Volume 4, Issue 5, May 2014 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Enhancing K-means Clustering Algorithm and Proposed Parallel K-means Clustering for Large Data Sets

Dr.Urmila R. Pol
Department Of Computer Science, Shivaji
University,Kolhapur.

***Abstract:** Clustering is one of the wide field of data mining. Cluster analysis is one of the significant data analysis method and the k-means clustering algorithm is widely used for many practical applications[1]. But the original k-means algorithm is computationally expensive and the quality of the resulting clusters heavily depends on the selection of initial centroids . In clustering data elements having similarities are placed in respective groups, is a well known problem.[2] Our contribution in this paper is the development of a parallel version of the k-means algorithm. This paper proposed parallel algorithm for clustering using MPI for passing message base in the Master-Slave based structural model.*

***Keywords:** K-means clustering, centroids, large data set.*

I. Introduction

Clustering is identified as a key technique in data mining. K-means clustering is a partitioning clustering technique in which clusters are formed with the help of centroids. On the basis of these centroids, clusters can vary from one another in different iterations. Moreover, data elements can vary from one cluster to another, as clusters are based on the random numbers known as centroids. It is obviously able to improve efficiency when using parallel data mining. In this paper, in order to achieve high-performance parallel computing, there is an algorithm which using Master-Slave structure and communicate by MPI between the hosts, make full use of the resources under the cluster environment.

1.1 The K-Means Clustering Algorithm

This section describes the original k-means clustering algorithm. The idea is to classify a given set of data into *k* number of disjoint clusters, where the value of *k* is fixed in advance. The algorithm consists of two separate phases: the first phase is to define *k* centroids, one for each cluster. The next phase is to take each point belonging to the given data set and associate it to the nearest centroid. Euclidean distance is generally considered to determine the distance between data points and the centroids. When all the points are included in some clusters, the first step is completed and an early grouping is done. At this point we need to recalculate the new centroids, as the inclusion of new points may lead to a change in the cluster centroids. Once we find *k* new centroids, a new binding is to be created between the same data points and the nearest new centroid, generating a loop. As a result of this loop, the *k* centroids may change their position in a step by step manner. Eventually, a situation will be reached where the centroids do not move anymore. This signifies the convergence criterion for clustering [2]. In this algorithm mostly Euclidean distance is used to find distance between data points and centroids [3]. Pseudocode for the k-means clustering algorithm is described in Algorithm. The Euclidean distance between two multi-dimensional data points X = (x1, x2, x3... xm) and Y = (y1, y2, y3... ym) is described as follows:

```
Algorithm : Algorithm 1: The k-means clustering algorithm [1]
Input:    D = {d1, d2,......,dn} //set of n data items.
                  k // Number of desired clusters
          Output: A set of k clusters.
          Steps:
                  1. Arbitrarily choose k data-items from D as initial centroids;
                  2. Repeat
                          Assign each item di to the cluster which has the closest centroid;
                          Calculate new mean for each cluster;
                          Until convergence criteria is met.
```

1.2 Proposed K – Means Clustering Algorithm

This section proposes a variation in the original K – means algorithm. In traditional K – means one randomly selects the number of clusters and the centroids. The formation of final clusters and iterations required depends on how one selects initial centroids which is computationally expensive and requires time proportional to the product of the number of data items, number of clusters and the number of iterations. In the proposed variation, number of clusters are fixed to be three and the initial centroids are initialised to minimum, maximum and the N/2th data point of the total data set. The notations and the algorithm are described below.

Table: Notation for Sequential Algorithm

Keywords	Description
K	Number of clusters
N	Number of data points
D	Data set
di	Data point in data set
xi	Attribute of data point
Ck	Initial centorids, centroid of previous iteration
n	Number of data points in cluster

Algorithm : Variation in Sequential K – Means

Data set D = {d_1, d_2, ………….. , d_n}
Each data point consist of n attributes such as di = {x1, x2, ………….. , xn}
Number of clusters K = 3

1. Initially sort given data using quick sort
2. Initialise K cluster centroids as
   C1 = {x1min, x2min, …………………… , xnmin}
   C2 = D(N/2) = {d(n/2)x1, d(n/2)x2, ………………… , d(n/2)xn}
   C3 = {x1max, x2max, …………………… , xnmax}
3. Calculate distance from each di to each Ck using Euclidean distance formula given as follows

$$\text{dist(di, Ck)}= \sqrt{\sum_{i=0}^{n}(dixi - Ckxi)^2}$$

4. Assign data points to the clusters having minimum distance from the centroid.
5. Calculate new cluster mean i.e. Cknew using formula given as follows
   Cknew = $\frac{1}{n}\sum_{i=0}^{n} di(x1, x2 \dots\dots\dots\dots\dots\dots, xn)$
6. If for each Cknew = Ck then a uniqur cluster is found else repeat from step 3.

1.3 K – Means Example

Problem: Cluster the following eight points (with (x1, x2) representing locations) into three clusters A1(2, 10) A2(2, 5) A3(8, 4) A4(5, 8) A5(7, 5) A6(6, 4) A7(1, 2) A8(4, 9).

Initial data points are:
d1=(2,10) d2=(2,5) d3= (8,4) d4=(5,8)
d5=(7,5) d6=(6,4) d7=(1,2) d8=(4,9)

Sorted data points are:
d1=(1,2) d2=(2,5) d3=(2,10) d4=(4,9)
d5=(5,8) d6=(6,4) d7=(7,5) d8=(8,4)

Initial centroids are:
C1 = (x1min, x2min) = (1,2)
C2 = ($d_{(n/2)}$x1, $d_{(n/2)}$x2) = (4,9)
C3 = (x1max, x2max) = (8,10)
Distance formula :

Dist(di, Ck) = $\sqrt[2]{(dix1 - Ckx1)^2 + (dix2 - Ckx2)^2}$

Iteration 1:

Centroid/data points	d1	d2	d3	d4	d5	d6	d7	d8
C1	0	3.1	8.06	7.61	7.21	5.38	6.70	7.28
C2	7.61	4.47	2.23	0	1.41	5.38	5	6.40
C3	10.63	7.81	6	4.12	3.60	6.32	5.09	6

Cluster 1	Cluster 2	Cluster 3
(1,2)	(2,10)	(8,4)
(2,5)	(4,9)	
(6,4)	(5,8)	
	(7,5)	

C1new = (1+2+6)/3, (2+5+4)/3 = (3, 3.67)
C2new = (2+4+5+7)/4, (10+9+8+5)/4 = (4.5, 8)
C3new = (8, 4)

Iteration 2:

Centroid/data points	d1	d2	d3	d4	d5	d6	d7	d8
C1	2.60	1.66	6.40	5.42	4.46	3.01	4.21	5.01
C2	6.94	3.90	3.20	1.11	0.5	4.27	3.90	5.31
C3	7.28	6.08	8.48	6.40	5	2	1.41	0

C1	C2	C3
(1,2)	(2,10)	(6,4)
(2,5)	(4,9)	(7,5)
	(5,8)	(8,4)

C1new = (2+1)/2, (5+2)/2 = (1.5, 3.5)
C2new = (2+5+4)/3, 10+9+8)/3 = (3.66, 9)
C3new = (8+7+6)/4, (4+5+4)/4 = (7, 4.33)

II. Parallel K – Means Algorithm

The algorithm assumes shared-nothing architecture and Master-slave processor model where each of processor has private memory and a private disk. The processors are connected by a communication network and can communicate only by passing messages . The communication primitives used by our algorithms are part of the MPI (Message Passing Interface) communication library . We can accomplish the parallelism of k-means in different ways; at instructional level or at data level or control level . We are following data level parallelism. Divide given data points into N partitions by Master Processor . Each partition will be assigned to every processor. Master processor calculates K centroids and broadcast to every processor.Now each processor calculates new centroids and broadcast to Master processor. Master processor recalculates global centroids and broadcast to every processor. Repeat these steps until unique cluster is found.

The proposed parallel K – Means algorithm is given as follows.

Keywords	Description
K	Number of clusters
N	Number of data points
P	Number of processors
D	Data set
di	Data point in data set
xi	Attribute of data point
Ck	Initial centorids, centroid of previous iteration
n	Number of data points in cluster

Table: Notation for Parallel Algorithm

2.1 Algorithm : Parallel K – Means

Data set D = {d_1, d_2, ………….. , d_n}
Each data point consist of n attributes such as di = {x1, x2, ………….. , xn}
Number of clusters K = 3
Number of processors P

1. Master processor assigns N/P data points to all other processors
2. Master processor calculates K centroids as follows and broadcast it to other processors
   C1 = {x1min, x2min, …………………… , xnmin}
   C2 = D(N/2) = {d(n/2)x1, d(n/2)x2, ………………… , d(n/2)xn}
   C3 = {x1max, x2max, …………………… , xnmax}
   Now each processor,

3. Calculate distance from each di to each Ck using Euclidean distance formula given as follows

$$\text{dist(di, Ck)}= \sqrt{\sum_{i=0}^{n}(dixi - Ckxi)^2}$$

4. Assign data points to the clusters having minimum distance from the centroid.
5. Calculate new cluster mean i.e. Cknew using formula given as follows

   Cknew = $\frac{1}{n}\sum_{i=0}^{n} di(x1, x2 \dots\dots\dots\dots\dots\dots, xn)$
6. Each processor broadcast local centroids to master processor
7. Master processor recalculates global centroids for each cluster and broadcast global centroids to all other processors.
8. Repeat from step 3 until unique cluster is found.

III. Conclusion

In this paper, we recommended variation in K-means algorithm and proposed parallel K-means clustering algorithm. The k-means algorithm is computationally very expensive. The proposed algorithm is found to be more accurate and efficient compared to the original k-means algorithm. The proposed algorithm produces the more accurate unique clustering results .We have contended that to make data mining practical for common people, data mining algorithms have to be efficient and data mining programs should not require dedicated hardware to run. On these fronts, we can conclude from that, Parallelization is a viable solution to efficient data mining for large data sets.

REFERENCES

1. K. A. Abdul Nazeer, M. P. Sebastian” Improving the Accuracy and Efficiency of the k-means Clustering Algorithm” Proceedings of the World Congress on Engineering 2009 Vol I WCE 2009, July 1 - 3, 2009, London, U.K.
2. Azhar Rauf, Sheeba, Saeed Mahfooz, Shah Khusro and Huma Javed “Enhanced K-Mean Clustering Algorithm to Reduce Number of Iterations and Time Complexity” Middle-East Journal of Scientific Research 12 (7): 959-963, 2012
3. K. A. Abdul Nazeer and M. P. Sebastian, “Improving the accuracy and efficiency of the k-means clustering algorithm,” in International Conference on Data Mining and Knowledge Engineering (ICDMKE), Proceedings of the World Congress on Engineering (WCE-2009), Vol 1, July 2009, London, UK.