To evaluate and improve DBSCAN algorithm with Back Propagation in data mining

Er. Paramvir Kaur Dhillon[1], Er. Jagdeep Kaur[2]

[1]CSE (Computer Science and Engineering) SBBSU (Sant Baba Bhag Singh University), Jalandhar, India

[2]CSE (Computer Science and Engineering) SBBSU (Sant Baba Bhag Singh University), Jalandhar, India

Abstract

The clustering is the technique in which similar and dissimilar type of data is clustered in different clusters for better analysis of the input data. The algorithm of DBSCAN is applied in which EPS is calculated which will be the central point and from the central point Euclidean distance is calculated to define similarity and dissimilarity of the input data. In the existing algorithm EPS is calculated dynamically but Euclidian distance statically which reduce accuracy of clustering. In this work, back propagation algorithm is been applied which calculate Euclidian distance dynamically and simulation study is conducted which shows that proposed improvement increase accuracy of clustering and reduce execution time.

Keywords: Clustering, DBSCAN, Back-propagation, Accuracy, Execution time.

1. INTRODUCTION

Data mining is viewed as a result of the natural evolution of information technology. The early development of data collection and database creation mechanisms proved to be important for the later development of effective mechanisms for data storage and retrieval, query and transaction processing [1][2][3]. The database and data management industry evolved in the development of several critical functionalities: data collection and database creation, data management and advanced data analysis (involving data warehousing and data mining). One of the emerging data repository architecture is the data warehouse. It involves multiple heterogeneous data sources organized under a unified schema at a single site to manage decision making. Cluster analysis has been widely used in various applications including market research, pattern recognition, data analysis, and image processing. In business, clustering can help marketers discover interests of their customers based on purchasing patterns and characterize groups of the customers [4][5]. Data mining is also used in biology to derive plant and animal taxonomies. It also categorize genes with similar functionality, and gain insight into structures inherent in populations. Data clustering (or just clustering), is an unsupervised classification method. This method aims at creating groups of objects or clusters, in such a way that objects in the same cluster are very similar and objects in different clusters are quite distinct[6][7]. Cluster analysis is one of the traditional topics in the data mining field. In partitioning methods clusters are formed on the basis of distance between objects. Spherical shaped clusters can be discovered by these methods and encounter trouble in discovering clusters of arbitrary shapes. So for arbitrary shapes new methods are utilized known as density-based methods which are based on the notion of density. In these methods the cluster is continued to develop as long as the density in the area exceeds some threshold [8][9][10].This strategy is based on the notion of density. The fundamental thought is to bear on the growing the given cluster as long as the density in the area exceeds some threshold i.e. for every data point inside a given cluster, the radius of a given cluster needs to contain no less than a minimum number of points. It discovers arbitrary shape clusters. It likewise handles clamor in the data[11][12]. It is one time scan. It requires density parameters additionally[14].

2. LITERATURE REVIEW

Guangchun Luo, et.al, proposed system of cluster analysis occupies a pivotal position in data mining, and the DBSCAN algorithm is a standout amongst the most broadly utilized algorithms for clustering. Nonetheless, when the existing parallel DBSCAN algorithms make data partitions, the original database is normally divided into several disjoint partitions; with the increase in data dimension, the splitting and consolidation of high-dimensional space will consume a lot of time [14]. To solve the problem, this paper proposes a parallel DBSCAN algorithm (S_DBSCAN) based on Spark, which can quickly realize the partition of the original data and the mix of the clustering results. It is divided into the following strides: 1) partitioning the raw data based on a random sample, 2) computing local DBSCAN algorithms in parallel, 3) merging the data partitions based on the centroid.

Dianwei Han, et.al, analyzed that DBSCAN is an outstanding clustering algorithm which is based on density and can identify arbitrary shaped clusters and eliminate noise data. Be that as it may, parallelization of DBSCAN is a testing work on the grounds that based on MPI or OpenMP environments, there exist the issues of lack of fault tolerance and there is no guarantee that workload is balanced. Also, programming with MPI requires data scientists to have an advanced experience to handle communication between nodes which is a big challenge [15]. DBSCAN algorithm has been extremely famous since it can identify arbitrary shaped clusters and additionally handle noisy data.

Nagaraju S, et.al, introduce an efficient approach for clustering analysis to detect embedded and nested adjacent clusters utilizing idea of density based notion of clusters and neighborhood difference. Basically the proposed algorithm is improved version basic DBSCAN algorithm, proposed to address the clustering problem with the utilization global density parameters in basic DBSCAN algorithm and problem of detecting nested adjacent clusters in EnDBSCAN algorithm. The experimental results that suggested that proposed algorithm is more effective in detecting embedded and nested adjacent clusters compared both DBSCAN and EnDBSCAN without adding any additional computational complexity [16].

Jianbing Shen, et.al, proposes a real-time picture superpixel segmentation method with 50fps by utilizing the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm. In order to decrease the computational costs of superpixel algorithms, the method received a quick two-stage framework. In the principal clustering stage, the DBSCAN algorithm with color similarity and geometric confinements is utilized to quickly cluster the pixels, and afterward small clusters are merged into superpixels by their neighborhood through a distance measurement defined by color and spatial features in the second merging stage [17]. A robust and straightforward distance function is defined for getting better superpixels in these two stages.

Ilias K. Savvas, et.al, propose standout amongst the most fascinating and productive techniques, in order to locate and extract information from data storehouses are clustering, and DBSCAN is a successful density based algorithm which clusters data concurring its characteristics [18]. Be that as it may, its fundamental burden is its severe computational complexity which proves the technique exceptionally inadequate to apply on big datasets. Despite the fact that DBSCAN is an exceptionally very much studied technique, a completely operational parallel version of it, has not been accepted yet by mainstream researchers. In this work, a three phase parallel version of DBSCAN is presented. The obtained experimental results are exceptionally promising and demonstrate the correctness, the scalability, and the effectiveness of the technique.

Ahmad M. Bakr, et.al, The proposed algorithm enhances the incremental clustering process by limiting the search space to partitions as opposed to the whole dataset which results in significant improvements in the performance compared to relevant incremental clustering algorithms [19]. Experimental results with datasets of various sizes and dimensions demonstrate that the proposed algorithm speeds up the incremental clustering process by factor up to 3.2 compared to existing incremental algorithms. The algorithm incrementally partitions the dataset to reduce the search space to every partition as opposed to filtering the whole dataset. After that the algorithm incrementally forms and updates dense regions in every partition. Following identifying possible dense regions in every partition, the algorithm utilizes an inter-connectivity measure to merge dense regions to shape the final number of clusters.

3. DBSCAN ALGORITHM

Density based clustering algorithms have a wide applicability in data mining. They apply a local criterion to group objects: clusters are viewed as regions in the data space where the objects are dense, and which are separated by regions of low object density (noise) [12]. Among the density based clustering algorithms DBSCAN is exceptionally well known due both to its low complexity and its capacity to detect clusters of any shape, which is a desired characteristic when one doesn't have any knowledge of the possible clusters' shapes, or when the objects are circulated heterogeneously, for example, along paths of a graph or a road network. In any case, this DBSCAN algorithm needs two numeric input parameters, to drive the process which together characterizes the desired density characteristics of the generated clusters. In particular, minPts is a positive integer determining the minimum number of objects that must exist inside a maximum distance of the data space all together for an object to have a place with a cluster. Since DBSCAN is extremely sensible to the setting of these input parameters they should be picked with incredible accuracy by considering both the scale of the dataset and the closeness of the objects all together not to affect an excessive amount of both the speed of the algorithm and the effectiveness of the outcomes. To settle the right values of these parameters one by and large engages an exploration phase of trials and errors in which the clustering is run several times with distinct values of the parameters. The DBSCAN algorithm can identify clusters in extensive spatial data sets by taking a gander at the local density of database components, utilizing one and only input parameter. Besides, the client gets a proposal on which parameter value that would be reasonable. Along these lines, minimal knowledge of the domain is required. The DBSCAN can likewise figure out what data ought to be classified as noise or outliers. Regardless of this, its working process is quick and scales extremely well with the extent of the database – linearly. By utilizing the density distribution of nodes in the database, DBSCAN can categorize these nodes into separate clusters that characterize the diverse classes. DBSCAN can discover clusters of arbitrary shape. In any case, clusters that lie close to each other have a tendency to have a place with a similar class.

4. RESEARCH METHODOLOGY

In the DBSCAN algorithm the most dense region is calculated from the dataset. The central point is calculated from the most dense region which is the called EPS value of the dataset. To calculate similarity between the data points of the data Euclidian distance is calculated from central point to all other points. The elements which are similar is clustered in one dataset and other are in the second dataset. In the base paper, to improve accuracy of clustering EPS values is calculated in the dynamic manner which leads to the clustering of the points which are remained unclustered. . To achieve more accuracy of clustering technique of back propagation will be applied which calculate Euclidian distance in the dynamic manner and increase accuracy and reduce execution time of improved DBSCAN algorithm.

Algorithm

Input: Dataset for clustering, desired and output patterns
Output: Clustering of input data

```
M← List of objects that may change their centriods
D← Most dense regions in the dataset
For each point p(i) in P do
C← nearest centriod ()
Function nearest centriod ()
  initpopulation P
     evaluate P ;
Network ConstructNetworkLayers()
 InitializeWeights(Network, test cases)
For ( i=0;i=P ;i++)
   SelectInputPattern(Input fault values)
    ForwardPropagate(p)
   BackwardPropagateError(P)
   UpdateWeights(P )
End
Return (P)
C←P;
M← update centriod
End
For each r to M
For each ri in M do
 c <-ri new_centroid
 Co<-ri old_centroid
Apply incDbscanDel to remove ri from co
Apply incDbscanAdd to insert ri to cn
Add updated dense regions to D
end for
For each di in D do
For each dj in D and i – j do
If inter_connectivty (di,dj) > a merge
merge(di,dj)
end if
end for
 end for
```

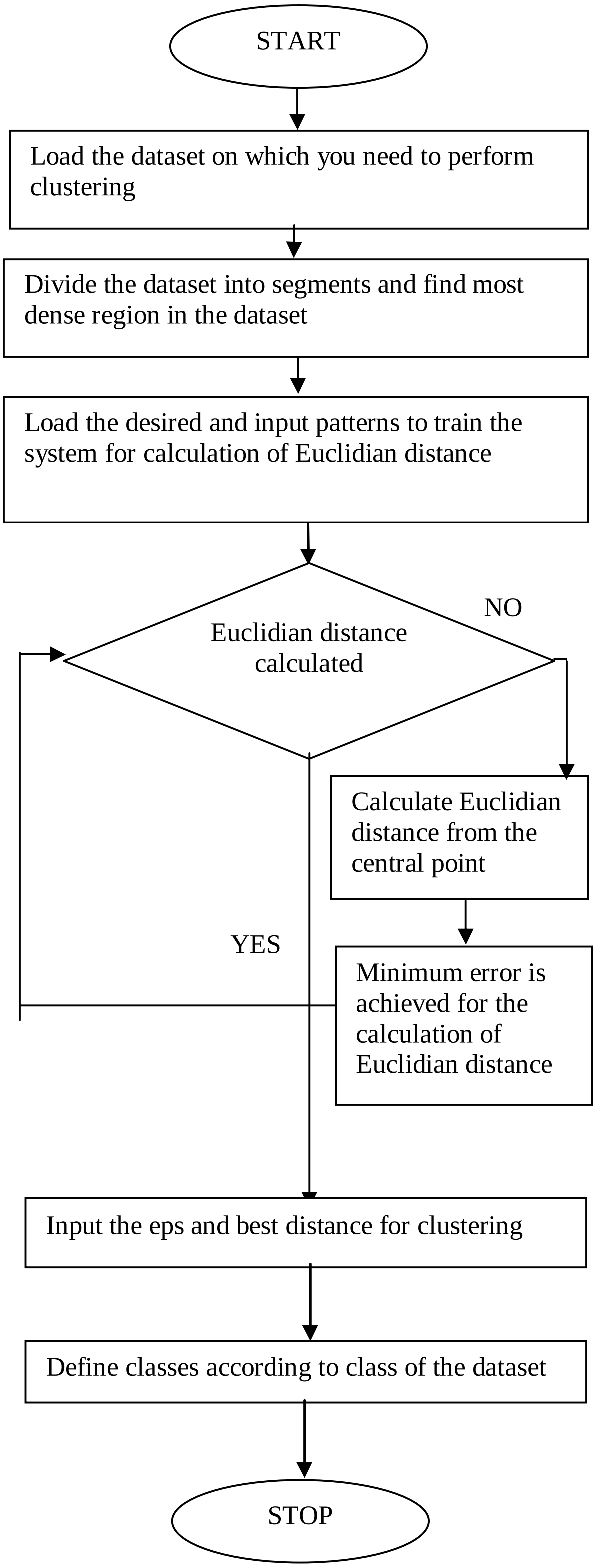
START
Load the dataset on which you need to perform clustering
Divide the dataset into segments and find most dense region in the dataset
Load the desired and input patterns to train the system for calculation of Euclidian distance
Euclidian distance calculated
NO
Calculate Euclidian distance from the central point
Minimum error is achieved for the calculation of Euclidian distance
YES
Input the eps and best distance for clustering
Define classes according to class of the dataset
STOP

Figure 1 Proposed work

As illustrated in figure 1, the flowchart of the proposed improvement which is done in the DBSCAN algorithm to improve accuracy of clustering. In the existing DBSCAN algorithm EPS value is calculated dynamically and Euclidian distance is calculated statically which reduce efficiency of the algorithm. This work is based on to calculate Euclidian distance dynamically due to which technique back propagation algorithm is applied which define the Euclidian distance in the iterative manner and distance at which error is minimum is the final Euclidian distance . When the final Euclidean distance is considered similar and dissimilar type of data is clustered for analysis.

5. RESULTS AND DISCUSSION

The proposed and existing algorithm is implemented in MATLAB to test on the desired dataset.

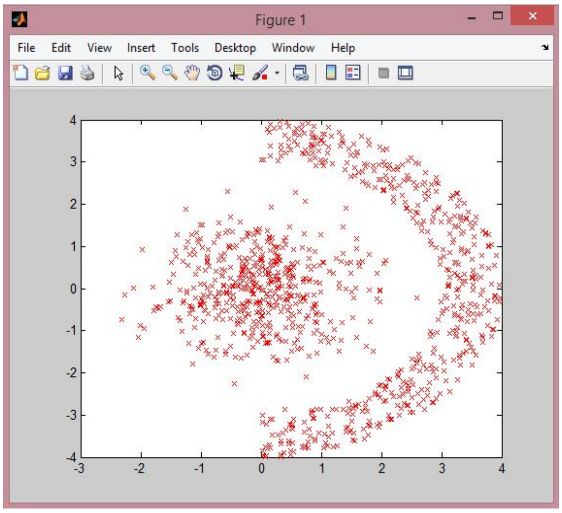

Figure 2 Incremental DBSCAN algorithm

As shown in figure 2, the algorithm of DBSCAN is applied which will cluster the similar and dissimilar type of data from the most dense region in the input dataset.

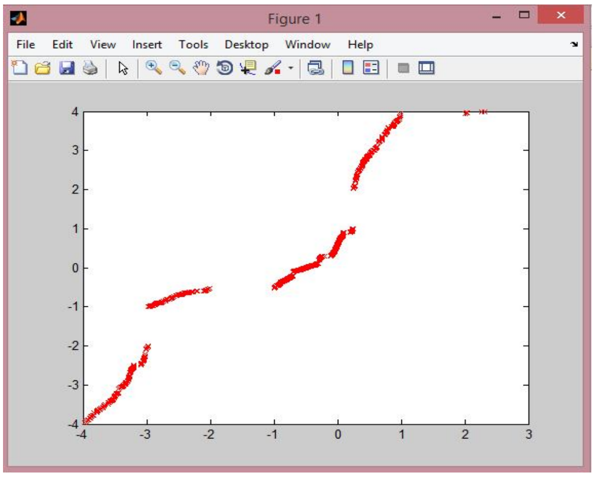

Figure 3 Enhanced DBSCAN Algorithm

As shown in figure 3, the improvement in the existing DBSCAN algorithm is been proposed in which back propagation algorithm is been applied to calculate Euclidean distance in iterative manner this leads to increase accuracy of clustering.

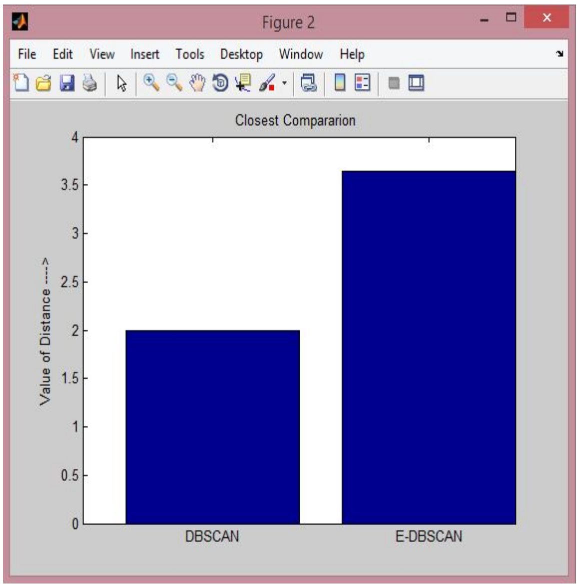

Figure 4 Distance Compression

As shown in figure 4, the distance of the proposed and Incremental DBSCAN algorithm is compared and it is been analyzed that distance of Competent DB-SCAN algorithm is more accuracy than existing DBSCAN Algorithm.

Table 1: Table of comparison

Parameter	Incremental DBSCAN	Enhanced DBSCAN
Accuracy	86 percent	92 percent
Time	5.5 second	4.71 seconds
Distance	2	3.7
EPS	1.33	0.9
Noise Ratio	22 Percent	16 percent
F-Measure	0.25	0.55

As illustrated in table 1, the Performance of Incremental DBSCAN algorithm and Enhanced DBSCAN algorithm is compared in terms of accuracy, time, distance and EPS.

6. CONCLUSION

In this work, it is been concluded that density based clustering is the efficient type of clustering in which clusters are defined on the density of the input data. The DBSCAN is the algorithm in which EPS value is calculated which will be central point and Euclidean distance is calculated from the central point which define similarity and dis-similarity of the data. In the existing work, Euclidian distance is calculated in the static manner which is made dynamic in proposed work using back propagation algorithm. The proposed improvement leads to increase accuracy of the clustering and reduction in execution time.

References

[1]. Anand M. Baswade, Kalpana D. Joshi and Prakash S. Nalwade, "A Comparative Study Of K-Means and Weighted K-Means for Clustering," International Journal of Engineering Research & Technology, Volume 1, Issue 10, December-2012

[2]. Neha Aggarwal, Kirti Aggarwal and Kanika Gupta, "Comparative Analysis of k-means and Enhanced K-means clustering algorithm for data mining," International Journal of Scientific & Engineering Research, Volume 3, Issue 3, August-2012

[3]. Ahamed Shafeeq B M and Hareesha K S, "Dynamic Clustering of Data with Modified Means Algorithm," International Conference on Information and Computer Networks, Volume 27, 2012.

[4]. Manpreet Kaur and Usvir Kaur, "Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection", International Journal of Advanced Research in Computer Science and Social , Volume 3, Issue 7, July 2013 ISSN: 2277 128X

[5]. Tapas Kanungo , David M. Mount , Nathan S. Netanyahu Christine, D. Piatko , Ruth Silverman and Angela Y. Wu, "An Efficient K-Means Clustering Algorithm: Analysis and Implementation ," IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 24, July 2002

[6]. Amar Singh and Navot Kaur, "To Improve the Convergence Rate of K-Means Clustering Over K-Means with Weighted Page Rank Algorithm," International journal of Advanced Research in Computer Science and Software Engineering, Volume 3, Issue 8, August 2012

[7]. Harpreet Kaur and Jaspreet Kaur Sahiwal, "Image Compression with Improved K-Means Algorithm for Performance Enhancement," International Journal of Computer Science and Management Research, Volume 2, Issue 6, June 2013

[8]. Osamor VC, Adebiyi EF, Oyelade JO and Doumbia S "Reducing the Time Requirement of K-Means Algorithm" PLoS ONE, Volume 7, Issue 12, 2012

[9]. 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, pages 959-963, 2012

[10]. Kajal C. Agrawal and Meghana Nagori, "Clusters of Ayurvedic Medicines Using Improved K-means Algorithm," International Conf. on Advances in Computer Science and Electronics Engineering, 2013.

[11]. M. N. Vrahatis, B. Boutsinas, P. Alevizos and G. Pavlides, "The New k-Windows Algorithm for Improving the k-Means Clustering Algorithm," Journal of Complexity 18, pages 375-391, 2002.

[12]. Chieh-Yuan Tsai and Chuang-Cheng Chiu, "Developing a feature weight self-adjustment mechanism for a K-means clustering algorithm," Computational Statistics and Data Analysis, pages 4658-4672, Volume 52, 2008

[13]. Guangchun Luo, Xiaoyu Luo, Thomas Fairley Gooch, Ling Tian, Ke Qin," A Parallel DBSCAN Algorithm Based On Spark", 2016, IEEE, 978-1-5090-3936-4

[14]. Dianwei Han, Ankit Agrawal, Wei−keng Liao, Alok Choudhary," A novel scalable DBSCAN algorithm with Spark", 2016, IEEE, 97879-897-99-4

[15]. Nagaraju S,Manish Kashyap, Mahua Bhattacharya," A Variant of DBSCAN Algorithm to Find Embedded and Nested Adjacent Clusters", 2016, IEEE, 978-1-4673-9197-9

[16]. Jianbing Shen, Xiaopeng Hao, Zhiyuan Liang, Yu Liu, Wenguan Wang, and Ling Shao," Real-time Superpixel Segmentation by DBSCAN Clustering Algorithm", 2016, IEEE, 1057-7149

[17]. Ilias K. Savvas, and Dimitrios Tselios," Parallelizing DBSCAN Algorithm Using MPI", 2016, IEEE, 978-1-5090-1663-1

[18]. Ahmad M. Bakr , Nagia M. Ghanem, Mohamed A. Ismail," Efficient incremental density-based algorithm for clustering large datasets", 2014, Elsevier Pvt. Ltd.

AUTHOR

Er. Paramvir Kaur Dhillon received the Bachelor degree in computer science from the Punjab Technical University, in 2015. She is presently a student of Master of Technology (Computer science) in Sant Baba Bhag Singh University. Her current research interests include data mining. She has her publication in 3rd DAV National Congress and presented a paper entitled "Expert System" in National Conference on Recent Trends in Computer Technology (RTCT-2014) and presented papers in National Conference on Emerging Trends on Engineering & Technology (ETET-2017) in University Inst. of Engg. & Tech. & University Inst. Of Computer, SBBS University, Punjab (India).