ICIC Express Letters
Part B: Applications
Volume **8**, Number **7**, July **2017**

 ISSN 2185-2766
pp. **1023–1029**

OPTIMIZATION AND IMPLEMENTATION OF K-MEANS CLUSTERING ALGORITHM ON HADOOP PLATFORM

Baolong Liu, Jin Su and Xiaohao Su

School of Computing Science and Engineering
Xi'an Technological University
No. 2, Xuefu Middle Road, Weiyang Dist., Xi'an 710021, P. R. China
liu.bao.long@hotmail.com

Received February 2017; accepted April 2017

Abstract. *Although there are many advantages of traditional K-means algorithm, the clustering criterion function has poor performance on classification of the data set of cluster uneven density. On the basis of weighted standard deviation criterion function, the paper proposes an optimized K-means parallel algorithm based on MapReduce on the Hadoop platform. Compared to the traditional K-means algorithm, the presented parallel algorithm has a significant improvement on the accuracy, speedup ratio, scalability and the convergence of clustering results. It also reduces the probability of misclassification caused by the uneven cluster density, and improves the clustering accuracy of the original algorithm. Experimental results show that the optimized algorithm is suitable to deal with a very large amount of data set.*
Keywords: K-means, Cluster density, Clustering accuracy, MapReduce, Hadoop

1. **Introduction.** The commonly used K-means algorithm is a cluster mining algorithm which is based on partition. The characteristics of the algorithm are that it is simple and has fast convergence speed as well as easily to be implemented. However, the K-means algorithm still has some defects. (1) Because the selection of K value has certain blindness and randomness, the clustering results often fall into the local optimum. (2) The initial cluster centroid selection is random, and for different initial centers may lead to different clustering results. (3) For a very large amount of data, the clustering iterations are cumbersome, which makes a low efficiency of K-means algorithm as well as the phenomenon of memory overflow. (4) Due to lack of scalability and difficulty to expansion, parallel processing ability of the algorithm is poor.

Many researchers have proposed different solutions for limitations above. [1] adopted multiple random sampling to determine the K value to solve the problem of selecting the number of cluster centers. Because the traditional K-means algorithm needs to traverse the data set many times on Hadoop platform, [2] presented the improved selection algorithm of M+K-means based on initial clustering center. It makes a more detailed optimization of the initial center point and the selection of K value, and greatly reduces the traversal time of the original algorithm. [3] optimized the memory leakage problem in the iterative computation process. [4-7] proposed a new clustering algorithm Mrk-means using reorganization technique based on MapReduce, which improved the running efficiency and time complexity of the original algorithm. [8] combined the Spark framework and Hadoop technology to realize the distributed K-means clustering algorithm which improved the throughput and expansibility of the algorithm. In order to avoid the influence of noise and outliers on K-means algorithm, [9-14] proposed a new clustering validity function, which improved the quality of data clustering. Banerjee and Ghosh proposed a proportional equilibrium clustering algorithm to improve the clustering effect of clusters [7]. Aimed at that the K-means algorithm is vulnerable to the interference of outliers, [15]

improved the clustering accuracy and convergence speed of the algorithm. [16,17] used the average error criterion function to achieve better clustering results in the iterative process of algorithm, which guarantees the validity and reliability of clustering results.

However, existing K-means algorithm does not explicitly point out when the algorithm achieves a convergent value as the finish of the algorithm. It also does not take the value of the criterion function as a sign of the end of the algorithm. This paper uses the advantages of the weighted standard deviation criterion to present an optimized K-means algorithm based on MapReduce distributed programming model. Compared to existing K-means algorithm, the presented parallel algorithm has a significant improvement on the accuracy, speedup ratio, scalability and the convergence of clustering results. It also reduces the probability of misclassification caused by the uneven cluster density, and improves the clustering accuracy of the algorithm.

2. **The Traditional K-means Algorithm.** The traditional K-means algorithm chooses k ($k \geq 1$) objects as initial clustering centers from n ($n \geq 1$) data objects. The remaining $n - k$ objects are distributed to the nearest cluster according to their similarities to the center of the clusters. For each new cluster, the clustering center is recalculated. This process is iteratively executed until the criterion function is converged [15,19]. The detailed K-means algorithm is described as follows.

(1) For any data set X, set the number of clusters as K, and randomly select any k data objects as the initial centroid of the cluster u_k, and $u_1, u_2, \ldots, u_k \in U$.

(2) For each data point $x_p \in X$ and $x_p \neq u_k$, the Euclidean distance to each centroid is calculated. The data point is assigned to the cluster of the smallest Euclidean distance. Euclidean distance is described in Formula (1).

$$d(x_p, u_i) = \sqrt{\sum_{i=1}^{k} (x_p - u_i)^2} \tag{1}$$

(3) For each new class U_k, the centroid is recomputed.

(4) Steps (1) and (2) are iteratively executed until Formula (2) reaches convergence.

$$J_{SSE} = \sum_{i=1}^{K} \sum_{x_p \in U_i} \arg\min |x_p - u_i|^2 \tag{2}$$

where, J_{SSE} represents the convergence value of the clustering criterion function, K represents the number of clusters, U_k represents the kth cluster, u_k represents the center of cluster U_k, x_p represents an arbitrary data object.

The major purpose of Formula (2) is to minimize the sum of the squares of the total error in the cluster so as to obtain the best clustering results. Obviously, if J_{SSE} value is smaller, the error is smaller, and the clustering result is better. The distribution of clustering function is only suitable to data samples which have substantially spherical and uniform density between clusters, and the data number of each sample has slight difference. The clustering criterion function cannot effectively deal with uneven density and different types of sample data sets, and it often results in many large clusters being split into small clusters. This heavily affects the quality of clustering results.

3. **The Optimization of K-means Algorithm.** In order to solve the deficiency of traditional K-means clustering criterion function, a weighted standard deviation clustering criterion function was proposed as shown in Equation (3) [17]. The function adopted the standard deviation to reflect the discrete degree of a data set. If the standard deviation is higher, the experimental data is more discrete which means less accuracy. Otherwise,

if the standard deviation is lower, the experimental data is more accurate.

$$\varepsilon = \sum_{i=1}^{K} \frac{m_i}{M} \sigma_i = \sum_{i=1}^{K} \frac{m_i}{M} \sqrt{\frac{\sum_{j=1}^{m_i} \arg\min |x_{ij} - u_i|^2}{m_i}} \tag{3}$$

where, M is total number of data objects, K is the number of clusters, σ_i is the standard deviation of ith cluster, m_i presents the number of data objects in cluster i, m_i/M refers to the weight of σ_i, which makes standard deviation of the clusters with big data objects have significant effect on the criterion function.

The following conclusions can be drawn from Formula (3). In order to get the minimum value ε, for any data point, it should select a cluster in which increment of the value ε is small in the process of K-means algorithm iteration. In other words, it should select the minimum value of the weighted distance of $z_i \cdot \arg\min |x - u_i|$ to join the cluster, where, $z_i = 1/\sqrt{\sigma_i}$ is the weighted coefficient. In this way, the data points in each iteration process will be assigned to the cluster which contains amount of data points. When the clusters with different size and density are adjacent to each other and the space is relatively small, the possibility that the data on a large sparse cluster boundary is divided into a small cluster with high density being reduced.

If we further analyze the value of the weighted standard deviation clustering criterion function, it is not difficult to find that the value of the function is a monotone decreasing curve under the ideal state in the iterative calculation process. This is because for any data object, data objects are assigned to the cluster which is nearest to them during the clustering iteration calculation. With the continuous adjustment of clustering centroid, the data objects will move toward the cluster in favor of their own closing to, and the ε will gradually approach a fixed value. When the value does not change, the whole algorithm achieves the optimal clustering.

However, existing algorithm does not explicitly point out when the algorithm converges to what extent can be considered as the finish of the algorithm. It also does not take the value of the criterion function as a sign of the end of the algorithm. Based on the advantages of Formula (3), the paper uses the minimum weighted distance to determine the class of data points that should be assigned, and increases the speed of the convergence of the K-means algorithm. The algorithm can be further optimized as follows.

(1) For any data set $X = \{x_1, x_2, \ldots, x_n\}$ and $x_p \in X$, randomly select any k data objects from data set X as the initial centroid of the cluster $u_1, u_2, \ldots, u_k \in R^n$, and set $q = 1$.
(2) For each data point x_p, calculate the weighted distance of each data x_p object to cluster centroid $d(x_p, u_i) = z_i \cdot \arg\min |x_p - u_i|$, $i = 1, 2, \ldots, k$. If it satisfies the equation of $d(x_p, u_i) = \min\{d(x_p, u_i)\}$, it assigns data point x_p to cluster U_i, $x_p \in U_i$. Finally, k new clusters will be generated.
(3) Set $q = q+1$ for each new class U_i, recompute the centroid of the class u_i: $u_i(q+1) = \frac{\sum_{p=1}^{n} x_p^{(U_i)}}{n}$, where, n represents the total number of data points assigned to the cluster.
(4) Calculate the value of the weighted standard deviation clustering criterion function:

$$\varepsilon(q+1) = \sum_{i=1}^{K} \frac{m_i}{M} \sqrt{\frac{\sum_{j=1}^{m_i} \arg\min |x_{ij} - u_i(q+1)|^2}{m_i}} \tag{4}$$

(5) If it satisfies the equation $0 \leq \varepsilon(q) - \varepsilon(q+1) < \delta$, where δ is a small constant, there are no data objects in the cluster having been adjusted. This means that the algorithm has reached the optimal clustering. Otherwise, the process returns to step (2) until the clustering criterion function converges.

4. **Implementation of Optimized K-means Algorithm.** This paper proposes an idea to design K-means algorithm on the Hadoop platform based on the MapReduce. The proposed implementation transforms the iterative process of serial K-means algorithm into MapReduce computation that can be executed independently with a number of MapReduce computing tasks working together on different computers. The iterative processing consists of three parts including the Map phase, Combine phase and Reduce phase as shown in Figure 1.

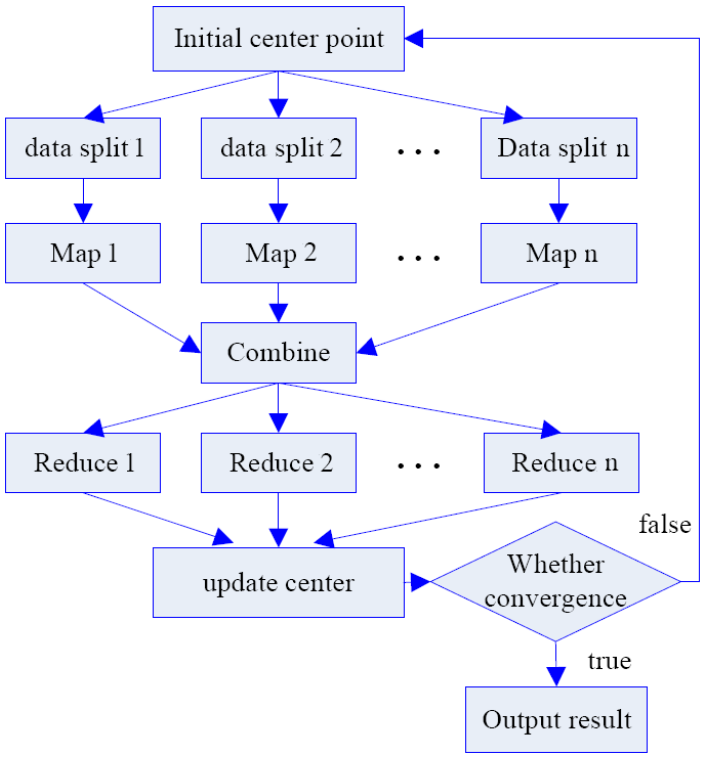

FIGURE 1. The parallel algorithm of K-means based on MapReduce

In the Map phase, the algorithm randomly selects K central points, stores the K central points in HDFS as the starting global variable. The whole data set is divided according to the minimum weighted distance. In Map processing, it causes excessive writing because of the large intermediate result in each data node, which consumes large amounts of system resources, and increases the operation cost. The paper introduces the Combine function to optimize MapReduce intermediate result, and the value of the same key in the key-values can greatly reduce the excessive amount of data in disk. In addition, this phase also takes final results of the Combine function as an input for the Reduce stage. It greatly reduces the burden of the remerge intermediate results in the Reduce phase and improves the execution speed of the algorithm. The output of the Reduce phase is taken as a new round of iterative calculation for the center point coordinate, and uploaded to HDFS. The iteration is completed until the algorithm converges.

5. Experimental Results.

5.1. Evaluation criteria.

This paper mainly uses the following measurement criteria: accuracy, speedup ratio and scalability of algorithms. In addition, experimental results are respectively compared to the traditional K-means algorithm. The experimental data is the real data selected from the UCI data sets (http://archive.ics.uci.edu/ml/datasets.html).

5.2. **Experimental results and analysis.** In order to test the acceleration ratio of the algorithm, five data samples are tested. The sizes of samples are 1 million, 2 million, 4 million, 8 million and 16 million, which are randomly selected from the UCI data set. The accuracy of the optimized K-means algorithm is significantly higher than the traditional K-means algorithms and the developed K-means [10] as shown in Table 1. This is easy to understand in data preprocessing phase. When data size and dimension increase linearly, the K value is also increasing. This means the distance calculation between the data vector and the class center becomes quite a lot for a data vector. However, the optimized K-means algorithm based on the standard deviation clustering criterion function is assigned the data object to the cluster in which the weighting distance is minimized. This improvement greatly reduces the times of distance calculation, avoids too much redundant computation, and greatly improves the operation efficiency of the algorithm.

Table 1. The comparison of two algorithms on Hadoop platform

UCI data set	*Hadoop platform*	*Convergence value of clustering criterion function*	*Running time/s*	*accuracy%*
Date set 1 (1 Million)	*K-means*	$J_{SSE} = 128.8$	4.5	50.4
	Developed K-means [10]	$J_{SSE} = 106.5$	3.2	60.8
	Optimized K-means	$\varepsilon = 82.3$	1.4	70.5
Date set 2 (2 Million)	*K-means*	$J_{SSE} = 119.2$	7.6	60.2
	Developed K-means [10]	$J_{SSE} = 88.3$	4.7	68.7
	Optimized K-means	$\varepsilon = 71.1$	2.7	75.1
Date set 3 (4 Million)	*K-means*	$J_{SSE} = 108.1$	17.8	65.3
	Developed K-means [10]	$J_{SSE} = 79.2$	9.2	70.7
	Optimized K-means	$\varepsilon = 67.9$	4.9	77.8
Date set 4 (8 Million)	*K-means*	$J_{SSE} = 100.5$	22.8	61.9
	Developed K-means [10]	$J_{SSE} = 75.3$	11.2	79.2
	Optimized K-means	$\varepsilon = 65.1$	7.1	85.1
Date set 5 (16 Million)	*K-means*	$J_{SSE} = 88.3$	30.3	55.2
	Developed K-means [10]	$J_{SSE} = 70.6$	16.6	83.4
	Optimized K-means	$\varepsilon = 60.4$	8.2	93.8

5.3. **Acceleration ratio.** Five data samples are tested with 1 million, 2 million, 4 million, 8 million and 16 million to evaluate the acceleration ratio of the algorithm. These data sets are randomly selected from UCI data set.

The acceleration ratio of the designed algorithm is basically linear as shown in Figure 2. When the data size is large, the acceleration ratio of the improved K-means parallel algorithm is close to linear growth. With increasing data nodes, the acceleration rate of the algorithm is gradually slowing down. When the data size is the same, the increasing of the node will cause the communication overhead of each node, and this takes up a part of necessary processing time. Combining Table 1, it can be seen that the acceleration ratio of the improved K-means parallel algorithm is significantly higher than traditional algorithm. This is because the optimized algorithm in the stage of Map adds the designed Combine function, which is used for localizing Reduce preprocessing for a large number of intermediate results generated in the Map phase. The benefits of this improvement reduces the I/O transmission between data nodes, and greatly saves the cost of the algorithm.

5.4. **Algorithm expandability analysis.** With increasing of the data nodes, the running efficiency of the algorithm is gradually declining on the testing data set of different

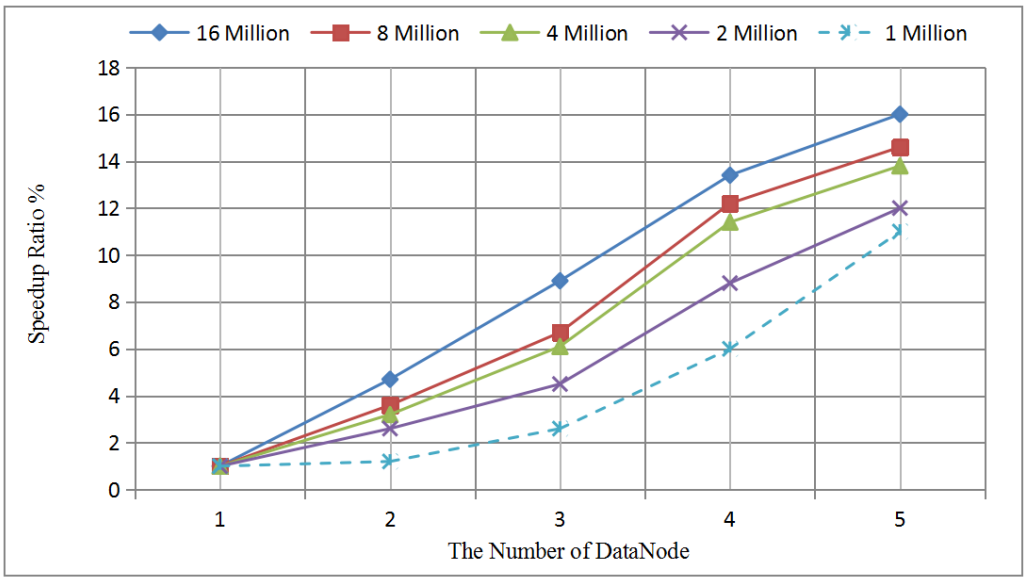

FIGURE 2. Acceleration ratio evaluation

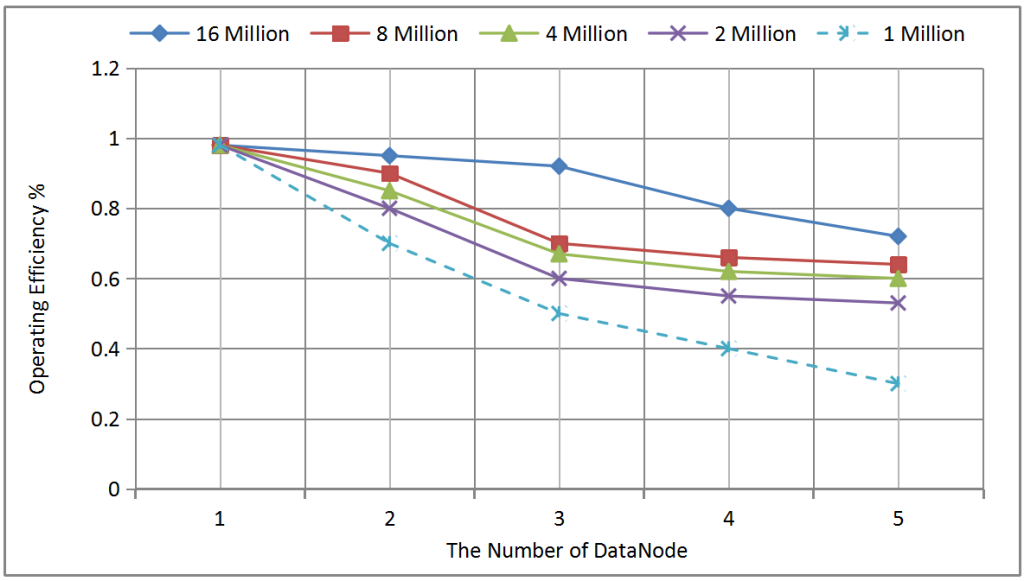

FIGURE 3. The expansion evaluation of optimized K-means algorithm

scales as shown in Figure 3. When there are multiple DataNode running on the Hadoop platform, each node needs to transmit and merge excessive number of intermediate data in the Reduce phase. It increases the cost of the communication between nodes, and consumes a lot of system resources. As for the same size of the data set, with the increasing of nodes, the efficiency of a large amount of data sets is significantly better than the small amount of sample sets. Especially in the fourth node, the efficiency of running 16 million data is significantly higher than that of the 8 million's. With increasing of data amount, data processing time is far greater than the communication overhead of each node, which makes the DataNode of the Hadoop platform is more likely to play its parallel computing power. It can be seen that the algorithm proposed in this paper has significantly improved the processing ability of the system, and has good scalability.

6. **Conclusion.** The paper optimizes the clustering criterion function of the traditional K-means algorithm using weighted clustering. In addition, this paper successfully designs

and implements the optimized K-means algorithm based on the MapReduce programming model. Experimental results show that the optimized algorithm improves clustering accuracy, accelerating ratio and expandability. Compared to existing K-means algorithm, the proposed algorithm is more suitable to deal with a large scale data processing.

Acknowledgment. This work is partially supported by Science & Technology Program of Shaanxi Province with project "2015KTCXSF-10-11", Science & Technology Program of Weiyang District of Xi'an City with project "201609".

REFERENCES

[1] X. Lei, K. Xie and Z. Xia, An efficient clustering algorithm based on local optimality of K-means, *Journal of Software*, vol.19, no.7, pp.1683-1972, 2008.

[2] X. Wu, Z. Dong and X. Meng, Research on K value optimization of clustering algorithm based on large data Hadoop platform, *Journal of Taiyuan University of Science and Technology*, vol.36, no.2, pp.92-96, 2015.

[3] Q. Yu, M. Dai and J. Li, ACO-Kmeans parallel clustering algorithm based on MapReduce, *Computer Engineering and Applications*, vol.49, no.6, pp.117-120, 2013.

[4] S. Shahrivari and S. Jalili, Single-pass and linear-time K-means clustering based on MapReduce, *Information Systems*, vol.60, no.3, pp.12-36, 2016.

[5] Z. Tang, *Design and Implementation of Machine Learning Platform Based on Spark*, Master Thesis, Xiamen University, Xiamen, 2014.

[6] X. Song and X. Liu, The optimized K-means algorithm based on the new clustering validity function, *Computer Application*, vol.12, no.28, pp.1256-1263, 2008.

[7] A. Banerjee and J. Ghosh, On scaling up balanced clustering algorithms, *Proc. of the 2nd SIAM ICDM*, Arslington, VA, pp.333-349, 2002.

[8] Y. Ye, X. Xia and J. Mo, The research and optimization of the K-means algorithm and the potential function clustering, *Computer Application*, vol.24, no.4, pp.124-135, 2015.

[9] Y. Shi, The analysis of K-means algorithm using mean error criterion function E, *Computer and Information Technology*, vol.3, no.21, pp.978-997, 2004.

[10] X. Zhang, G. Zhang and P. Liu, The optimized K-means algorithm based on the clustering criterion function, *Computer Engineering and Applications*, vol.47, no.11, pp.165-172, 2011.

[11] W. Zhao, H. Ma and Y. Fu, Research on parallel K-means algorithm design based on the Hadoop platform, *Computer Science*, vol.38, no.10, pp.166-176, 2011.

[12] L. Zhou, H. Wang and W. Wang, Parallel K-means algorithm for massive data, *Journal of Huazhong University of Science and Technology (Nature Science)*, pp.150-152, 2012.

[13] R. Jia and Y. Li, Parallel genetic K-means clustering algorithm based on MapReduce model, *Computer Engineering and Design*, vol.2, no.2, pp.31-35, 2014.

[14] W. Cheng, *Parallel Clustering Algorithm on the MapReduce to Achieve*, Zhejiang University, Hangzhou, 2011.

[15] J. Dean and S. Ghemawat, MapReduce: Simplified data processing on large clusters, *Proc. of the 6th International Conference on Operation Systems Design & Implementation (OSDI)*, Berkeley, CA, USA, pp.137-150, 2004.

[16] J. Chang and C. He, K-means acceleration algorithm based on the principle of triangle inequality, *Computer Engineering and Design*, vol.28, no.21, p.11, 2007.

[17] W. Andrew, The anchors hierarchy: Using the triangle inequality to survive high dimensional data, *Proc. of the 16th Conference on Uncertainty in Artificial Intelligence*, pp.156-177, 2000.

[18] X. Jiang, C. Li, X. Zhang and H. Yan, MapReduce parallel implementation of K-means clustering algorithm, *Journal of Huazhong University of Science and Technology*, vol.39, no.1, pp.120-124, 2011.

[19] C. Elkan, Using the triangle inequality to accelerate K-means, *Proc. of the 20th International Conference on Machine Learning*, Washington DC, 2003.