ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 11, November 2017

Distributed Clustering Using Density Concepts and K-Means

[1] Babita Kumari, [2] Chetan Kumar
[1][2] All India Institutes of Medical Sciences

***Abstract*— Distributed Clustering is in itself a non-trivial challenge and it has further constraints of limiting the communication overhead and the number of processors or deciding the number of parameters required for clustering. There have been several attempts to perform K-means in a distributed environment and some density-based clustering approaches in the distributed environment. Every approach has its own advantage and drawbacks. This paper proposes how density-based approach and a K-means approach can be combined, such that very less information is exchanged among the processor. A Local Clustering is performed based on density concepts. These summaries are then combined together to obtain the global clustering labels through K-means. The K-means is not performed on any portion of data set, rather the information provided by the processors through local clustering. Thus given very less information exchanged the clustering can be performed. We have compared the results against a centralized algorithm baseline, to show the effectiveness.**

***Index Terms*— Clustering, distributed clustering, k-means, density based clustering, ddc, dbscan**

I. INTRODUCTION

Data analysis forms a very crucial step for discovering useful information from the data in any field. Research in current time for applications like marketing and purchasing assistance, molecular biology, natural language processing etc require knowledge discovery from huge amounts of data. Knowledge Discovery in Databases (KDD) tries to identifying valid, novel, potentially useful, and ultimately understandable patterns in data. Nowadays, with the advent of internet the amount of data available for analysis is very huge. The volume is such that it cannot be stored in one place for complete processing. Moreover, the data does not originate from a single source. Rather the data is available at various locations and needs to be collected if traditional methods of centralized analysis are applied. Traditional KDD applications require full access to the data which is going to be analyzed. All data has to be located at that site where it is scrutinized. But it is not possible in current scenario where people are well connected throughout the globe through internet; companies have global presence with offices spread across the world. Thus, the data is being created and stored in multiple and complex forms at various locations. a centralized approach to analyze such data would require heavy exchange of data among these locations and a central repository. The demand of resources for collection and processing at the centre will be very high. This is a very costly affair both economically and time-wise. Hence, it is not a practical solution. It will be better to have a distributed approach. The requirement to extract knowledge out of distributed data, without a prior unification of the data, created the rather new research area of Distributed Knowledge Discovery in Databases (DKDD). Many new researches are being conducted and published where it is suggested how the data can be analyzed at local level and the information generated be so collected and combined so as to produce results equivalent to a central analysis being performed on all the data at once. In particular, we can define the problem of distributed clustering as clustering the data that is available on different locations locally and combining the results such that the representatives and clusters produced are equivalent to a clustering performed on all data together. The aim of clustering is to partition data into homogenous groups such that the representatives of the groups summarize data well. These representatives depend heavily on the method used for clustering and get too much affected by the distribution of the data in the object space. Like, k-means [1] is a very popular method for clustering but depends on the parameters of Gaussian distribution followed by the data in Euclidean space. Since these distributions and spaces are very different at different locations (where data is available), the distributed clustering becomes a challenging task. Even the clustering methods that are free from the assumptions of the distribution model of the data, like density based clustering DBSCAN [2], require exact information of how many data points are similar to each other for computing density. When the objects are distributed over different locations, the estimates of density can get too much deviated and may lead to wrong results. So, the challenge is to identify which part of space is covered at the local level, and what minimum possible information is required to be exchanged so that the global server can detect the overlap of spaces and

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 11, November 2017

conduct global clustering accordingly. This paper proposes how density based clustering at the local level and fast partitioning method like k-means can be employed at global level be combined to produce good results in distributed clustering applications. The amount of information sent to the global server by the local machines is the cluster representatives and the boundary values of the space being covered at the local dataset. This information is partitioned through k-means at global level and simple mapping provides the overall clustering results. This paper is organized into following sections: Section II briefly outlines the relevant work in field of distributed clustering. Section III discusses the proposal and the results of experiments are provided in Section IV.

II. RELATED WORK

Januzaj et al [3] proposed a Density based distributed clustering algorithm in 2003. This method is based on DBSCAN algorithm. The data is clustered locally and independently. Only aggregated information of data is sent to the central server. There is no communication between different local sites. The aggregated information has a set of pairs containing a representative r and ε- range value ε_r showing the validity area of representative. Global clustering is done by DBSCAN algorithm with two input parameters Eps_{global} and $MinPts_{global}$ selected in such a way so that the local models are processed in the best way possible. Aouad et al [4] in 2007, proposed minimum variance increases criterion based lightweight distributed clustering algorithm .This algorithm improves the quality of clustering as well as has low communication cost. In this algorithm optimal local numbers of clusters are selected locally using approximation technique. These clusters are then merging at global level according to increasing variance criterion. Balcan et al [5], introduced algorithms for distributed clustering based on k-medians and k-means. These algorithms have low communication cost compared to existing methods. They eliminate the problem of finding low cost clustering with finding a small size coreset. Coreset is the weighted set of points which summarize local data. The presented algorithm can construct a global coreset in distributed environment with low communication complexity.

In 2015, Bendechache et al [6] proposed a new clustering approach for very large distributed and heterogeneous spatial datasets. Based on k-means algorithm it constructs global clusters dynamically. Due to aggregation phase the final global clusters are compact and accurate.

In 2016, Ding et al [7] proposed new communication efficient approximation algorithm for k-means in distributed environment DISTDIM-K-MEANS algorithm which achieve constant approximation ratios. In Random projection method each local site computes multiplication on a sub-matrix R_l and its own sub-matrix P_l . R_l is selected randomly without utilizing the properties of each P_l. For less communication in their method the distribution of sub-matrix P_l in each site in random projection method is observed. The server constructs a weighted grid in the whole space$\mathbb{R}^d$. Now any k-means clustering algorithm is applied to the grid.

III. PROPOSED WORK – LDGK

A. *Basic Idea*

A distributed k-means approach is always more efficient in terms of messages exchanged and time. Yet, it suffers from a huge drawback: the number of clusters, k, needs to be set a priori. Since, the distribution of data at local sites is very different from the overall probability distribution parameters, setting same value of k at every local site and the global server seems to be a bad idea. It may results into fragmentation of some naturally occurring clusters which happen to exist at a local site. For example, only two naturally occurring clusters exist at a local site, while five such clusters occur at another local site. But both sites are asked to produce 3 clusters because globally value of k is 3. This will result into bad results. Hence, the first thumb rule of distributed clustering should be deciding value of k according to the local data instead of setting a common global value. A density-based method is good as it is independent of the model assumptions about data. Yet, the density based approaches require user-defined parameters that vary according to characteristics of the data available. If density based clustering is to be used at global level, all these characteristics be well communicated to the global server for appropriate setting of parameters. So, we propose to use the advantages of density based clustering at local level so that the assumptions about distribution models are not required. Later, the global server combined the density concepts of the local clusters into global clusters through k-means. This mapping between local clustering labels and global clustering labels is communicated to the local machines and overall clustering is achieved. The proposed algorithm is called **Local DDC Global k-means (LDGK)** algorithm.

B. *Local Clustering Model*

Let the entire data to be clustered be X a set of n vectors, where any vector $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \ldots, x_{id} \rangle$ represents values of i-th object of d attributes. This dataset occurs randomly distributed over the N processors in system. The number of

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 11, November 2017

data objects available at all processors need not to be of same order and the datasets at different processors do not overlap. The Delta Density Clustering (DDC) [8] style of clustering is performed at all processors. Suppose p-th processor has X^pdataset available for clustering, wherein an object is a vector of values $\mathbf{x}_i = \langle x_{i1}, x_{i2}, \dots, x_{id} \rangle$. For every data point the local density is computed as

$$\rho_i = \sum_j \chi(d_{ij} - d_c) \quad (1)$$

Where j is any point other than i, d_{ij} is the distance between them, d_c is a cut-off distance and χ is an activation function such that it gives value of 1 for negative values and 0 for others. Thus, local density counts the number of points within cut-off distance from a point i. Originally in [8], cut-off distance d_c is a user-defined input and needs to be decided by user. We propose to determine its value through the dataset itself. For every point, the distance to its 4th nearest neighbour is recorded, and then the minimum value of it for all points is set as d_c. Delta distance of a point is computed as the minimum distance between itself and other points of higher density

$$\delta_i = \min_{j:p_j > p_i} (d_{ij}) \quad (2)$$

For the point with highest local density δ is

$$\delta_i = \max_j (d_{ij}) \quad (3)$$

The two concepts of density and delta distance indicate that high values pertain to points which are likely to be at the centers of high density region. Hence, they are combined as gamma,

$$\gamma_i = \rho_i . \delta_i \quad (4)$$

The points with extraordinarily high values of gamma are centres of clusters. Each processor thus constructs different number of clusters, as indicated by values of gamma of the points in its dataset. once the points with very high values of gamma are selected as centres, the cluster labels of all remaining points are decide as cluster label of the nearest neighbor of higher density.

Besides the cluster labels for all the points, a local server also determines the edge points of each cluster produced. This is done by including all those points which have either the minimum or maximum value in any dimension in each cluster. Formally, any point which satisfies the following property is an edge point in a cluster C, $\mathbf{x}_i \in C$ and $x_{ij} = \max_{\forall \mathbf{x}_l \in C} x_{lj}$ or $x_{ij} = \min_{\forall \mathbf{x}_l \in C} x_{lj}$ for any j. Since there are d dimensions, at most $2dk_p$ points are sent to the global server along with the cluster labels (called local labels). Where k_p is the number of clusters formed by the p-th processor.

C. Global Clustering Model

The global server receives the edge points of all the local machines, and clusters them according to k-means. The value of k used is the final number of clusters desired for overall dataset. The number of edge points is very small as compared to the size of entire dataset. The k-means process used is the standard version [1], consisting of following two iterative steps:

i. Assign each point to nearest centroid
ii. Update centroid as the mean value of all points in a cluster

The initial centroids are selected at random and the convergence criterion to stop the iterations is that no centroids change. The cluster labels assigned to the points in this phase are called temporary global labels.

D. Mapping of Models

The temporary global labels are corresponded to the local labels in the last step. The edge points from each local cluster have same local label, but may have different temporary global label. Using simple majority voting scheme a single global label corresponding to a local label is selected. Thus, all local labels now have corresponding global labels, which are now the final clustering output.

E. Communication Complexity

The local phase does not require any information exchange among the local servers. At the end, edge points are communicated to the server. This communication is of order $O(dk)$ for each processor. If there are N processors, the communication during the local phase is $O(Ndk)$. During the global phase no other information exchange takes place other than the mapping of the labels. It is of $O(k)$. Thus, the communication complexity of the proposed method is very low.

IV. RESULTS

Clustering results for various datasets are recorded and compared against a centralized DDC clustering output to observe how close to a centralized output the proposal can produce. The quality of output is measured as purity. Also, we record how much time is saved as compared to a centralized solution. We use the popular iris dataset, synthetic Gaussian mixtures called s1, s2, s3 and s4 which have an increasing cluster overlap as shown in Fig 1.

ISSN (Online) 2394-6849

International Journal of Engineering Research in Electronics and Communication Engineering (IJERECE)
Vol 4, Issue 11, November 2017

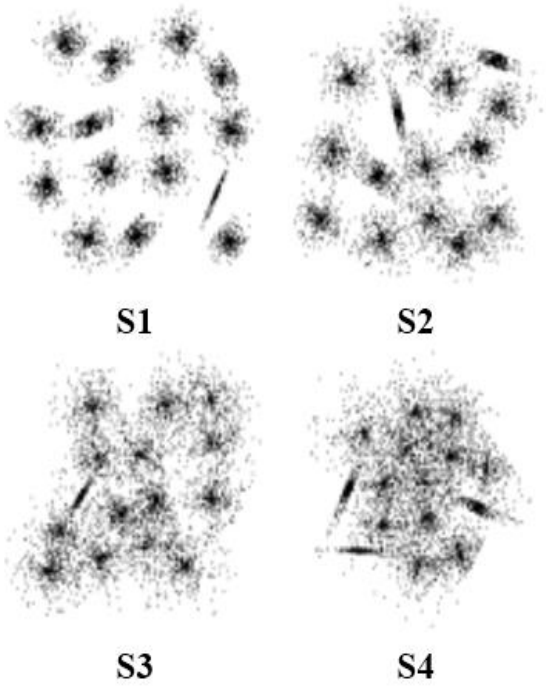

Fig 1. Visualization of the s datasets

The amount of time consumed is shown in Table 1. In every case, average time taken by centralized clustering and proposed LDGK with different number of processors is recorded. We can see as the number of processors increase, the time gets reduced because the amount of work gets reduced at local level. But the decrease is not linear because the time consumed at global server increases slightly. For each dataset, the gain in time achieved through distribution is shown in Fig 2 for better insight. As the number of processors are increased the gain in runtime reaches to a factor of 300.

Table 1 Runtime for clustering various datasets

	Centralized DDC clustering	**LDGK with 5 processors**	**LDGK with 10 processors**	**LDGK with 20 processors**
Iris	0.011161	0.005741	0.002258	0.000777
S1	3.53297	0.146114	0.038774	0.011601
S2	3.71372	0.144265	0.036888	0.011462
S3	3.731424	0.142884	0.037687	0.011458
S4	3.757925	0.146426	0.037523	0.011293

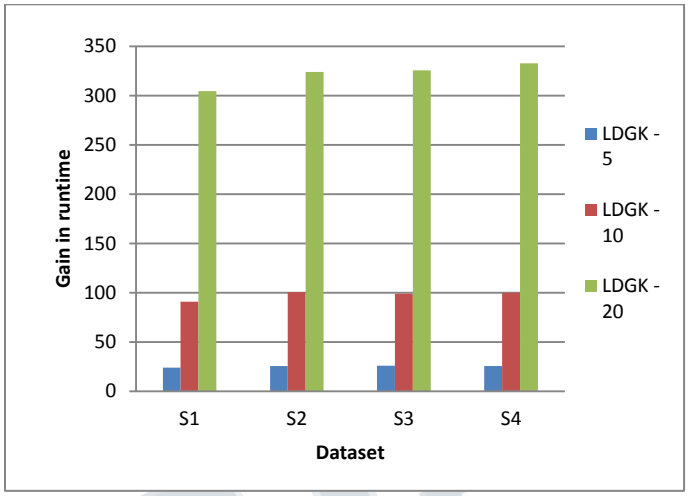

Fig 2 Gain in runtime due to distributed clustering for different datasets at various numbers of processors.

The comparison of purity is given in Fig 3. The fall in purity when clustering using LDGK against a centralized DDC is obvious as the data is now distributed. But purity increases when number of processors are increased, because due to more processors, the number of points available at the global phase increase. This improved quality of clustering. Thus, in a way our proposed method reduces time when number of processors increase and also improves quality of clustering. The trade-off is for the amount of information that will be processed in the global phase and hence communicated between the processors and the server.

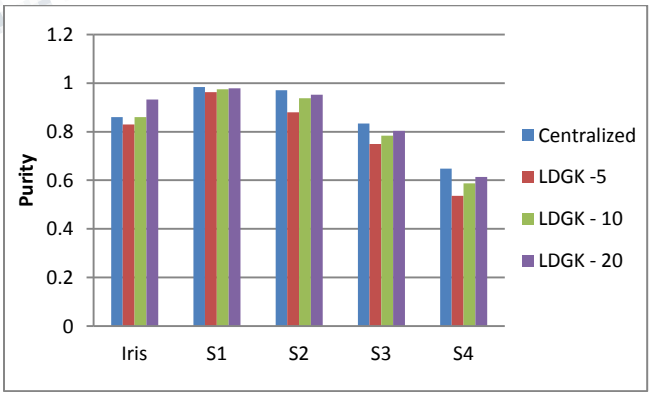

Fig 3 Purity of clusters obtained by the proposed work at different number of processors compared against the centralized clustering of the considered datasets

For small dataset like Iris, purity obtained through distributed method LDGK is more than the centralized method. Among the 's' datasets, the increasing overlap in the cluster structure

makes it difficult to separate the clusters properly. This affects value of purity. For S1, the clusters are well separated and hence purity more than 95% is achieved by all methods. It slightly decreases for S2. But in S4 the purity is below 70%.

V. CONCLUSION

Distributed clustering methods need to be developed as in current situations the data arises at different sources and is too huge to be collected at a single repository. It is better to cluster portions of data wherever they are residing and then combine the results such that a coherent clustering result is produced. We have proposed here a Local DDC Global K-means (LDGK) clustering method. We first suggest how to conduct a parameter free DDC clustering at local machines. LDGK collects the local cluster labels and cluster edge points at the global server. Thereafter a k-means style clustering and majority mapping is used to decide which local labels correspond to global labels. Thus, global clustering output is produced. This proposal requires very less amount of information exchange between processors and global server; no communication is required among the processors themselves. Distributed version saves much time and does not much affect the purity of output as compared against a centralized DDC clustering.

REFERENCES

[1] E.W.Forgy, "Cluster analysis of multivariate data: efficiency v/s interpretability of classifications", Biometrics, Vol. 21, pp. 768–769, 1965.

[2] M. Ester, H.P. Kriegel. J. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", Proceedings of the 2nd ACM SIGKDD Conference, pp. 226-231, 1996.

[3] E.Januzaj, H. Kriegel and M Pfeifle, "Towards Effective and Efficient Distributed Clustering", Workshop on Clustering Large Data Sets (ICDM2003), Melbourne, FL, 2003.

[4] L. M. Aouad, N. Le-Khac, and T. M. Kechadi, "Lightweight Clustering Technique for Distributed Data Mining Applications", Proceedings of the 7th industrial conference on Advances in data mining: theoretical aspects and applications, pp 120-134, Leipzig, Germany, July 14 - 18, 2007.

[5] M.F. Balcan, S. Ehrlich and Y. Liang, "Distributed k-means and k-median clustering on general topologies", Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, pp 1995-2003, Nevada December 2013.

[6] M. Bendechache, T. Kechadi and C.C. Chen, "Distributed Clustering Algorithm for Spatial Data Mining", International conference on Integrated Geo-spatial Information Technology and its Application to Resource and Environmental Management towards GEOSS (IGIT 2015), Hungary, 16-17 January 2015.

[7] H. Ding, Y. Liu, L. Huang and J. Li, "K-Means Clustering with Distributed Dimensions", JMLR: W&CP volume 48, Proceedings of the 33 rd International Conference on Machine Learning , New York, USA, 2016.

[8] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks", Science, Vol. 344, pp. 1492–1496, 2014.