ENHANCED DENSITY BASED METHOD FOR CLUSTERING DATA STREAM

MR. K. SRIDHARAN[1], P. ANTONY ALDRIN[2], P. BHUVANESH[3], S. LOKESH[4], M. SUJINRAJ[5]

[1]*Associate Professor, Department of B.tech .Information technology Engineering, Panimalar engineering college, Tamilnadu, India.*

[2,3,4,5] *Students, Department of B.tech .Information technology Engineering, Panimalar engineering college, Tamilnadu, India.*

Abstract - *Data stream clustering is an active area of research in big data. It refers to clustering constantly arriving new data records and updating existing cluster patterns and outliers in light of the newly arriving data. Density-based algorithms for solving this problem have the promise for finding arbitrary shape clusters and detecting anomalies without prior knowledge of the number of clusters. a new incremental algorithm known as Enhanced Density-based Data Stream (EDDS) is developed to overcome limitations with the existing solutions. The algorithm detects clusters and outliers in an incoming data chunk, merges new clusters from the chunk with the existing clusters, and filters out new outliers for the next round. It modified the traditional DBSCAN algorithm to summaries each cluster in terms of a set of surface-core points. The algorithm applies the density-reachable concept of DBSCAN as its merging strategy and prunes the internal core points using a heuristic solution. The algorithm also removes the aged core points and outliers depending on a fading function. The paper investigates three versions of the algorithm for three possible representations of clusters where either all core points are maintained (EDDS-I), only core points of the new clusters from the incoming chunk are kept (EDDS-II), or only the surface-core points of the cluster shapes are kept (EDDS-III) to examine the balance between the efficiency gain for the algorithm and the amount of overhead time committed for pruning internal core points. The algorithm was evaluated on selected datasets using various quality measures.*

Key Words: **edds, dbscan, chunk, core, kmeans clustering**

1. INTRODUCTION

Advances in information and networking technologies have led to a rapidly growing flux of massive data, known as Big Data, in almost every sector of life ranging from the stock market, online shopping, banking, social media, and healthcare systems. Big data fundamentally refers to a huge volume of data that are generated by various applications and stored in different sources and locations with different formats. Big data requires frequent updating and analysis with the aim of the enhanced competitiveness and improved performance of institutions. One of the most important characteristics of big data is velocity. It means that data may arrive and require processing at different speeds. While for some applications, the arrival and processing of data can be performed in a batch processing style, other analytics require continuous and real-time analyses of incoming data streams. Data stream clustering is defined as the grouping of new data that frequently arrive in chunks with the objective of gain understanding about underlying cluster patterns that may change over time. It is also of interest to know the impact of the underlying cluster pattern changes to data objects outside the clusters, i.e. outliers. In this paper, a new algorithm called an Enhanced Density based Data Stream (EDDS) is presented. At any incremental round, the inputs consist of a new data chunk, summaries of the current set of clusters in terms of a set of core points, and a list of outliers from the previous iteration. The outputs to the next round consist of the summaries for a modified list of clusters and an updated list of outliers. The EDDS algorithm modifies traditional DBSCAN algorithm to present not only the outliers outside the clusters but also the summary of each cluster which include the core points on the surface of each cluster. In addition, it applies a merging strategy based on the density-reachable concept between core points to merge overlapped new and existing clusters. Moreover, it prunes the output clusters using a fading function to reduce the impact of aged core points and outliers whose relevance decrease over time. algorithm intends to reduce the computational costs of maintaining the output clusters by removing the data points inside each cluster and just keeping the core points on the surface of each cluster. A heuristics-based depth-first search method is embedded inside the traditional DBSCAN algorithm to locate the surface-core points of each cluster. The algorithm also prunes the core points inside a merged cluster to keep only the surface-core points for the cluster. To examine the trade-off between the efficiency gained and the amount of overhead computation needed for pruning, the paper presents three versions of the EDDS algorithm: EDDS-I maintains all the core points of each cluster, EDDS-II keeps all the core points of the new incoming chunk, and EDDS-III keeps only the surface-core points. The EDDS algorithm can adapt to changes in data over time by associating the minifying decay function, developed in, with each data point in surface-core and outlier. The evaluated on a selected collection of data sets using various measurements. The experimental results show that the proposed algorithm improves clustering correctness with a comparable time complexity to the existing methods of the same type. The structure of the algorithm is designed to be modular for easy accommodation of further improvements and the parallelization of the algorithm. The rest of this paper is organized as follows. State of the art of the related work on data stream clustering algorithms in the current literature. A systematic evaluation of the performance of the algorithm and compares it with one of selected existing algorithms through theoretical analysis and practical experiments using the synthesized datasets. A number of further issues regarding the proposed algorithm will be discussed. It concludes the work and outlines the possible future directions of this research.

2. MODULE DESCRIPTION:

2.1 Load data and convert data

In this module, the data is collected from the dataset and then it is loaded. The loaded dataset is converted into grid format and then processed.

2.2 creation and execution

In this module, the table is created for the loaded dataset. After creating the table, data is extracted from table which is created and then extracted dataset is loaded. DBScan is used to classify the attributes.

2.3 compute density value

In this module, the attributes are selected and then density value is calculated based on attribute selection.

2.4 Attribute Selection

In this module, the attributes are selected and weights of the outliers, distance of the candidate are calculated.

2.5 Clustering Data:

In this module, the data is clustered and disjoint fragments are generated and estimating the distance of the outliers.

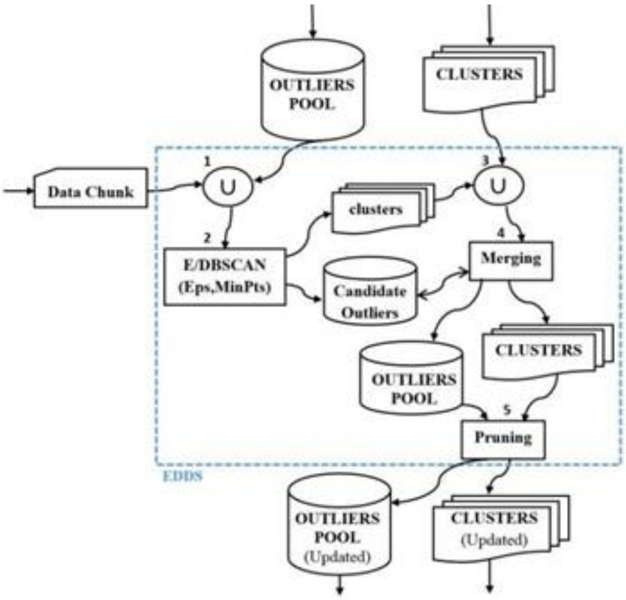
3. PROPOSED SYSTEM:

Edds proposed system develop and evaluate a new method to address this problem for micro-cluster-based algorithms. We introduce the concept of a shared density graph which explicitly captures the density of the original data between micro-clusters during clustering and then show how the graph can be used for re-clustering micro-clusters. proposed Clustering based subset Selection algorithm uses minimum spanning tree-based method to cluster features. our proposed algorithm does not limit to some specific types of data. Irrelevant features, along with redundant features, severely affect the accuracy of the learning machines. Thus, feature subset selection should be able to identify and remove as much of the irrelevant and redundant information as possible. More over,“ good feature subsets contain features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other. ”In our proposed Cluster based subset Selection algorithm, it involves 1) the construction of the minimum spanning tree from a weighted complete graph; 2) the partitioning of the MST into a forest with each tree representing a cluster; and 3) the selection of representative features from the micro-clusters.

4. SOFTWARE AND HARDWARE REQUIREMENTS

Processor	: Core2Duo
Hard Disk	: 500 GB.
Monitor	: LCD/LED.
Mouse	: Logitech.
RAM	: 4 GB.
Operating system	: Windows 7/8.
Coding Language	: JAVA/J2EE
IDE	: Netbeans 8.0.2
Database	: MYSQL

5. ARCHITECTURE:



6. CONCLUSIONS

A new incremental density-based algorithm EDDS for data stream clustering. The algorithm emphasizes on simplicity, modularity, and adaptively. The key ingredients of the algorithm are to keep the surface-core points using a variant DBSCAN, merge the overlapped clusters using a density-reachable principle, and to prune the output clusters using a heuristic solution. The algorithm proposed a novel way of representing output clusters using only surface-core points. The evaluation on various datasets demonstrated the effectiveness of the algorithm in finding correct and good quality clusters of various shapes and at the same time maintain a similar level of time complexity of other alternative Algorithms. Our future work will focus on enhancing the algorithm. Since the algorithm is modular, those enhancement efforts can focus on the main functions within the algorithm. The heuristics solution employed in the EDBSCAN function can be further improved by utilizing computational geometry and topology techniques to more accurately locate the surface cores and determining if a core point is inside a shape or outside. Besides, a fuzzy shape based cluster could be embedded into the Prune function to specify minimum surface-cores representative. Finally, distributed and parallel solutions could have embedded to present more efficient, robust, and reliable new version of EDDS algorithm.

7. REFERENCES

[1] C. Liu, R. Ranjan, X. Zhang, C. Yang, D. Georgakopoulos, and J. Chen, “Public Auditing for Big Data Storage in Cloud Computing -- A Survey,” 2013 IEEE 16th Int. Conf. Comput. Sci. Eng., pp. 1128–1135, Dec. 2013.

[2] E. Olshannikova, A. Ometov, and Y. Koucheryavy, “Towards Big Data Visualization for Augmented Reality,” 2014 IEEE 16th Conf. Bus. Informatics, pp. 33–37, Jul. 2014.

[3] M. Z. Islam, “A Cloud Based Platform for Big Data Science,” Dep. Comput. Inf. Sci. Linköping Univ., pp. 1–57, 2013.

[4] Yogita and D. Toshniwal, “Clustering Techniques forStreaming Data – A Survey,” pp. 951–956, 2012.