Comparative Study of Various Distributed Clustering Algorithms

Amitkumar Kishorbhai Dudhaiya[1] Bakul Panchal[2]
[1] Student [2]Assistant Professor
[1] Department of Computer Science & Engineering [2]Dept. of Information Technology
[1]Govt. Engg. College Modasa, Gujarat (India) [2]L. D. College of Engineering Ahmedabad, Gujarat

Abstract---Clustering is the process of data segmentation in which data is grouped together based on similarity to form a cluster. These clusters are very useful to extract interesting patterns from large data. But as application grows it generates large amount of data. The growing need for distributed clustering algorithms is required to handle huge size of databases that is common nowadays. Normal (Centralized) clustering Algorithms lags when we have large amount of data. In this paper gives general classification and Analysis of Traditional Distributed Clustering Algorithms.

Keywords: Distributed Data Mining, Distributed Clustering DBDC Algorithm

I. Introduction

Clustering has become an increasingly important task in modern applications such as marketing, Bioinformatics, Spatial Analysis, molecular biology as well. In all these applications generates large amount of data. On other hand data are originally collected at different sites. These lead us to the requirement of new branch of data mining known as DDM (Distributed Data Mining).

In order to extract information out of these data, they are brought together and then clustered [1]. Typical centralized clustering algorithms cluster a data set stored in a single site. It is impossible to transfer all different sites data to a single site. In distributed environment how to perform clustering is very challenging problem. We required sending only Limited data to the central site for Cluster analysis. For Example WAL-MART featuring the largest civil database in the world, consisting of more than 200 terabytes of data [4]. Every night all data is transmitted to Bentonville from the different stores via the largest privately hold satellite system. Such a company would greatly benefit, if it were possible to cluster the data locally at the stores and then determine and transmit all suitable local representatives which allow reconstructing the complete clustering at the central in Bentonville [4]. The transmission of huge amounts of data from one site to another central site is in some application areas almost impossible. To eliminate these problems first parallel versions of different clustering algorithms introduced. To cluster large data on single site processor and memory power of a single machine is not sufficient. Se we apply to run these clustering algorithms parallel on to many processors with large memory. This is same as Parallel Processing. The parallel version of one of the famous clustering algorithm DBSCAN is developed. This parallel algorithm firstly uses R*- tree to organize data in the central site, then stores the preprocessed data in each sub site, which communicate with each other by exchanging messages [3]. The parallel version of K-Means is also uses R*- tree to organize data in the central site and message passing principle.

Many existing studies try to improve the efficiency of Parallel DBSCAN algorithm. For example, TI-DBSCAN uses the triangle inequality property to quickly reduce the neighborhood search space without using spatial indices. Some methods enhance DBSCAN by first using CLARANS to partition the dataset for reducing the search space of each partition instead of scanning the whole dataset. GRIDBSCAN constructs a grid that allocates the data points into similar partitions and then DBSCAN processes each partition separately. These algorithms improve the efficiency of DBSCAN, but are still not efficient enough to process massive data. Therefore, a method is proposed as a distributed mining algorithm for DBSCAN to address the scalability problem. The proposed algorithm DBSCAN-MR, which stands for distributed DBSCAN with Map/Reduce, is designed on the Hadoop platform, which uses Google's Map/Reduce-style. The Concept of Distributed Clustering is reverse than Parallel

Clustering. Distributed Clustering techniques cannot carry out a preprocessing step on the server site as the data is not centrally available [2].

II. Distributed Clustering Algorithms

A. *K-Means Based Distributed Clustering Algorithms*

K-Means is a distributed clustering algorithm based on partition. Later a parallel version of K-Means is developed in multiprocessors of distributed memory. [5] Another parallel version of K-Means, which transfers the clustering center. The amount of data in clustering center is usually smaller than the amount of data being divided, the traffic is reduced. [6] One more parallel version of K-Means which only transfer statistic data and has a higher efficiency. [3] Put forward a distributed clustering algorithm

K-DMeans which is based on K-Means. The main site divides the data set into k subsets randomly and stores these k subsets in k subsites. Each sub site calculates its central point and informs other k-1 sub site of its central point. Each sub site calculates the distance from each data point of its local point set to each central point, and cluster data by the idea that each data point belongs to a cluster whose central point is the closest to this data point among all central points. The data which does not belong to this sub site is transferred to another sub site which is in the same cluster with the data. This process iterates until discriminate function value. Because of the characteristics of K-Means, this algorithm is simple, efficient and easy to implement. Furthermore, this algorithm efficiently solves the scalability of K-Means algorithm as *K-DMeans* divides data set to *k* Subsites. But each iteration causes a high computation and communication cost. To solve the disadvantages of K-DMeans, [3] put forward an improved method-*DK-Means* algorithm. Each sub site only needs to send the central point and the number of data points in its cluster to the main site in the process of local clustering, so the communication and computation cost is efficiently reduced. Compared with K-

DMeans, *DK-Means* algorithm reduces the step that the central point of each cluster are transmitted among sub sites, thus reduce the communication cost and increase the efficiency. This algorithm has a good clustering effect for a data set which has a pre-defined number of divisions, but for a data set whose divisions are not pre-defined, different k value will lead to different clustering results. For the single point failure problem of *K-DMeans* and other distributed clustering algorithms, [3] put forward a distributed clustering algorithm based on P2P network-*K-DMeansVM*. It is built on P2P networks and implements clustering without central nodes, thus achieves the same clustering quality as that of K-DMeans and scalability improves.

B. *Density Based Distributed Clustering Algorithms*

Based on datasets clustering classify in to two ways: *Hard clustering*-In which each data object can exist only in one cluster. So, the clusters in hard clustering are separated. Second is *soft clustering:* In which every data object belongs to each and every cluster. Distributed clustering follows two of the following architectures. In *homogeneous* architecture, each local site has the same dataset attributes. In *heterogeneous* architecture a common key attribute make link between clusters of local sites.

DBSCAN is one the most efficient density based clustering algorithm. The parallel version of DBSCAN is given by [7]. Januzaj ET. al. [1] proposed distributed version of DBSCAN algorithm as Density Based Distributed Clustering (DBDC) algorithm. The basic idea of DBDC Algorithm is: firstly, local clustering is carried on in each sub site, secondly, each sub site sends its localized clustering results to the main site which can get global clustering model by carrying on global clustering; finally, the main site will send the global clustering model to each sub site which will update its clustering results based on the global clustering model. Figure (1) gives the three different layers of DBDC algorithm. [2] The main advantage of DBDC algorithm is high clustering speed and suitable for any shape. DBDC uses DBSCAN and K-Means algorithms. Both algorithms are very sensitive to the input parameters. Hence it results in to the disadvantage of DBDC algorithm. Another disadvantage of DBDC algorithm is if representatives send by local sites cannot reflect characteristics of the data set, and then clustering quality will be decreased.

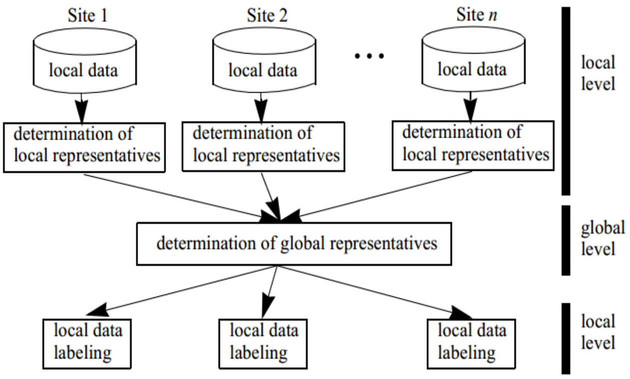
Site 1
Site 2
Site *n*
local data
local data
local data
determination of local representatives
determination of local representatives
determination of local representatives
local level
determination of global representatives
global level
local data labeling
local data labeling
local data labeling
local level

Fig.1: Distributed Clustering Levels

To overcome the limitations of DBDC algorithm [3] gave S-DBDC algorithm. In this algorithm, local noise points are Processed by a proper threshold defined by users in order to control the number of representative points sent from each sub site to the main site. The higher the threshold is set to be better the clustering quality. [3] But the problem with SDBDC algorithm is difficulty for users to find a proper threshold. To eliminate this problem Ni. [8] Brought distributed clustering algorithm based on local density-LDBDC. The LDBDC algorithm is works on the principle of DBDC algorithm, and it focuses on the local density clustering idea. Theoretical analysis and experiment results show that LDBDC algorithm is better than DBDC algorithm in both the clustering quality and the clustering efficiency. [3]

C. *Distributed Algorithms Based On Fuzzy Methods*

Rahimi ET. al. [9] brought the parallel implementation of fuzzy c- means algorithm called as PFCM. In order to overcome the problems of PFCM, [10] gave the distributed version of PFCM, known as PFCM –c* algorithm. This algorithm automatically calculates the number of clusters. Intuitionistic Fuzzy Sets (IFS) are generalized fuzzy sets that are useful in coping with the hesitancy originating from imperfect and imprecise information [11]. Intuitionistic Fuzzy based Distributed Fuzzy Clustering (IFDFC) algorithm was proposed by Visalakshi in [12] works on homogeneous datasets.

D. *Distributed Clustering Algorithms Based On Privacy Protection*

To solve a problem found in some distributed clustering algorithm, such as DBDC, that information about a site may be exposed when transmitting real data of this site to other sites in the process of clustering. This security problem leads us to requirement of some Secured Distributed clustering algorithms. These algorithms are called as Privacy Preserving Distributed clustering algorithms. Privacy Preserving Distributed clustering algorithms are for the distributed database with either horizontal division or vertical division. PPDK-Means (Privacy Preserving clustering with Distributed K-Means) is for the distributed database with horizontal division. [3] PPDK-Means works on a semi-trusted third party in the process of clustering, follows secure multi-party technology to protect real data of this site from being sent to other sites, thus achieves the purpose of privacy protection. It is assumed that any site except the main site is a semi-trusted third party. [3] The semi-trusted third party can create random vectors to disrupt local clustering information, which ensures that the local clustering information is not exposed and thus achieves the purpose of the privacy protection.[11] In DBDC and other Distributed Clustering Algorithms data are at different sites, during the clustering process exchange of information between sites is also takes place. There may be some Critical information there. So it is required to keep this information secret all the time. These leads to the requirement of data security in DBDC Distributed Clustering Algorithms. The process of PPDK-Means algorithm is as follows: [3]

(1) The main site generates k initial clustering centers randomly and broadcasts them to the sub sites
(2) The site as a semi-trusted third party generates two random vectors and sends a disrupted value to corresponding sites

(3) Each sub site clusters according to the received initial clustering centers and encrypts the clustering information

(4) Each sub site sends the encrypted local clustering information to the main site to calculate the global clustering center.

(5) The main site broadcasts the global clustering center to each sub site afterward each sub site restarts clustering.

III. Comparative Analysis

In the TABLE-I the analysis of different Distributed clustering algorithm is given. TABLE-I shows the details such that Algorithm name, Author(s) and advantages/disadvantages. So reader can easily analyze Distributed Clustering Algorithms.

Sr. No	*Algorithm*	*Author(s)*	*Advantages*	*Disadvantages*
1	Density Based Distributed Clustering (DBDC) Algorithm	E.Januzaj, Kriegel, Pfeifle	High quality clusters with scalable transmission cost	Difficult for users to find proper threshold
2	Parallel Fuzzy c – Means Algorithm	Rahimi, Zargham, Thakre, Chhillar	Robust to outliers and noise.	No of clusters to be defined at the beginning
3	PFCM –c* Algorithm	Coletta, Vendramin, Hruschka Pedrycz	Does not require number of clusters to be predefined	Data comes from the same population
4	Distributed Density – Based Clustering (DDC) Algorithm	Khac, Aouad, Kechadi	Suitable for arbitrary shaped clusters and deals well with noise and outliers	Communication cost is high
5	Privacy Preserving Distributed k-clustering Algorithm	Jagannathan Pillaipakkam natt, Wright	efficient simple and communication	no of clusters to be defined at the beginning
6	Distributed Combining Algorithm (DCA)	More, Hall	Datasets are randomly divided and given to local sites.	Restricted to hard clusters at global site
7	Parallel DBSCAN (PDBSCAN) Algorithm	Xu, Jager, Kriegel	High Scalability, Suitable for arbitrary shaped clusters	Communication cost is high
8	Kernel Density Estimation Clustering (KDEC) Algorithm	Klusch, Lodi, Moro	Maintains privacy, as it transmits kernel based density Estimates.	Do not support heterogeneous datasets

Table. 1: the analysis of different Distributed clustering algorithm

IV. Conclusion

In this paper we give the comparative study of various distributed clustering algorithms. We cannot apply the Centralized clustering algorithm when we have large and heterogeneous database. Distributed clustering algorithms satisfy this need. One of the major problems of most of distributed clustering algorithms is their sensitivity to the input parameters. Our future work will be in direction of finding efficient way to deal with input parameters problem of distributed clustering algorithms.

References

[1] E. Januzaj, H-P. Kriegel, M. Pfeifle, Towards Effective and Efficient Distributed Clustering, Institute for Computer Science University of Munich Germany, 2003.

[2] E.Januzaj, H.-P.Kriegel, M.Pfeifle. DBDC: Density-Based Distributed Clustering[C]//Proceedings of the 9th International Conference on Extending Database Technology, 2004: 88-105. K. Elissa, “Title of paper if known,” unpublished.

[3] Mo Hai,Shuyun Zhang, Lei Zhu, Yue Wang. A Survey of Distributed Clustering Algorithms, 2012 International Conference on Industrial Control and Electronics Engineering IEEE 2012

[4] E.Januzaj, H.-P.Kriegel, M.Pfeifle. Scalable Density-Based Distributed Clustering[C]//Proceedings of PKDD 2004:231-244.

[5] M.N.Joshi. Parallel K-Means Algorithm on Distributed Memory
Multiprocessors[J].Computer, 2003, 9:3-15.

[6] G.Forman, B.Zhang. Distributed data clustering can be efficient
and exact[J]. SIGKDD Explorations, 2000, 2(2):34-38

[7] X. Xu, J. Jgerand, H.P. Kriegel. A fast parallel clustering
algorithm for large spatial databases[J]. Data Mining and Knowledge Discovery, 1999, 3(3): 263-290.

[8] W.Ni,G.Chen,Y.J.Wu,etc.Local Density Based Distributed Clustering Algorithm. Journal of Software, 2008, 19(9):2339-2348.

[9] S. Rahimi, M. Zargham, “A. Thakre, and D. Chhillar, A parallel fuzzy c-mean algorithm for image segmentation”, Proceedings of the IEEE Annual. Meeting of the Fuzzy Information Process. Society, Vol. 1, 2004, 234–237.

[10] L.F.S. Coletta, , L. Vendramin, E.R. Hruschka, R.J.G. B. Campello, and W. Pedrycz, “Collaborative Fuzzy Clustering Algorithms: Some Refinements and Design Guidelines”, IEEE Transactions On Fuzzy Systems, 20(3), 2012,444-462.

[11] Deepika Singh, Anjana Gosain, “A Comparative Analysis of Distributed Clustering Algorithms: A Survey” IEEE 2013

[12] N.K.Visalakshi, K.Thangavel, and P.Alagambigai, , “Distributed Clustering for Data Sources with Diverse Schema”, Third 2008 International Conference on Convergence and Hybrid Information Technology, Busan 2008,1056-1061