AN EFFICIENT ANALYSIS FOR COMPETITIVE ALGORITHM TO FIND BEST CLUSTERS

Miss Dhanshree Hadawale, Miss Prajakta Mande, Miss Sushama Patil

***Abstract*— Cluster analysis is one of the attractive data mining techniques that have been used in many fields. One of the popular kind of clustering algorithms is the center based clustering algorithm. K-means used as one of the popular clustering method due to its simplicity and high speed in clustering big amount of datasets. However, K-Means has two limitations, K-Means is dependent on the initial state and convergence to local optima in some of the large problems. The *K*-Means objective is a simple, intuitive, and widely-cited clustering objective for data in Euclidean space. However, although many clustering algorithms have been designed with the K-Means objective in mind, very few have approximation guarantees with respect to this objective. we give a streaming algorithm for the K-Means problem. We are not aware of previous approximation guarantees with respect to the *K*-means objective that have been shown for simple clustering algorithms that operate in either online or streaming settings. The Map Reduce programming model, along with its open-source implementation – Hadoop – has provided a cost effective solution for many data intensive applications. Hadoop stores data distributive and exploits data locality by assigning tasks to where data is stored.**

Keywords: K-Means; streaming K-Means; Competitive K-Means; Map Reduce

I. INTRODUCTION

According to the big amount of data in the world, we require new data analysis and extracting information techniques. Therefore, new optimization algorithms are being presented every day. One of the most popular techniques of data analysis is clustering. Clustering can be considered the most important unsupervised learning problem. So, as every other problem of this kind, it deals with finding a structure in a collection of unlabeled data. A cluster is a collection of objects which are similar between them and are dissimilar to the objects belonging to other clusters. Clustering that is the subject of active research in several fields such as data mining, applied in a large variety of applications, like image segmentation, market segmentation, etc. Clustering algorithms can be classified as hierarchical clustering, partition-based clustering, exclusive clustering, and overlapping clustering. One of the most used classes of data clustering algorithms is the center based learning algorithms that follow a simple and easy way to classify a given data set through a certain number of clusters fixed a priori However, K-Means has two shortcomings: dependency on the initial state and convergence Clustering is an exploratory data analysis tool which aims at sorting different objects into groups in a way that the degree of association between two objects is maximal if they belong to the same group and minimal otherwise. Cluster analysis can be used to discover structures in data without providing an explanation [21]. The standard clustering process consists of the following steps:

(1) data preparation and Attribute selection ,
(2) Similarity measure selection,
(3) Algorithm and parameter selection,
(4) Cluster analysis
(5) Validation.

As a consequence, cluster analysis faces new challenges of processing tremendously
large and complex datasets stored and analysed across many computers. Since moving a large amount of data between machines is more costly than moving the computation to the data, a recent trend is to move algorithms, which typically represent a few KB, to process chunks of the dataset independently.The MapReduce approach is a seamless solution to distributed computation that can be used to solve this problem; however, it requiresnew algorithms that can benefit from MapReduce technology.In the Map reduce model,applications consist of map and reduce tasks.The input and output data of these tasks are split and stored on a distributed file system (dfs) with the granularity of data block. Fault tolerance is achieved through the creation and the random placement of data block replicas.
One of the fundamental concepts that guided the design of Map Reduce and Hadoop is moving computation to data. Exploring locality increases throughput because network bandwidth is always the bottleneck for large scale systems.Running map task on a node that contains the input data (data-local execution) is a primary objective of scheduling, but when this is impossible, running on the same rack (rack-local execution) is preferred to running off-rack.

Manuscript received March, 2015
***Dhanshree Hadawale**, Information Technology, Pune University/ ALCOE/ Pune, India, 7738539245*
***Prajakta Mande,**, Information Technology, Pune University/ ALCOE/ Pune, India, 8180889804*
***Sushama Patil**, Information Technology, Pune University/ ALCOE/ Pune, India,8082323157*

ISSN: 2278 – 1323

II THEORY & BACKGROUND

2.1 K-Means

K-means is a data mining algorithm which perform clustering k-means algorithm determines spherical shaped cluster, whose center is the magnitude center of points in that cluster, this center moves as new points are added to or detached from it. This proposition makes the center closer to some points and far apart from the other points, the points that become nearer to the center will stay in that cluster, so there is no need to find its distances to other cluster centers. The points far apart from the center may alter the cluster, so only for these points their distances to other cluster centers will be calculated, and assigned to the nearest center.

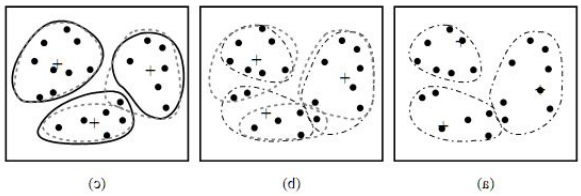

Fig.2.1 Example of K-means Algorithm

K-means algorithm uses an iterative process in order to cluster database. It takes the number of desired clusters and the initial means as inputs and produces final means as output. Mentioned first and last means are the means of clusters. After termination of K-means clustering, each object in dataset becomes a member of one cluster. This cluster is determined by searching all over the means in order to find the cluster with nearest mean to the object. Shortest distanced mean is considered to be the mean of cluster to which observed object belongs. K-means algorithm tries to group the items in dataset into desired number of clusters. To perform this task it makes some iteration until it converges. The K-Means algorithm is simple and straightforward: first, it randomly selects *k* points from the whole dataset.These points represent the initial centroids (or *seeds*). Each remaining point in the dataset is assigned to a cluster whose centroid is closest to that point. The coordinates of the centroid are then recalculated The accuracy of large dataset cluster analyses using K-Means depends on the accuracy of centroid initialisation methods that are adapted to datasets distributed across several machines.

Advantages of K-mean Clustering:

• K-mean clustering is simple and flexible.
• K-mean clustering algorithm is easy to understand and implements.

Disadvantages of K-mean Clustering:

• In K-mean clustering user need to specify the number of cluster in advance.
• K-mean clustering algorithm performance depends on a initial centroids that why the algorithm doesn't have guarantee for optimal solution.

2.2 SK-Means

In which the algorithm is allowed to output more than *k* centers, and a streaming clustering algorithm in which batch clustering algorithms are performed on small inputs (fitting in memory) and combined in a hierarchical manner. Empirical evaluations on real and simulated data reveal the practical utility of our method. S Kmeans follow a divide-and-conquer strategy, whereby the dataset is partitioned into smaller subsets, and analyses are done in parallel to each partition. defines a divide-and-conquer strategy to combine multiple bi-criterion approximation algorithms for the *k*-medoid problem to yield a one-pass streaming approximation algorithm for *k*-median. with the goal of yielding a streaming clustering algorithm with a constant approximation to the *k*-means objective.

The algorithm divides the input into *m* equal-sized groups in step 1. In step 2, the algorithm runs in each group a variant of K-Means++ that selects $3 \times log(k)$ points in each iteration a total of k times (traditional K-Means selects only a single point). In step 4, the algorithm weights the *m* sets of points. The process of weighting the points is not clearly specified in Ailon et al.
(2009), and the authors provide noguidelines on how to divides the input into *m* equal-sized groups in step 1.

Algorithm :Streaming divide-and-conquer clustering

Input: A set of data points *M;* the number of clusters *k* and the number of partitions,*x.*
A,SK-Means modified to select 3log(*k*) points per iteration.
A', SK-Means selecting one point per iteration.

Output: *X* points grouped into *k* clusters and respective final centroids *FC*
1: Partition *M* into *M*1, *M*2, ..., *Mm*
2: For each $i \in \{1, 2, \ldots, x\}$ do
3: Run *A* on *Mi* to get $3k \times \log(k)$ centroids
Ti = {*ti*1, *ti*2, …}
4: Denote the induced clusters of *Mi* as *Si*1 ∪ *Si*2 ∪ …
5: Sw ← *T*1 ∪ *T*2 ∪ .. ∪ *Tm*
6: Run *A′* on *Sw* to get *k* centroids *C*
7: Run K-Means on *M* using the set of initial centroids *FC*

Divides the input into *m* equal-sized groups in step.We extend their analysis to the *k*-means problem and then use S*k*-means.in the divide-and-conquer strategy, yielding an extremely efficient streaming algorithm with an approximation guarantee. Empirical evaluations, on simulated and real data, demonstrate the practical utility of our techniques.

2.3 CK-Means

We extend their analysis to the *k*-means problem and then use S*K*-means and C*K*-means in the divide-and-conquer strategy, yielding an extremely efficient streaming algorithm with an $O(c!\ \log(k))$-approximation guarantee, where *! !* log *n/* log*M*, *n* is the number of input points in the stream and *M* is the amount of work memory available to the algorithm. Empirical evaluations, on simulated and real data, demonstrate the practical utility of our techniques.

By using our new CK-Means, we pick initial centroids based on a probability distribution according to the distance of the centroids. Thus, we do not favour or exclude any specific regions that are pre-mapped in the clustering

space, neither we assume any particularly structure in the data. Therefore, if a dataset has clusters situated in the opposite corners of the diagonal.

Algorithm : Our new CK-Means

Input: A set of data points *X* shuffled into a random order*;* the number of centroids *k;* the number of competitors *m* and a fitness measure *f*
Output: *X* points grouped into *k* clusters and respective final centroids *CF*
1: Partition *X* into *x*1, *x*2, ..., *xm*
2: For each $i \in \{1, 2, \ldots, m\}$ do
3: Run K-Means++ on *xi* to get *k* centroids *ICi* and;clusters *clxi*
4: $Si = f(clxi)$
5: $C \leftarrow ICi$, where $i \leftarrow Best - fit(Si)$
6: Run K-Means on *X* with *C* as initial centroids to obtain cluster analysis output

---

Our new CK-Means reduces the running time compared with SK-Means. Several runs of our new algorithm using the same initial conditions produces results with less variance and better quality than the SK-Means. Thus our new algorithm improves the seeding compared with SK-Means. Our new approach reduces the execution time of the SK-Means by performing several cluster analysis instances of SK-Means over subsets of the dataset in parallel. The result of each cluster analysis instance on a subset of the dataset is then scored using a fitness measure. The SK-Means instances compete with each other, and the winner is the instance with the best-fit cluster analysis. The set of initial centroids (*IC)* of the winner SK-Means is then used as the initial centroids' *(IC)* for K-Means cluster analysis over the entire dataset.

Our CK-Means picks points in those regions with high probability as long as the centroids are far away from each other. Making assumptions about the structure of datasets in advance might not be a major problem for clustering small and simple datasets. In a Big Data scenario this is usually not the case, therefore our new algorithm is more suitable for Big Data.

2.4 Map Reduce

Hadoop (White, 2010) provides a distributed file system and a framework for the analysis and transformation of very large datasets using the MapReduce (Dean and Ghemawat, 2008) programming model. The Hadoop distributed file system (HDFS) (Shvachko et al., 2010) is designed to reliably store very large datasets across several systems.HDFS is suitable for storing very large files (up to TBs in size) to be processed in a write-once, read-many-times pattern. A typical MapReduce job involves reading a large proportion, if not all, of the dataset; so the time required for reading the whole dataset is more important than the latency incurred in reading the first record.

Algorithm : Map Reduce

Input: A HDFS path to the stored data points and the number of clusters *k*
Output: *X* points grouped into *k* clusters and respective centroids *C*
1: Run MapReduce seeder on HDFS path to get *k* centroids *C*
2: Run K-Means_MR on HDFS path with *C* as initial centroids and get *CLx1*
3: Run SK-Means_MR on HDFS path with *C* as initial centroids and get *CLx2*
4: Run CK-Means_MR on HDFS path with *C* as initial centroids and get *CLx3*.

The MapReduce platform launches several map tasks, each one processing a partition *xi* in the machine where it is stored. At the end of the computation, each map emits to a reducer a candidate solution with a set of the initial centroids and a fitness score.

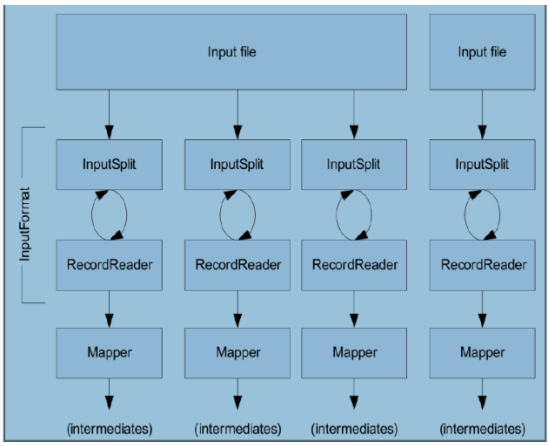

Fig .1. The MapReduce Flow: The Mapper

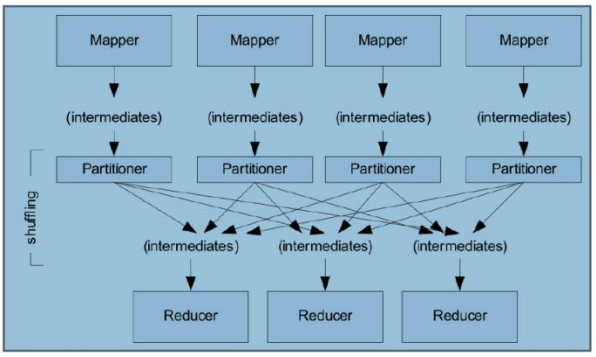

Fig .2.The MapReduce Flow: Shuffle and Sort

ISSN: 2278 – 1323

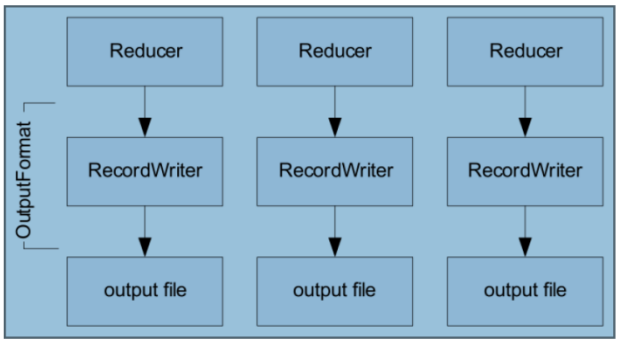

Fig.3. The MapReduce Flow: Reducers to Outputs

The reducer chooses the fittest set of centroids.as produces output file.

III COMPARISON

The volume of data generated daily is growing at an exponential rate. This growth is due in part to the proliferation of sensors and the increase in resolution of those sensors. To distil meaningful information from this growing mountain of data, there is a growing need for advanced data analysis techniques, such as cluster analysis. Clustering is a key factor in the Big Data problem. In a Big Data context it is not feasible to 'label' large collection of objects. . In this paper,we propose a new parallel seeding algorithm named competitive K-Means (CK-Means) that addresses problems affecting serial K-Means and Sk-Means. We also propose an efficient MapReduce implementation of our new CK-Means that we found scales well with large datasets.

3.1 K-Means

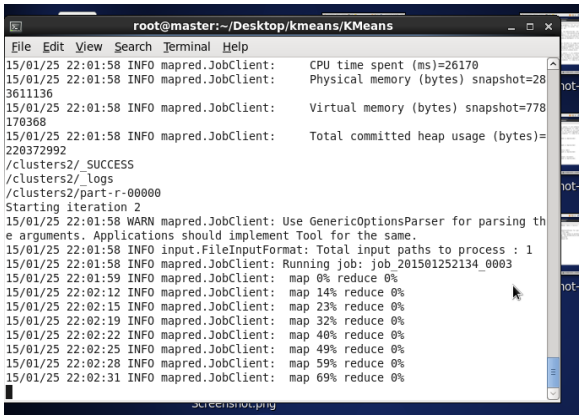

Fig 3.1 Execution of K-Means

The k-means algorithm is well known for its efficiency in clustering large data sets. k-means clustering algorithm, requring a simple data structure to store some information in every iteration,which is to be used in the next interation.The improved method avoids computing the distance of each data object to the cluster centers repeatly, saving the running time.As working only on numeric values prohibits it from being used to cluster real world data containing categorical values.

3.2 SK-Means

We observed two major problems when we applied SK-Means seeding method to large datasets. First, SK-Means is a stochastic algorithm, which means that the results produced by SK-Means were considerably different across several analysis runs using the same initial conditions. We observed that the difference in the results grows as the dataset contains more points and has higher feature dimensionality.

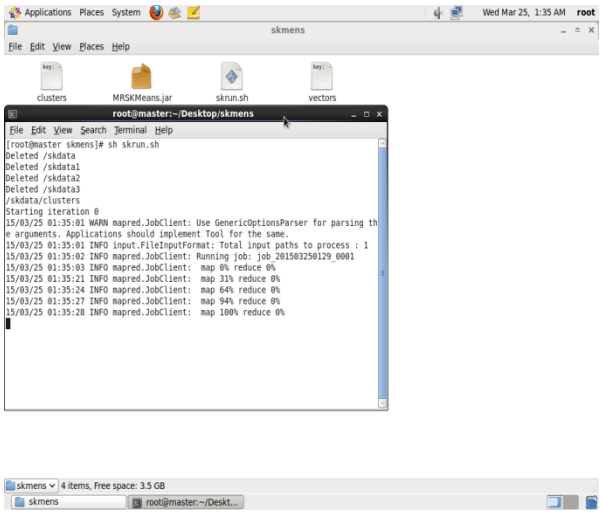

Fig 3.2 Execution of SK-Means

3.3 CK-Means

Here we use a new search "Competitive Algorithm1" to find the best clusters with best numbers of clustering. In this algorithm we granted each clustering solution with special clusters number and use a new time function to calculate the clustering analysis in each and every step, We compared proposed algorithm with new algorithm in clustering techniqu,such as K-means,SK-Means by implementing them on several well-known datasets. Our observation show that the proposed algorithm works better than the others according to time function.

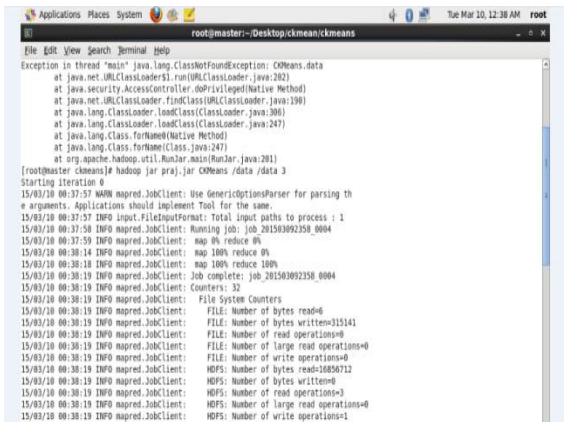

Fig 3.3 Execution of CK-Means

IV.CONCLUSION

Data mining in recent years with the database and artificial intelligence developed a new technology, its aim the large amount of data ,to achieve the effective consumption of data resources. As one important function of data mining, clustering analysis either as a separate tool to discover data sources distribution of information,as well as other data mining algorithms as a preprocessing step, the cluster analysis has been into the field of data mining.K-Means is a fast clustering method. However, initial seeding heavily influences the cluster quality. In this paper, we presented a new strategy to parallelise SK-Means that improves the speed and accuracy of cluster analysis for large datasets.our new CK-Means consistently improves cluster analysis accuracy compared with K-Means and SK-Means.
MapReduce largely decreased the running time. We found that our new algorithm scales with the dimension of the dataset. The running time is more sensitive to variations in the number of data points and of clusters than to variations in the number of dimensions.With our findings, we have addressed the problem of finding a good initial seeding in less time. Thus performing accurate cluster analysis over large datasets can now be done by using our new CK-Means approach.

REFERENCES

[1] A new approach for accurate distributed clustermanalysis for Big Data: competitive K-Means.Rui Máximo Esteves*,Thomas Hacker, Chunming Rong.

[2] Streaming *k*-means approximation, Nir Ailon, Ragesh Jaiswal, Claire Monteleoni,

[3] Ankit Aggarwal, Amit Deshpande and Ravi Kannan: Adaptive Sampling for k-means Clustering. APPROX, 2009.

[4] Enhanced K-Means Clustering Algorithm to Reduce Time Complexity for Numeric Values Bangoria Bhoomi M. PG Student Noble Engineering College, Junagadh, Gujarat.

[5] Nir Ailon and Edo Liberty: Correlation Clustering Revisited: The ”True” Cost of Error Minimization Problems. To appear in ICALP 2009.

[6] An efficient cost function for imperia list competitive algorithm to find best clusters
mojgan ghanavati, 2 mohamad reza gholamian, 3 behrouz minaei, 4 mehran davoudi.

[7] Noga Alon, Yossi Matias, and Mario Szegedy.: The space complexity of approximating frequency moments. In Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing,pages 20–29, 1996.

[8] David Arthur and Sergei Vassilvitskii: Worst-case and smoothed analyses of the icp algorithm, withban application to the k-means method. FOCS, 2006

[9] David Arthur and Sergei V assilvitskii: *k*-means++: the advantages of careful seeding. SODA, 2007.

[10] [AGKM+04] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit: Local search heuristics.

[11] for k-median and facility location problems. Siam Journal of Computing, 33(3):544–562, 2004.

[12] Niknam, T., Fard, E.T., Pourjafarian, N. and Rousta, A. (2011) ‘An efficient hybrid algorithm based on modified imperialist

[13] competitive algorithm and K-Means for data clustering’, Engineering Applications of Artificial Intelligence, Vol. 24, No. 2, pp.306–317, doi:10.1016/j.engappai.2010.10.001.

[14] Ostrovsky, R. and Rabani, Y. (2006) ‘The effectiveness of lloydtype methods for the k-means problem’, in 47th IEEE Symposium on the Foundations of Computer Science (FOCS), pp.165–176

[15] White, T. (2010) *Hadoop: The Definitive Guide*, 2nd ed., O’Reilly Media/Yahoo Press. Zaharia, M., Konwinski, A., Joseph, A.D., Katz, R. and Stoica, I.

[16] ‘Improving Map Reduce performance in heterogeneous environments’, in Proceedings of the 8th USENIX Conference on Operating Systems Design and Implementation, pp.29–42, USENIX Association, San Diego, California.

[17] J. Han and M. Kamber. Data Mining: Concepts and Techniques. Morgan Kaufmann, 2000Jain, M. Murty and P. Flynn, “Data Clustering: A Review”, *ACM* Computing Surveys, Vol.31, No. 3, Sep 1999, pp. 264–323.

[18] Garbriela derban and Grigoreta sofia moldovan, “A comparison of clustering techniques in aspect mining”, Studia University, Vol LI, Number1, 2006, pp 69-78.

[19] fahim a.m., salem a.m., torkey f.a., ramadan m.a.” An efficient enhanced k-means clustering algorithm” J Zhejiang Univ SCIENCE A 2006 7(10):1626-1633

ISSN: 2278 – 1323