\mathrm{x}, \mathrm{C})$ closest to the center point C in section $\mathrm{x} \in \mathrm{X}$, then

$$\mathrm{p(x)} = \mathrm{d}^2\,(\mathrm{x},\mathrm{C}) / \sum_{\mathrm{x}\in\mathrm{X}} \mathrm{d}^2(\mathrm{x},\mathrm{C}) \tag{5}$$

In the stage of Reduce, to improve the clustering quality of the final implementation of the k-means++ algorithm, initialized weights, pre specified number of clusters K, contain the N object data set $\langle num, C\rangle$, set $\mathrm{C} \leftarrow \emptyset$; perform random sampling the center X, loop until $|C| \leq k$; calculate the distance interval $d^2(\mathrm{x}, \mathrm{C})$ closest to the center point C in section $\mathrm{x} \in \mathrm{X}$, then

$$\mathrm{p(x)} = num * d^2\,(x, C) / \sum_{x \in X} d^2(x, C) \tag{6}$$

The improved algorithm with the biggest difference is that the classical algorithm, k-means++ initialization algorithms are executed on the Map stage and Reduce stage, different Map stage execute standard k-means++ initialization algorithm, and Reduce phase is the implementation of the k-means++ initialization algorithm weighted.

If implementing the standard algorithm in the k-means++ initialization algorithm of Map phase and Reduce phase, will make the quality of clustering results is very poor, SSE is away from the optimal solution of the K-means clustering, result $\mathrm{SSE}/\mathrm{SSE_{OPT}} \gg \alpha$. Choose k center in the Map stage, another important work is to calculate the number of each selected center point on behalf of the point, the value is used as a weight in the Reduce phase; for each point X using the weighted probability value is $\mathrm{p(x)} = \mathrm{num} * \mathrm{d}^2\,(\mathrm{x},\mathrm{C}) / \sum_{\mathrm{x}\in\mathrm{X}} \mathrm{d}^2(\mathrm{x},\mathrm{C})$, determine whether it is the initial center point of k-means clustering algorithm; if the clustering result is generated by the standard k-means++, it will causeSSE≤αSSE_OPT, among them

$$\alpha = 8(2 + \ln k) \tag{7}$$

$$\mathrm{SSE}_{OPT}=\sum_{x\in X}\min_{c\in C}\left\|x-\hat{c}\right\|^2 \qquad (8)$$

If the weighted k-means++ initialization algorithm runs in the Reduce stage, then

$$\sum_{x\in X}\min_{c\in C}\|\emptyset(x)-c\|^2\le\alpha\sum_{x\in X}\min_{c\in C}\left\|\emptyset(x)-\hat{c}\right\|^2 \qquad (9)$$

At the end of the Reduce task, the weighted k-means++ initialization algorithm will produce K central points. In spite of selecting on the center point sets Y of all Map tasks, but it takes into account the number of center points in the Y represent the points in X. At the same time, the k-means++ initial method need to iterate K times to select k centers, which is inherent in nature and can't be reduced. However, when a new central point is determined, there is no need to recalculate the distance between all points and the center point.

III. Experimental Design and Analysis

A. Experimental Environment

All the experiments were run on the Hadoop cluster of isomorphic nodes, used Hadoop 1.2.1 to successfully build a database containing 13 nodes of massive data analysis and processing platform, this platform contains 1 master nodes, 12 slave nodes, the specific configuration information is as follows: Intel Xeon E5-2620 2.0GHz CPU, 8GB memory, 2TB hard disk, 100M/1000M adaptive Ethernet, Gigabit Ethernet controller, the operating system is Ubuntu 16.04.2, the cloud computing platform is Hadoop 1.2.1, JDK 1.8.131 development environment, development tools is Eclipse 4.5 neon. The data information used in the experiment is shown as follows:

The Oxford building data sets is a real data set, which contains 5062 images drawn from Filckr on a specific landmark in Oxford. Each image is extracted into 128 dimensional SIFT features, which contain 16,334,870 features, the size is 5.67GB.

Each point of synthetic data sets is the 128 dimensional vector, which contain 5000 centers, namely 5000 clusters, the other points are around these center points are generated, each cluster contain about 4000 points, the entire data set contain the number of points more than 20 million, this data set the size is about 15GB.

B. Comparison and Analysis of Experimental Results

First, compare the efficiency of the different initialization algorithms. As mentioned above, the improved parallel extensible k-means++ initialization algorithm (IPSKMI) use only 1 MapReduce can choose the k center, but the classic k-means++ initialization algorithm (CMRKMI) MapReduce need 2K MapReduce operation, the improved IPSKMI algorithm doesn't eliminate the iterative character of k-means++ initialization algorithm, just replace the original iteration across network nodes into the iteration of multiple local node, this IPSKMI algorithm can economize a lot of network and I/O cost. Network and I/O cost usually is the bottle neck of the MapReduce processing model, especially the network overhead and performance in the running time, IPSKMI algorithm has a shorter running time. Take K as 1000, the number of iterations is 13 times, the sampling factors respectively is γ=0.1k, γ=0.5k, γ=2k, the contrastive experiment results are shown in Table 1.

Table 1. Running time comparison of IPSKMI and CMRKMI

CMRKMI			IPSKMI
$\gamma = 0.1k$	$\gamma = 0.5k$	$\gamma = 2k$	
189min	435min	615min	25min

Then, choose the different algorithms to compare the SSE of the center point. Compare the SSE of parallel scalable IPSKMI initialization algorithms and random initialization algorithm (RandI) in Oxford buildings and synthetic datasets. As the data sets are high dimensional data, and the value of K is larger, leading to CMRKMI method execution time is too long, so the degree of approximation of this experiments no comparison between IPSKMI and CMRKMI. While the value of K varies from 1000 to 5000, the experimental results are shown in Figure 1 and figure 2. As the SSE range of synthetic data varies widely, to show the experimental results more clearly, while the K changes from 1000 to 5000, the experimental results of synthetic data is shown in Table 2. From Figure 1 and Figure 2, whether Oxford building real data sets or synthetic data sets, the SSE of IPSKMI and RandI reduce along with the increase of the center points; but the former can get a better approximation, especially the experiments on Oxford building data sets. In the same number of center points, while the SSE of IPSKMI and RandI are in the same order of magnitude, but the IPSKMI SSE is smaller than RandI; and as the center number increased, due to the random characteristics of RandI, its SSE has obvious fluctuations, but the change trend of SSE IPSKMI is more stable than RandI.

Table 2. SSE comparison on synthetic data of IPSKMI and RandI

k	1000	2000	3000	4000	5000
RandI	5.36E12	1.41E12	7.59E11	3.98E11	2.46E11
IPSKMI	1.58E12	3.51E11	1.42E11	6.40E10	1.98E10

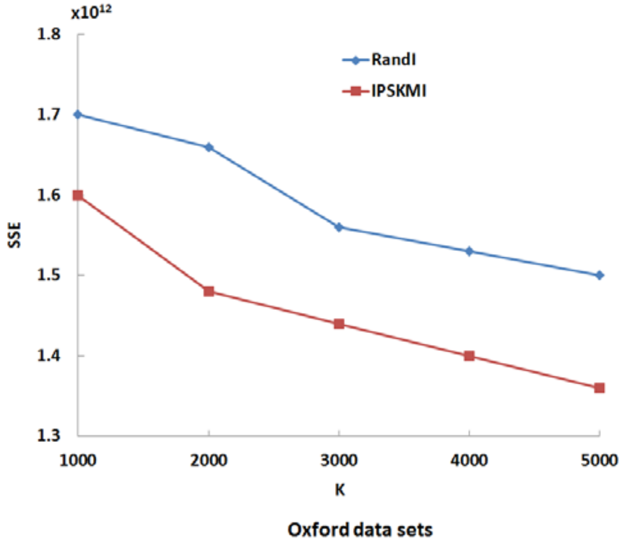

Fig.1. Changes in SSE under Oxford data sets

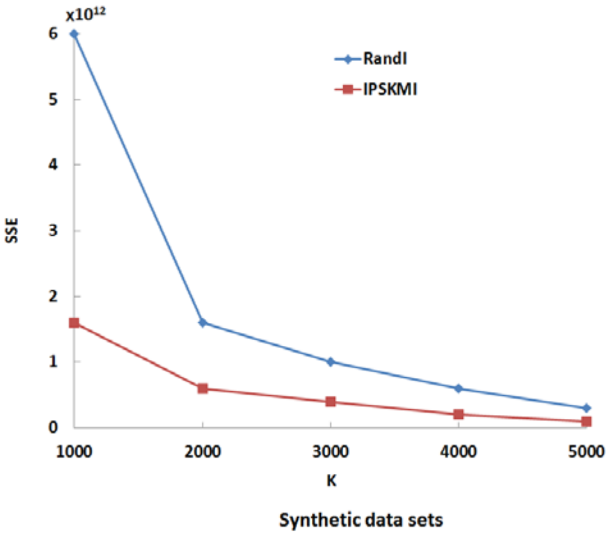

Fig.2. Changes in SSE under Synthetic data sets

Finally, compare the clustering results SSE of different algorithms. Compare the SSE of parallel scalable IPSKMI initialization algorithms and classic k-means initialization algorithm (CMRKMI) in Oxford buildings and synthetic datasets. Set the K value is 5000; use 5000 initial center points of previous set of experiments; to save time, the number of iterations of the experiment is set to five, the results of the experiment are shown in Figure 3 and Figure 4, and put the experimental results of the synthetic data into table 3. From Figure 3 and Figure 4 can see, while the number of iterations increased from 1 to 5, both the IPSKMI and the CMRKMI SSE are reduced, which also reflects the characteristics of the K-means clustering algorithm of gradient descent, but the IPSKMI algorithm is better than the CMRKMI in approximation. For the Oxford building data sets, the largest SSE gap occurs at the first iteration between the IPSKMI and the CMRKMI, with the increase of the number of iterations, the gap is getting smaller and more stable, when the last iteration occurs, the gap between the two reaches the minimum. For the synthetic data sets, compared to CMRKMI, IPSKMI on the degree of approximation has a great advantage, when in the first iteration; the SSE that occurred at the first iteration is 1/10 of CMRKMI, when the number of iterations varies from 3 to 5, IPSKMI SSE is close to a constant, from this point can know the proposed algorithm has faster convergence speed; after 5 iterations, IPSKMI in the real data set and synthetic data set on the SSE are stabilized, so use an improved parallel k-means++ initialization method can be extended to get better clustering results.

Table 3. SSE comparison on synthetic data of IPSKMI and CMRKMI

iteration	1	2	3	4	5
CMRKMI	1.13E11	6.59E10	5.77E10	5.42E10	5.31E10
IPSKMI	1.124E11	1.025E10	1.015E10	1.014E10	1.012E10
IPSKMI	1.124E11	1.025E10	1.015E10	1.014E10	1.012E10

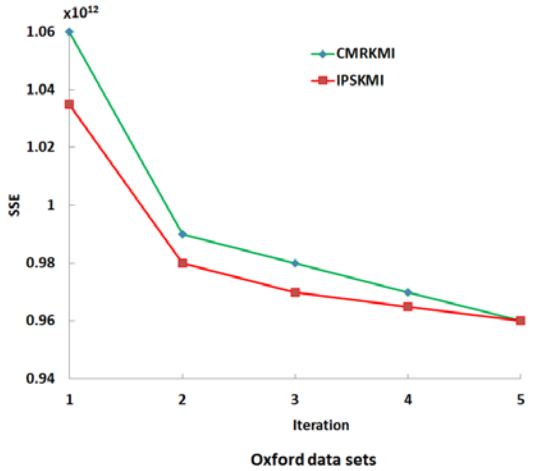

Fig.3. SSE comparison of IPSKMI and CMRKMI in Oxford data sets

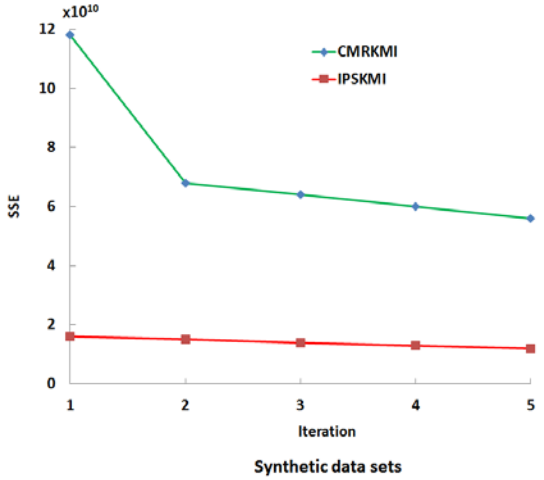

Fig.4. SSE comparison of IPSKMI and CMRKMI in Synthetic data sets

IV. Conclusions

This thesis studied the existing problems of k-means++ clustering algorithm in large data scenarios, proposed an improved scalable parallel k-means++ clustering algorithm (IPSKMI), can implement efficiently in the MapReduce framework. The improved parallel scalable k-means++ clustering algorithm proposed in this paper, can use only one MapReduce job to select k points for k-means. As the IPSKMI algorithm reduced the number of MapReduce operations, economized a lot of network overhead and I/O, and proved that it is approximate K-means optimal clustering results, further improved the efficiency of the algorithm.

The proposed optimization strategy can avoid a lot of distance calculation in real datasets and synthetic datasets. More importantly, as K becomes larger, the pruning ability will become more and more obvious. The result is almost perfect sequence of linear complexity of the optimal clustering results almost perfect nonlinear function approximation, while the almost perfect nonlinear function has the very good difference property, is the best function of the difference evenness. Finally, the experimental results show that the algorithm is effective and superior. To further optimize the algorithm, we will study the optimization of pruning strategies in the future.

REFERENCES

[1] Gog S, Petri M. “Optimized succinct data structures for massive data”, Software: Practice and Experience, vol.44, no.11, pp. 1287-1314, 2015.

[2] Mohebi A, Aghabozorgi S, Wah T Y, et al. “Iterative big data clustering algorithms: a review”, Software-practice & Experience, vol.46, no.1, pp. 107-129, 2016.

[3] Traganitis P A, Slavakis K, Giannakis G B, “Big Data Clustering via Sketching and Validation”, IEEE Journal of Selected Topics in Signal Processing, vol.9, no.4,pp.1-1, 2015.

[4] Singh I, Dwivedi P, Gupta T, et al. “An enhanced K-means clustering algorithm for big data in cloud”, International Journal of Pharmacy & Technology, vol.8, no.4, pp. 26066-26075, 2016.

[5] Lathiya P, Rani R, “Improved CURE clustering for big data using Hadoop and Mapreduce”, International Conference on Inventive Computation Technologies, IEEE, 2017, pp.1-5.

[6] Kumar D, Palaniswami M, Rajasegarar S, et al. “cultivate: A mixed visual/numerical clustering algorithm for big data”, IEEE International Conference on Big Data, IEEE, 2013, pp.112-117.

[7] Bharill N, Tiwari A, Malviya A. “Fuzzy Based Clustering Algorithms to Handle Big Data with Implementation on Apache Spark”, IEEE Second International Conference on Big Data Computing Service and Applications, IEEE, 2016, pp.95-104.

[8] Ketu S, Prasad B R, Agarwal S. “Effect of Corpus Size Selection on Performance of Map-Reduce Based Distributed K-Means for Big Textual Data Clustering”, International Conference on Computer and Communication Technology, ACM, 2015, pp.256-260.

[9] Srikanth P. “Clustering algorithm of novel distribution function for dimensionality reduction using big data of OMICS: Health, clinical and biology research information”, IEEE International Conference on Computational Intelligence and Computing Research, IEEE, 2017, pp.1-6.

[10] Bordogna G, Frigerio L, Cuzzocrea A, et al. “An Effective and Efficient Similarity-Matrix-Based Algorithm for Clustering Big Mobile Social Data”, IEEE International Conference on Machine Learning and Applications, IEEE, 2017, pp.514-521.

[11] Hatamlou A, Hatamlou M. “Hybridization of the Gravitational Search Algorithm and Big Bang-Big Crunch Algorithm for Data Clustering”, Fundamenta Informaticae, vol.126, no.4, pp.319-333, 2013.

[12] Akasapu, A. K., et al. "Density Based k-Nearest Neighbors Clustering Algorithm for Trajectory Data" International Journal of Advanced Science & Technology, no.31, pp.47-5731, 2011.

Shuzhi Nie was born on Nov. 10, 1977**.** He received the PhD degree in Mechanical Manufacturing and Automation from South China University of Technology of China. Currently, he is a researcher (assistant professor) at Jiangmen Polytechnic, China. His major research interests include advanced manufacturing technology, intelligent computing and simulation optimization. He has published many papers in related journals.