



An Extended Linked Clustering Algorithm for Spatial Data Sets

By K. Lakshmaiah, Dr. S Murali Krishna & Dr. B Eswara Reddy

Sir Vishveshwaraiah Institute of Science & Technology

Abstract- Spatial data mining techniques and for the most part conveyed clustering are broadly utilized as a part of the most recent decade since they manage huge and differing datasets which can't be assembled midway. Current disseminated clustering approaches are typically producing universal models by amassing neighborhood outcomes that are acquired on every region. While this approach mines the data collections on their areas the accumulation stage is more perplexing, which may deliver inaccurate, and equivocal all universal clusters and in this manner mistaken learning. In this paper we propose an Extended Linked clustering approach for each huge spatial data collections that are assorted and appropriated. The approach in view of K-means algorithm yet it produces the quantity of all universal clusters progressively. In addition this approach utilizes an explained collection stage. The conglomerations stage is outlined in such way that the general procedure is proficient in time and memory assignment. Preliminary outcomes demonstrate that the proposed approach delivers excellent outcomes and scales up well. We likewise contrasted it with two prominent clustering algorithms and demonstrate that this approach is substantially more proficient.

Keywords: *spatial data, extended linked clustering, distributed data mining, data analysis, k-means, aggregation.*

GJCST-C Classification: *I.5.3*



Strictly as per the compliance and regulations of:



An Extended Linked Clustering Algorithm for Spatial Data Sets

K. Lakshmaiah^α, Dr. S Murali Krishna^σ & Dr. B Eswara Reddy^ρ

Abstract- Spatial data mining techniques and for the most part conveyed clustering are broadly utilized as a part of the most recent decade since they manage huge and differing datasets which can't be assembled midway. Current disseminated clustering approaches are typically producing universal models by amassing neighborhood outcomes that are acquired on every region. While this approach mines the data collections on their areas the accumulation stage is more perplexing, which may deliver inaccurate, and equivocal all universal clusters and in this manner mistaken learning. In this paper we propose an Extended Linked clustering approach for each huge spatial data collections that are assorted and appropriated. The approach in view of K-means algorithm yet it produces the quantity of all universal clusters progressively. In addition this approach utilizes an explained collection stage. The conglomerations stage is outlined in such way that the general procedure is proficient in time and memory assignment. Preliminary outcomes demonstrate that the proposed approach delivers excellent outcomes and scales up well. We likewise contrasted it with two prominent clustering algorithms and demonstrate that this approach is substantially more proficient.

Keywords: *spatial data, extended linked clustering, distributed data mining, data analysis, k-means, aggregation.*

I. INTRODUCTION

Over a wide assortment of fields, datasets are being gathered and amassed at a sensational pace and enormous measures of data that are being assembled are put away in various destinations. In this specific situation, data mining (DM) strategies have turned out to be vital for removing valuable learning from the quickly developing substantial and multi-dimensional datasets [1]. Keeping in mind the end goal to adapt to vast volumes of data, analysts have created parallel forms of the consecutive DM algorithms [2]. These parallel renditions may help to speedup serious calculations, yet they present critical correspondence overhead, which make them wasteful. To decrease the correspondence overheads circulated data mining (DDM) approaches that comprise of two principle steps are proposed. As the data is normally circulated the main stage comprises of executing the

*Author α: B. Tech, M. Tech, [Ph.D]., MISTE., MIAENG., Assoc., Professor Dept of Computer Science and Engineering Sir Vishveshwaraiah Institute of Science & Technology MADANAPALLE-517325, Chittoor Dist, Andhra Pradesh.
e-mail: klakshmaiah78@gmail.com*

Author σ: Professor in CSE Dept, SV College Of Engineering, TIRUPATI. Chittoor Dist, Andhra Pradesh. India.

Author ρ: Professor and Principal, JNTUA College of Engineering, Kalikiri, Chittoor Dist, Andhra Pradesh. India.

mining procedure on neighborhood datasets on every node to make nearby outcomes. These neighborhood results will be collected to fabricate all inclusive ones. Along these lines the effectiveness of any DDM calculation depends nearly on the productivity of its collection stage. In this unique situation, appropriated data mining (DDM) systems with proficient total stage have turned out to be fundamental for investigating these expansive and multi-dimensional datasets. In addition, DDM is more proper for expansive scale disseminated stages, for example, Clusters and Grids [3], where datasets are regularly geologically circulated and possessed by various associations. Many DDM techniques, for example, disseminated affiliation governs and circulated characterization [4], [5], [6], [7], [8], [9] have been proposed and created over the most recent couple of years. Be that as it may, just a couple of research concerns disseminated clustering for dissecting vast, diversified and conveyed datasets. Ongoing investigates [10], [11], [12], [13] have proposed conveyed clustering approaches in view of a similar 2-step process: perform halfway examination on nearby data at singular destinations and after that send them to a local region to create all universal models by accumulating the neighborhood comes about. In this paper, we propose a conveyed clustering approach in view of a similar 2-step process, be that as it may, it diminishes fundamentally the measure of data traded amid the total stage, produces consequently the right number of groups, and furthermore it can utilize any clustering algorithm to play out the investigation on nearby datasets. A contextual analysis of a proficient conglomeration stage has been produced on unique datasets and turned out to be extremely effective; the data traded is lessened by over 98% of the first datasets [15].

Whatever remains of this paper is sorted out as takes after: In the following segment we will give a diagram of dispersed data mining and examine the constraints of customary strategies. At that point we will introduce and talk about our approach in Section 3. Area 4 introduces the usage of the approach and we talk about exploratory outcomes in Section 5. At last, we finish up in Section.

II. SPATIAL DISTRIBUTED DATA MINING

Existing DDM procedures comprise of two principle stages: 1) performing halfway investigation on

nearby data at singular destinations and 2) producing all universal models by amassing the neighborhood comes about. These two stages are not autonomous since credulous ways to deal with neighborhood investigation may deliver erroneous and questionable all inclusive data models. So as to exploit mined data at various areas, DDM ought to have a perspective of the learning that encourages their reconciliation as well as limits the impact of the nearby outcomes on the general models. Quickly, a productive administration of appropriated learning is one of the key variables influencing the yields of these procedures.

Additionally, the data that will be gathered in various areas utilizing diverse instruments may have distinctive arrangements, highlights, and quality. Conventional incorporated data mining procedures don't consider every one of the issues of data driven applications, for example, adaptability in both reaction time and exactness of arrangements, appropriation and heterogeneity [8], [16].

Some DDM approaches depend on outfit realizing, which utilizes different procedures to total the outcomes [11], among the most referred to in the writing: greater part voting, weighted voting, and stacking [17], [18]. A few methodologies are appropriate to be performed on disseminated stages. For example, the incremental calculations for finding spatio-transient examples by breaking down the hunt space into a progressive structure, tending to its application to multi-granular spatial data can be effectively streamlined on various leveled disseminated framework topology. From the writing, two classifications of methods are utilized: parallel procedures that frequently require devoted machines and instruments for correspondence between parallel procedures which are exceptionally costly, and systems in light of conglomeration, which continue with an absolutely conveyed, either on the data construct models or in light of the execution stages [7], [12]. Nonetheless, the measure of data keeps on expanding as of late, in that capacity, the larger part of existing data mining strategies are not performing admirably as they experiences the versatility issue. This turns into an exceptionally basic issue as of late. Numerous arrangements have been proposed up until this point. They are for the most part in view of little changes to fit a specific data close by.

Clustering is one of the major strategies in data mining. It Clusters data objects in view of data found in the data that portrays the articles and their connections. The objective is to streamline closeness measure inside a bunch and the dissimilarities between groups with a specific end goal to distinguish fascinating structures/designs/models in the data [12]. The two principle classes of bunching are parceling and various leveled. Diverse expounded scientific classifications of existing grouping calculations are given in the writing and numerous appropriated bunching variants in light of

these calculations have been proposed in [12], [20]–[25], and so forth. Parallel bunching calculations are grouped into two sub-classifications. The principal comprises of techniques requiring various rounds of message passing. They require a lot of synchronization. The second sub-class comprises of techniques that manufacture nearby bunching models and send them to a focal site to construct all inclusive models[15].In [20] and [24], message-passing versions of the widely used k-means algorithm were proposed. In [21] and [25], the authors dealt with the parallelization of the DBSCAN density based clustering algorithm. In [22] a parallel message passing version of the BIRCH algorithm was presented. A parallel version of a hierarchical clustering algorithm, called MPC for Message Passing Clustering, which is especially dedicated to Microarray data, was introduced in [23]. Most of the parallel approaches need either multiple synchronization constraints between processes or a universal view of the dataset, or both [12].

Both dividing and various leveled classes have a few shortcomings. For the parceling class, the k-means algorithm requires the quantity of clusters to be settled ahead of time, while in the lion's share of cases K isn't known, moreover various leveled clustering algorithms have beaten this restriction, however they should characterize the halting conditions for clustering deterioration, which are not direct. limitation, but they must define the stopping conditions for clustering decomposition, which are not straightforward.

III. EXTENDED LINKED SPATIAL DISTRIBUTED CLUSTERING

The proposed circulated approach takes after the regular two-advance system; 1) it initially creates neighborhood clusters on each sub-dataset that is allotted to a given preparing node, 2) these nearby groups are accumulated to frame all universal ones. This approach is produced for clustering spatial datasets. The nearby clustering algorithm can be any clustering algorithm. For purpose of clearness it is been K-Means executed with guaranteed (K_i) which can be diverse for every node (see Figure 1). K_i ought to be sufficiently huge to recognize all clusters in the nearby data sets.

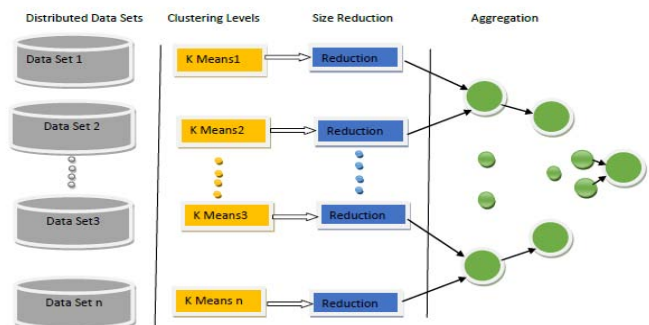


Fig. 1: The Frame work of the Proposed Approach

In the wake of producing nearby outcomes, every node contrasts its neighborhood Clusters and its neighbors' groups. A portion of the nodes, called pioneer, will be chosen to combine neighborhood Clusters to frame bigger groups utilizing the overlay method. These pioneers are chosen by a few conditions, for example, their ability, their handling fake, and so on. The way toward blending groups will proceed until the point that we achieve the root node. The root node will contain the widespread Clusters (models).

Amid the second stage, imparting the neighborhood groups to the pioneers may produce gigantic overhead. Thusly, the goal is to limit the data correspondence and computational time, while getting precise general outcomes. Actually our approach limits the overheads because of the data trade. Thusly as opposed to trading the entire data (entire Clusters) between nodes (neighborhood nodes and pioneers), we initially continue by lessening the data that speak to a group. The span of this new data group is significantly littler than the underlying one. This procedure is done on every nearby node.

There are the number of data diminishment methods proposed in the literature. A significant number of them are centering just in dataset measure i.e., they endeavor to decrease the capacity of the data without focusing on the learning behind this data. In [26], a proficient decrease method has been proposed; it depends on density based clustering algorithm. Each cluster comprises of its agents. Notwithstanding, choosing agents is as yet a test regarding quality and size. We can pick, for instance, medoids points, core points, or even specific core points [10] as representatives [15].

We revolve around the outline and the density of the clustering. The condition of a gathering is addressed by its farthest point centers (called frame) (see Fig 2). Various computations for removing the breaking points from a group can be found in the literature work [27], [28], [29], [30], [31]. We used the figuring proposed in [32] which relies upon Triangulation to make as far as possible. It is a successful figuring for creating non-angled points of confinement. The computation can definitely portray the condition of a broad assortment of dissimilar point flows and densities with a sensible disperse nature of $O(n \log n)$.

The limits of the Clusters speak to the new dataset, and they are substantially littler than the first datasets. So the limits of the Clusters will turn into the nearby outcomes at every node in the system. These neighborhood comes about are sent to the pioneers following a tree topology. The general outcomes will be situated at the foundation of the tree.

IV. IMPLEMENTED APPROACH

a) *Extended Linked Distributed Clustering Algorithm (ELDCA)*

In the main stage, called the parallel stage, the neighborhood grouping is performed utilizing the K-means calculation. Every node (d_i) executes K-means on its nearby dataset to create K_i neighborhood Clusters. When all the nearby groups are resolved, we ascertain their forms. These shapes will be utilized as delegates of their comparing groups. The second period of the method comprises of trading the forms of every node with its neighborhood nodes. This will enable us to check whether there are any covering shapes (Clusters).

In the third step every pioneer endeavors to consolidate covering shapes of its gathering. The pioneers are chosen among nodes of each gathering. In this way, every pioneer produces new shapes (new Clusters). We rehash the second and third steps till we achieve root node. The sub-groups collection is finished after a tree structure and the all inclusive outcomes are situated in the best level of the tree (root node).

As in all Cluster calculations, the normal huge inconstancy in groups shapes and densities is an issue. Be that as it may, as we will appear in the following segment, the calculation utilized for producing the group's form is proficient to distinguish all around isolated clusters with any shapes. In addition ELDCA decides likewise progressively the quantity of the clusters without from the earlier data about the data or an estimation procedure of the quantity of the groups. In the accompanying we will depict the principle highlights and the necessities of the calculation and its condition. The nodes of the distributed computing system are organised following a tree topology.

- A. Each node is dispensed a dataset speaking to a part of the scene or of the general dataset.
- B. Each leaf node (n_i) executes the K-means algorithm with K_i parameter on its neighborhood information.
- C. Neighbouring nodes must share their groups to shape much bigger clusters utilizing the overlay system. The results must reside in the father node (called ancestor).
- D. Repeat C and D until reaching the root node.

In the following we give a pseudo-code of the algorithm:

Algorithm 1: Extended Linked Distributed Clustering Algorithm (ELDCA)

Input : D_i : Dataset Fragment, K_i : Number of sub-clusters for Node $_i$, T : tree degree.

Output: K_u : Universal Clusters (universal results)
level = treeheight;

1. K-means(D_i, K_i);
// Node $_i$ executes K-Means algorithm locally.

```

II. Contour(Ki);
// Node-i executes Contour algorithm to create the limit of
each cluster produced locally.
III. Nodei joins a group G of T elements;
// Nodei joins his neighbourhood.
IV. Compare cluster of Nodei to other Node's clusters
in the same group;
// search for covering between Clusters
V. j= Elect leader Node();
// choose a node which will combine the covering
Clusters
if (i <> j) then
Send(contour i, j);
else
if( level > 0) then
level - - ;
Repeat III, IV, and V until level=1;
else
return (Ku: Nodei's selected clusters);

```

b) Example of execution

We suppose that the system contains five Nodes (N=5), and every Node executes K-Means algorithmic rule with totally different K_i , because it is shown in Fig 2. Node1 executes the K-Means with $K=40$, Node2 with $K=80$, Node3 with $K=120$, Node4 with $k=180$, and Node5 with $K=220$. so every node within the system generates its native clusters. future step consists of merging overlapping clusters at intervals the neighborhood. As we are able to see, though we have a tendency to started with totally different values of K , we have a tendency to generated solely 5 clusters results (See Fig 2).

V. EXPERIMENTAL RESULTS

In this segment, we examine the execution of ELDCA Algorithm and show its viability contrasted with BIRCH and CURE calculations:

BIRCH: We utilized the execution of BIRCH gave by the creators in [33]. It plays out a pre-grouping and after that uses a centroid-based various leveled bunching calculation. Note that the time and space many-sided quality of this approach is quadratic to the quantity of focuses after pre-grouping. We set parameters to the default esteems recommended in [33].

CURE: We utilized the usage of CURE gave by the creators in [34]. The calculation utilizes agent focuses with contracting towards the mean. As portrayed in [34], when two groups are converged in each progression of the calculation, agent focuses for the new blended group are chosen from the ones of the two unique clusters as opposed to every one of the focuses in the consolidated clusters.

ELDCA: Our calculation is portrayed in Section IV. The key point in our approach is to pick K_i greater than the right number of groups. As portrayed toward the finish

of Section IV, when two groups are converged in each progression of the calculation, delegate purposes of the new consolidated bunch are the association of the shapes of the two unique groups instead of all focuses in the new group. This paces up the execution time without unfavorably affecting on the nature of the created groups. Also, our system utilizes the tree topology, store information structures and Agglomerative various leveled grouping. Accordingly, this additionally enhances the many-sided quality of the calculation.

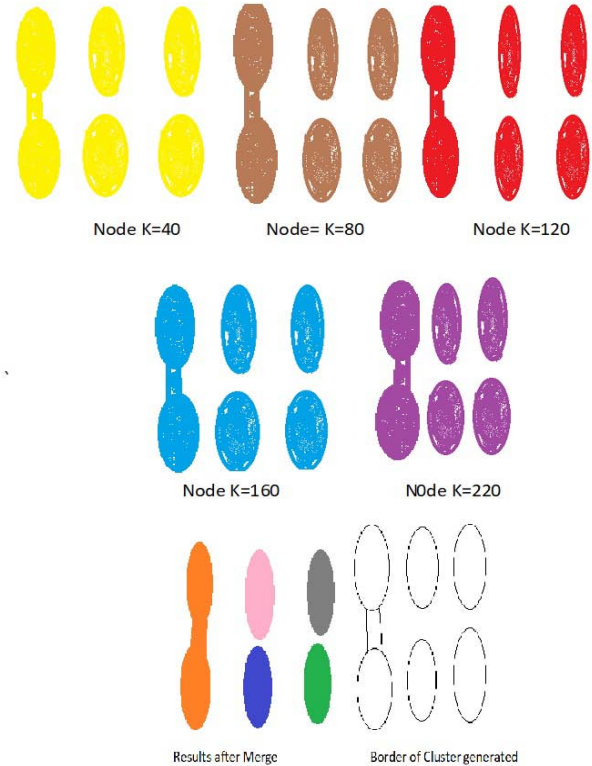


Fig. 2: Extended Linked Distributed Clustering Algorithm (ELDCA)

a) Data sets

We run experiments with different datasets. In this paper we use three types of datasets. These are summarised in Table 1. The number of points and clusters in each dataset is also given in Table 1. We show that ELDCA algorithm not only correctly clusters the datasets, but also its execution time is much quicker than BIRCH and CURE.

b) The Obtained Quality of Clustering

We run the three algorithms on the three datasets to compare them with respect to the quality of clusters generated and their response time. Fig 3, Fig 4 and Fig 5 show the clusters found by the three algorithms for the three datasets (dataset1, dataset2 and dataset3). We use different colours to show the clusters returned by each algorithm.

Fig 3 shows the clusters generated from the dataset1. As expected, since BIRCH uses a centroid-based hierarchical clustering algorithm for clustering the pre-clustered points, it could not find all the clusters correctly. It splits the larger cluster while merging the others. In contrast, the CURE algorithm succeeds to generate the majority of clusters but it still fails to discover all the correct clusters. Our distributed clustering algorithm successfully generates all the clusters with the default parameter settings described in section IV. As it is shown in Fig 3, after merging the local clusters, we generated five final clusters.

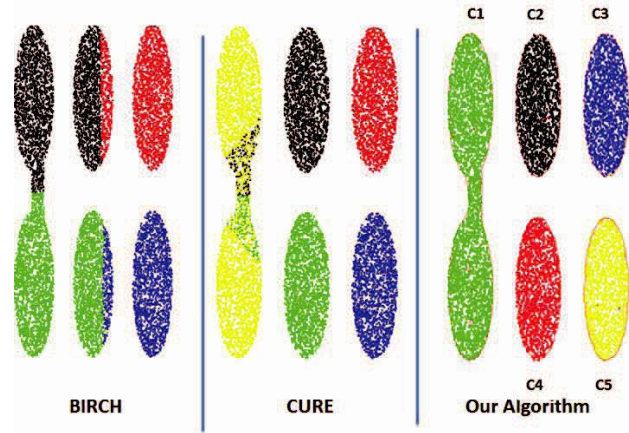


Fig. 3: Clusters generated from dataset 1.

Table 1: Datasets

Data Sets	Numbers of points	Shape of Clusters	Number of Clusters
Data set 1	16000	Big Oval (Egg Shape)	Five
Data set 2	41350	2 Small Circles, 1 Big Circle and 2 Ovals Linked	Four
Data set 3	19080	4 Circles and 2 Circles Linked	Five

Fig 4 shows the outcomes found by the three algorithms for the dataset 2. Once more, BIRCH and CURE neglected to create every one of the clusters, while our algorithm effectively produced the four right clusters.

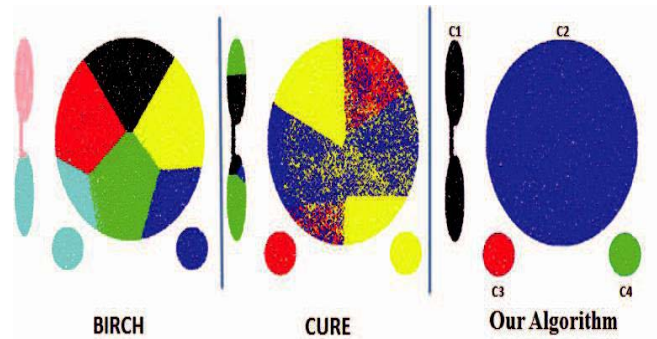


Fig. 4: Clusters generated from dataset 2.

Fig 5 Represents the clusters we found from the dataset 3. As should be obvious BIRCH still neglects to discover every one of the clusters effectively. Interestingly CURE found the 5 clusters, yet not flawlessly. For example, we can see some red focuses in the blue cluster and some blue focuses in the green cluster. Our Algorithm produced the five clusters effectively and impeccably.

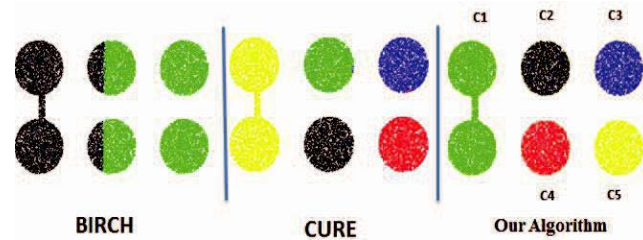


Fig. 5: Clusters generated from dataset 3.

c) Observations

As should be obvious, our method effectively produced the last groups for the three datasets. This is because of the way that:

At the point when two groups are combined, the new bunch is spoken to by the association of the two shapes of the two unique bunches. This paces up the execution times without affecting the nature of groups generated. The number of all inclusive bunches is dynamic.

d) Comparison of ELDCA's Execution Time to BIRCH and CURE

The goal here is to demonstrate the impact of using the combination of parallel and distributed architecture to deal with the limited capacity of a node in the system and tree topology to accelerate the speed of computation.

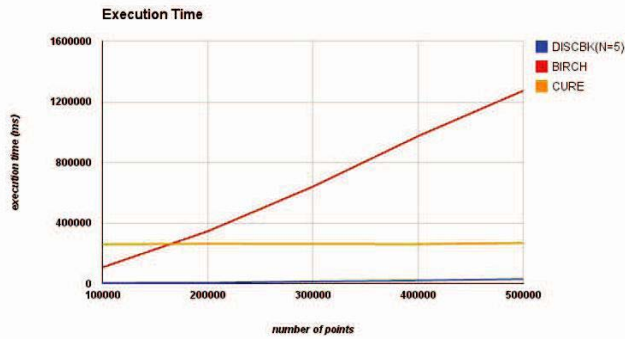


Fig. 6: Comparison to BIRCH and CURE.

Fig. 6 shows the execution of our algorithm contrasted with BIRCH and CURE as the quantity of information directs increments from 100,000 toward 500,000 the quantity of groups and their shapes are not adjusted. In this manner, for our calculation we think about the quantity of nodes in the system: $N=5$. The execution times do exclude the ideal opportunity for showing the clusters since these are the same for the three algorithms.

As can be found in Fig 6, ELDCAs execution time is much lower than CURE's and BIRCH's execution times. Moreover, as the quantity of focuses expands, our execution time is about even, while, the executions time of BIRCH increments quickly with the dataset estimate. This is on the grounds that BIRCH sweeps the whole database and uses every one of the focuses for pre-clustering. At long last as the quantity of focuses expands the CURE's execution time is about even, since CURE utilizes a testing method, where the span of this example remains the same and the main extra cost brought about by CURE is simply the inspecting strategy.

The above outcomes affirm that our disseminated clustering algorithm is extremely proficient contrasted with both BIRCH and CURE either in nature of the clusters created and in the computational time.

e) Scalability

The objective of the adaptability tests is to decide the impacts of the quantity of nodes in the framework on the execution times. The dataset contains 1000,000 focuses. Fig 7 demonstrates the execution time against the quantity of nodes in the framework. Our calculation took just a couple of moments to group 1000,000 focuses in a conveyed framework that contains more than 100 nodes. In this way, the calculation can serenely deal with high-dimensional data in view of its low multifaceted nature.



Fig. 7: Scalability Experiments.

VI. CONCLUSIONS

In this paper, we propose another and imaginative Extended Linked DCA, to manage spatial datasets. This approach misuses the preparing intensity of the appropriated stage by augmenting the parallelism and limiting the interchanges and for the most part the measure of the information that is traded between the hubs in the framework. Nearby models are created by executing a grouping calculation in every hub, and afterward the neighborhood comes about are converged to construct the all inclusive clusters. The nearby models are spoken to with the goal that their sizes are sufficiently little to be traded through the system.

Trial comes about are likewise displayed and talked about. They likewise demonstrate the viability of ELDCAs either on amount of the clustering produced or the execution time contrasting with BIRCH and CURE calculations. Besides, they show that the calculation outflanks existing calculations as well as scales well for extensive databases without giving up the grouping quality. ELDCAs is not quite the same as present dispersed grouping models introduced in the writing, it describes by the dynamic number of clusters created and its proficient information decrease stage.

A more broad assessment is continuous. We will plan to run tries different things with different neighborhood algorithms and investigate the conceivable outcomes of stretching out the strategies to different sorts of expansive and appropriated datasets.

REFERENCES RÉFÉRENCES REFERENCIAS

1. U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "Knowledge discovery and data mining: Towards a unifying framework," in *Proc. KDD-96*, 1996, pp. 82–88.
2. A. A. Freitas and S. H. Lavington, *Mining very large databases with parallel processing*. 1st edition, Springer; 2000 edition, 30 November 2007.

3. I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers Inc. San Francisco, CA, USA, 1999.
4. T. G. Dietterich, "An experimental comparison of three methods for Constructing ensembles of decision trees: Bagging, boosting and randomization," *Machine Learning*, vol. 40, pp. 139–157, 2000.
5. H. Kargupta and P. Chan, *Advances in distributed and Parallel Knowledge Discovery*, USA MIT Press Cambridge, MA, October 2000.
6. R. Agrawal and J. C. Shafer, "Parallel mining of association rules," in *proc. IEEE Transactions on Knowledge and Data Engineering*, vol. 8, pp. 962–969, 1996.
7. L-M. Aouad, N-A. Le-Khac, and M-T. Kechadi, "Performance study of a distributed apriori-like frequent itemsets mining technique," *Knowledge Data Systems*, vol. 23, pp. 55-72, Apr. 2010.
8. L-M. Aouad, N-A. Le-Khac, and M-T. Kechadi, "Grid-based approaches for distributed data mining applications," *algorithms Computational Technology*, vol. 3, pp. 517–534, 10 Dec. 2009.
9. N-A. Le-Khac, L-M. Aouad, and M-T. Kechadi, "Toward a distributed knowledge discovery on grid systems," in *Emergent Web Intelligence: Advanced Semantic Technologies*, London. Springer, April 2010, pp 213-243.
10. E. Januzaj, H-P. Kriegel, and M. Pfeifle, "DBDC: Density-based distributed clustering," in *Advances in Database Technology*, vol. 2992, Greece, March 14-18, 2004, pp. 88-105.
11. N-A. Le-Khac, L-M. Aouad, and M-T. Kechadi, "A new approach for distributed density based clustering on grid platform." In *Data Management. Data, Data Everywhere*, Volume 4587, Springer-Verlag Berlin, Heidelberg, 2007, pp. 247–258.
12. L-M. Aouad, N-A. Le-Khac, and M-T. Kechadi, "Lightweight clustering Technique for distributed data mining applications," in *Advances in Data Mining. Theoretical Aspects and Applications*, Germany. Springer Berlin Heidelberg, 2007, pp. 120–134.
13. L-M. Aouad, N-A. Le-Khac, and M-T. Kechadi, *Advances in Data Mining. Theoretical Aspects and Applications*. Ed Berlin Heidelberg, Germany Springer 14-18 July 2007.
14. J. Han, M. Kamber, J. Pei, *Data Mining Concept and Techniques*, 2nd edition. Morgan Kaufmann, 6 April 2006.
15. J.F. Laloux, N-A. Le-Khac, and M-T. Kechadi, "Efficient distributed approach for density-based clustering," *Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 20th IEEE International Workshops*, pp. 145 – 150, 27-29, June 2011.
16. M. Bertolotto, S. Di Martino, F.Ferrucci, and M-T. Kechadi, "Towards a framework for mining and analysing spatio-temporal datasets," *International Journal of Geographical Data Science – Geovisual Analytics for Spatial Decision Support*, vol. 21, pp. 895-906, January 2007.
17. P. Chan and S. J. Stolfo, "A comparative evaluation of voting and meta-learning on partitioned data," in *In Proceedings of the Twelfth International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 90–98.
18. C. R. Reeves, *Modern heuristic techniques for combinatorial problems*. 1st edition, John Wiley & Sons, Inc. New York, NY, USA, May 1993.
19. L-M. Aouad, N-A. Le-Khac, and M-T. Kechadi, "Performance study of a distributed apriori-like frequent itemsets mining technique," *Knowledge Data Systems*, Springer-Verlag, vol. 23, pp 55-72, 2009.
20. I. S. Dhillon and D. S. Modha, "A data-clustering algorithm on distributed memory multiprocessor," in *large-Scale Parallel Data Mining, Workshop on Large-Scale Parallel KDD Systems, SIGKDD*. Springer-Verlag London, UK, 1999, pp. 245–260.
21. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." In *proc. KDD96*, 1996, pp. 226–231.
22. A. Garg, A. Mangla, V. Bhatnagar, and N. Gupta, "PBirch: A scalable parallel clustering algorithm for incremental data," in *proc. Database Engineering and Applications Symposium. IDEAS' 06. 10th International, Delhi*, 2006, pp. 315-316.
23. H. Geng, and X. Deng, "A new clustering algorithm using message passing and its applications in analyzing microarray data," in *proc. ICMLA '05 Proceedings of the Fourth International Conference on Machine Learning and Applications. IEEE*, 15-17 December 2005, pp. 145–150.
24. I. D. Dhillon and D. S. Modha, "A data-clustering algorithm on distributed memory multiprocessors," in *proc. Large-Scale Parallel Data Mining. Springer Berlin Heidelberg*, 2000, pp. 245–260.
25. X. Xu, J. Jager, and H. P. Kriegel, "A fast parallel clustering algorithm for large spatial databases," in *Data Mining and Knowledge Discovery archive*, vol. 3, September 1999, pp. 263 – 290.
26. N-A. L-Khac, M. Bue, and M. Whelan, "A knowledge based data reduction for very large spatio-temporal datasets," in *proc. International Conference on Advanced Data Mining and Applications (ADMA'2010)*. Springer Verlag LNCS/LNAI, Chongqing, China, November 19-21,
27. J. M. Fadili and M. Melkemi and A. ElMoataz, "Non-convex onion-peeling using a shape hull algorithm," *Pattern Recognition Letters*, vol. 25, pp. 1577 – 1585, 14-15 October 2004.

28. A. R. Chaudhuri and B. B. Chaudhuri and S. K. Parui, "A novel approach to computation of the shape of a dot pattern and extraction of its perceptual border," *Computer vision and Image Understanding*, vol. 68, pp. 257- 275 , 03 December 1997.
29. M. Melkemi and M. Djebali, "Computing the shape of a planar points set," *Pattern Recognition*, vol. 33, pp. 1423–1436, 9 September 2000.
30. H. Edelsbrunner, D. Kirkpatrick, and R. Seidel, "On the shape of a set of points in the plane," *IEEE Transaction on Data Theory*, vol. 29, pp. 551 – 559, July 1983.
31. A. Moreira and M. Y. Santos, "Concave hull: A k-nearest neighbours approach for the computation of the region occupied by a set of points," *in proc. proceedings of the International Conference on Computer Graphics Theory and Applications*, March 2007.
32. M. Duckham, L. Kulik, and M. Worboys, "Efficient generation of simple polygons for characterizing the shape of a set of points in the plane," *Pattern Recognition*, vol. 41, pp. 3224–3236, 15 March 2008.
33. T. Zhang, and R. Ramakrishnan and M. Livny, "Birch: An efficient data clustering method for very large databases," *in proc. SIGMOD '96 Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, vol. 25. ACM New York, NY, USA, 1996, pp. 103–114.
34. S. Guha and R. Rastogi and K. Shim, "Cure: An efficient clustering algorithm for large databases," *Data Systems*, vol. 26, pp. 35– 58, Nov. 2001.

