Dataset Clustering Using K-Means Algorithm

[1]Mayur Kulkarni, [2]Karan Jamdade, [3]Krushna Kashid, [4]Akash Dhotre

[1,2,3,4] Dept of Computer Engineering NBN Sinhgad School of Engineering, Pune, India

Abstract: **In this paper, we consider the clustering of very large distributed datasets over a network using a decentralized K-means algorithm. Analysis of this data and identifying clusters is challenging due to processing, storage, and transmission costs. Many algorithms has been invented for distributed data clustering, which are applied when datasets cannot be concentrated on a single machine, for instance because of some reasons or due to net-work bandwidth limitations or because of the big amount of distributed data. Low overhead analysis of huge distributed data sets is must for current data centers and for future sensor networks. Our experimental evaluations show that dataset Clustering using K-means can discover the clusters more efficiently with scalable transmission cost, and also expose its supremacy in compare to the popular method LSP2P.**

Keywords: **Distributed systems, clustering, dynamic system, partition-based clustering, and density-based clustering.**

1. INTRODUCTION

Clustering is important for analyzing large data sets. Clustering partitions data into groups of similar objects, each cluster is described concisely using a summary. Distributed core database technology has been an active research of area for decades; distributed data analysis and mining have been researched since the early nineties motivated by issues of scalability, bandwidth, privacy, and cooperation among competing data owners. Analyzing this data, using centralized processing, is often infeasible due to communication, storage and computation overheads, Several algorithms have been developed for distributed data clustering. A general scheme for all approaches is to first locally extract suit-able aggregates, and then send aggregates to a middle site where they are processed and combined into a global model. In distributed clustering algorithms, the data set as a full remains dispersed, and the participating distributed processes will gradually discover various clusters. The kind of aggregates and combine algorithm depend on the data types and the distributed environment under consideration.

In this paper we investigate decentralized K-means clustering,where many processing nodes shipped with their own local dataset and instantiated with two popular partition-based and density-based clustering methods over cloude is used.

2. SYSTEM MODEL

Suppose we have n networked nodes N={ n1,n2….n}. Each node 'n' is stored and shared a set of data items which is also called as internal data. Internal data is denoted as D_n^{int}**.**It can be changed time to time. Each data item d has an attributes that is metadata.This vector is denoted as d_{atr}. If transmission of data item is required then transmission of attribute vector is necessary. While discovering cluster, nodes in network can share attribute vector of data items. These items are called as external data of n and denoted as D_n^{ext}.

Union of internal and external data :

$$D_n = D_n^{int} \text{ U } D_n^{ext}$$

While execution of algorithm, each node n makes a summarized view of D by maintaining representatives.

$$R_p = \{r_1^n ,r_2^n , . . .\}$$

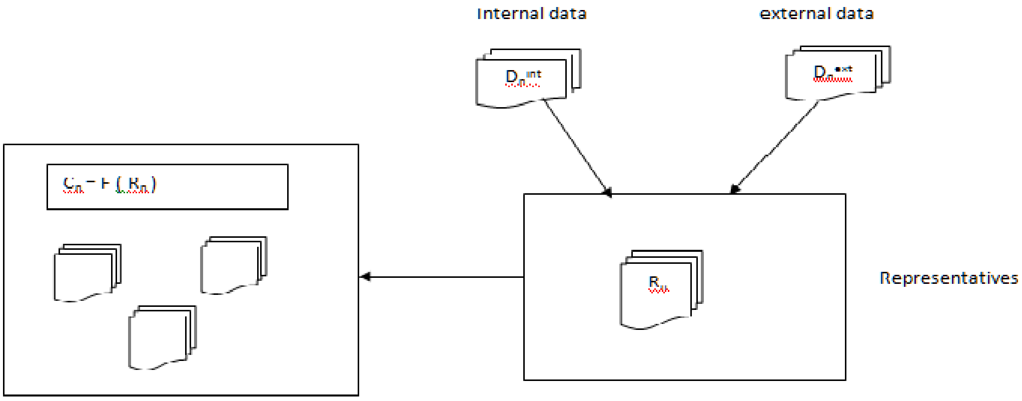

Fig . System model

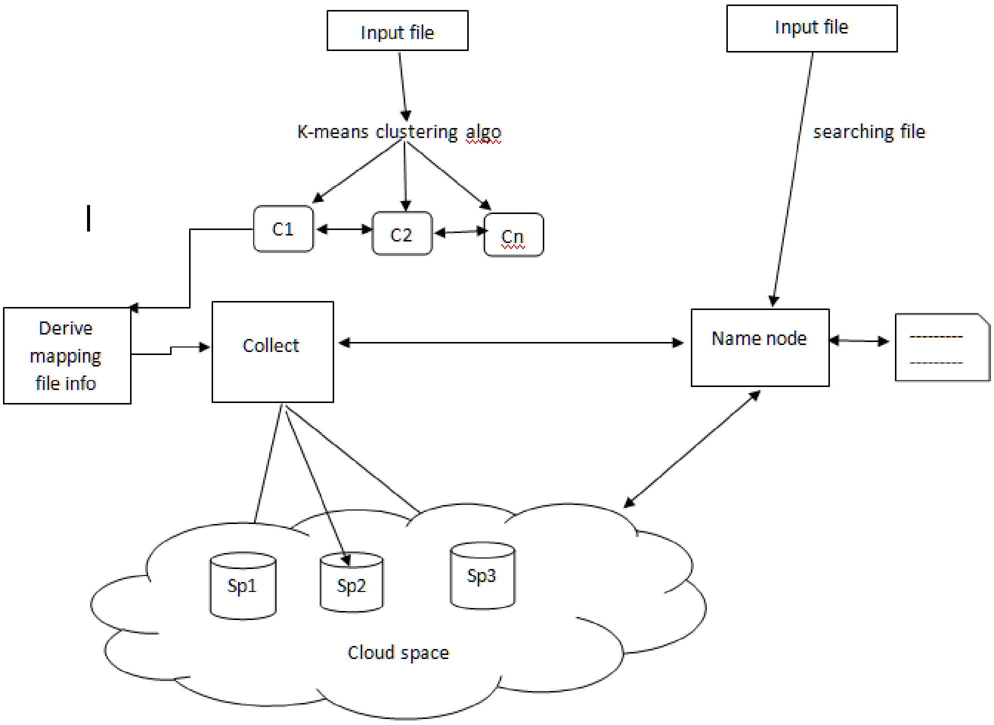

Fig. Overall System Architecture

3. DECENTRALIZED CLUSTERING

Each node unspectacularly makes a summarized view of D, on which it will execute the clustering algorithm F. In following section we will discuss on how we use clustering algorithm.

Each node n is responsible for deriving accurate representatives for part of the data set located near D_n^{int}. For other parts, it solely collects representatives. Accordingly, it slowly builds a global view of D. Each node is performed two tasks in parallel: 1) Representative derivation, which we name DERIVE and 2) representative collection, which we name COLLECT. To derive representatives for part of the data set locate d near D_n^{int}, n must have an accurate and up-to-date view of the data located around each data . In each round of the DERIVE task, each node n selects another node q for a three-way information exchange. It will first send D to node q. If size of D_n^{int} is large, it can summarize the internal data by an arbitrary method such as grouping the data with clustering and sending one data from each group. Node n then receives from q, data items located in radius r of each d . based on a distance function d. r is a user-defined threshold, which will be adjusted as p continues to discover data.

In the same manner, it will also send to q the data in Dn that lie within the r radius of data in D. The operation update Local Data uses to add the received data to D_n^{ext}.

3.1 K-means Clustering:

k-means clustering is a type of vector quantization , originally from signal processing, that is popular for cluster analysis in data mining. *k*-means clustering goals to partition *n* observations into *k* clusters in which each observation belongs to the cluster with the closest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells.

The algorithm has a loose relationship to the *k*-nearest neighbor classifier, a popular machine learning method for classification that is often confused with *k*-means because of the *k* in the name. One can apply the one-closest neighbor classifier on the cluster centers obtained by*k*-means to classify new data into the existing clusters. This is known as nearest centroid classifier or Rocchio algorithm.

3.2 Applications of K-Means Clustering:

K-means clustering in specific when using heuristics such as Lloyd's algorithm is rather easy to implement and apply even on huge data sets. As such, it has been successfully used in various topics, including market segmentation, computer vision, geostatistics and agriculture. It often is used as a preprocessing step for other algorithms, for example to find a starting configuration.

3.3 Adventages of Dataset Clustering:

1) Fast, strong and easier to understand.

2) Relatively efficient.

3) Provides good result when data set are distinct or well separated from each other.

4. DYNAMIC DATA SET

Real-world distributed systems changes continuously, because of nodes fetching and defetching the systems, or because their set of internal data is modified.

To refer freshness, of data, each data item will have an associated age denotes the time that node believes has passed. Time is measured in terms of gossiped rounds. The age of an internal data always remains zero to reflect that it is stored at its owner. If a node p receives a copy d0 of a data item d it already stores. When a data item d is removed from the original peer, the minimal recorded age among all its copies will only increase. Node p can remove data item d if age p(d) >MaxAge, where MaxAge is some threshold value, presuming that the original data item has been deleted. The weight of a data item is a function of its age. For a data item d, the weight function is ideally one for all age values not greater than MaxAge. The data items summarized by a representative have different lifetimes according to their age. Therefore, the weight of the representative should capture the number of data items summarized by the representative at each age value. When the weight value falls to zero, the representative can be safely removed.

5. ENHANCEMENTS

In this section number of improvements to the basic algorithm that will be discussed here. **Storage** nodes can have limited storage, processing and communication resources. Multiple files can be cluster at a time that will be weakness of our concept and this will be enhanced in future.

6. DENSITY-BASED CLUSTERING

In density-based clustering, a cluster is a set of data objects scattered in the data space over a adjacent region of high density of objects. Density-based clusters are segregated from each other by adjacent regions of low density of objects. Data objects located in minimum-density regions are typically considered noise or outliers Data clustering is serving to describe the data mining task which aims at partitioning a usually involving a number of distinct data sets into groups, such that the data objects in single group are similar to each other and are different from those in other groups. Therefore, a clustering K-means algorithm is a mapping from any data set of objects to a clustering of a collection of subsets. Clustering techniques inherently pivoted on the notion of distance between data objects to be grouped.

7. PARTITION-BASED CLUSTERING

In partition-based clustering the big Data set partitioned into several sets of data objects such a way that the similarity will be maintained. In Partition based clustering if data is big then that will be partitioned in similar group of objects.

8. CONCLUSION

In this way we have studied first k-means clustering algorithm efficiency of it and how it work. The solution is present for unstructured data. But we have used same concept for structured data. For unstructured data we can easily use k-means algorithm for creating cluster and searching data. But this same concept is not available for structured data. Hence we have applied same concept to structured data in row and column format. Due to that overall searching speed will increased.

REFERENCES

[1] S. Lodi, G. Moro, and C. Sartori, “Distributed data clustering in multi-dimensional peer to peer networks, 2010, vol. 104.

[2] J. Fellus, D. Picard, and P.-H. Gosselin, “Decentralized k-means using randomized gossip protocols for clustering large datasets,” 2013, pp. 599–606.

[3] Eyal, I. Keidar, and R. Rom. “Distributed data clustering in sensor networks,”vol. 24, no. 5, pp. 207–222, 2011.

[4] Guha, S., Rastogi, R., and Shim,.“K.: CURE: An efficient clustering algorithm for large databases” Information System Journal, (2001) 35-58

[5] Jain, A., and Dubes, R. “Algorithms for Clustering Data” Pentice-Hall advanced reference series. Prentice-Hall(1988).

[6] Jain, A., Murty M., Flynn, “Data Clustering: A Review. ACM Computing Surveys,”Vol 31, No. 3, September (1999). 264-322