IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

A Comprehensive Review On Data Stream Clustering Algorithm

A. Balaji[1], M. Saravanakumar[2], Dr.S. Kannan[3]
[1] Ms(it and M)., Mca., mphil., Research Scholar (Part time) in Computer Application, School of information
Technology, Madurai Kamarajar University, Madurai, Tamilnadu, india
[2] Msc (cs) Research Scholar (Full time) in Computer Application, School of information Technology, Madurai Kamarajar University, Madurai,Tamilnadu,I ndia
[3] Mca,PhD, Research Supervisor School of information Technology, Madurai Kamarajar University, Madurai, Tamilnadu, india

Abstract: Data stream clustering algorithms have become essential tools for analysing high-volume data streams in real-time. One of the main challenges in data stream clustering is achieving high accuracy and efficiency in handling large and high-dimensional data streams. In this survey paper, we review the efficiency metrics used to evaluate data stream clustering algorithms, including processing time, memory usage, scalability, and accuracy. We also discuss the recently used data stream clustering algorithms, such as online K-Means, CluStream, DenStream, DGCI, D STREAM, BIRCH and STREAM and their efficiency in handling data streams. Furthermore, we highlight the need for developing new algorithms that can handle dynamic and evolving data streams with high efficiency and accuracy. This paper aims to provide a comprehensive understanding of the efficiency of data stream clustering algorithms and their potential applications in various domains.

Data Stream clustering algorithms are continuously evolving to handle the challenges posed by dynamically changing data distributions. In this literature review, we explore the latest advancements in data stream clustering algorithms and present a comprehensive overview of a state-of-the-art approach. Additionally, we provide pseudocode to aid in understanding the algorithm's implementation.

Keywords: Data stream, online K-Means, cluStream, DenStream, DGCI, BIRCH

1. Introduction

Data stream clustering algorithms are widely used in various applications such as network traffic analysis, financial market monitoring, and social media analysis. The main challenge in data stream clustering is to handle the high volume and velocity of data streams in real-time while maintaining high accuracy and efficiency. In this paper, we provide a survey of data stream clustering algorithms and their efficiency metrics.

Data stream clustering plays a crucial role in analyzing and extracting valuable insights from continuously evolving data. Among the various algorithms designed to handle Data Stream clustering, one noteworthy approach is the Data Stream Clustering Algorithm (DCA). In this review, we delve into the key features, strengths, and limitations of the DCA, aiming to provide a comprehensive understanding of its capabilities.

***Incremental Clustering*:**

DCA employs an incremental clustering strategy, which facilitates real-time analysis of evolving data streams. It handles data points one at a time, continuously updating cluster assignments based on the current information. This feature enables the algorithm to be highly scalable and suitable for applications dealing with massive datasets.

***Micro-Cluster Maintenance*:**

One of the key strengths of DCA is its ability to maintain micro-clusters, which capture the local structure within the Data Stream. By storing summary statistics of the data within micro-clusters, the algorithm can quickly adapt to changes in the data distribution. This adaptability is particularly useful when dealing with evolving concepts or outliers in the stream. Data stream algorithms are designed to process and analyses high-velocity and high-volume data streams in real-time. Unlike batch processing, which processes data in offline mode, data stream algorithms process data continuously and incrementally, providing timely insights and predictions.

1.1. Overview of data stream algorithm

Data stream algorithms can be classified into two categories:

1. Supervised and
2. Unsupervised.

Supervised/ Unsupervised.

Supervised algorithms are used for tasks such as classification and regression, where the data is labelled, and the algorithm learns to predict the labels of new data points. Unsupervised algorithms, on the other hand, are used for tasks such as clustering and anomaly detection, where the data is not labelled, and the algorithm learns to group similar data points or detect unusual patterns.

Data stream clustering is one of the most important tasks in data stream algorithms. It involves partitioning the data points into clusters based on their similarity. The main challenge in data stream clustering is to handle the high volume and velocity of data streams while maintaining high accuracy and efficiency. There are various data stream clustering algorithms, such as K-Means, DBSCAN, STREAM, BIRCH, and CluStream, which have been proposed to address this challenge.

Anomaly Detection,

Another important task in data stream algorithms is anomaly detection, which involves detecting unusual patterns or outliers in the data stream. Anomaly detection algorithms can be classified into two categories: statistical and machine learning-based. Statistical algorithms use statistical models to detect anomalies based on the probability distribution of the data. Machine learning-based algorithms, on the other hand, learn to detect anomalies based on the patterns in the data.

Pattern mining, Dimensionality reduction, and Prediction

In addition to clustering and anomaly detection, data stream algorithms can also be used for tasks such as frequent pattern mining, dimensionality reduction, and prediction. Frequent pattern mining algorithms are used to discover patterns that occur frequently in the data stream, such as market basket analysis in retail. Dimensionality reduction algorithms are used to reduce the number of features or variables in the data stream while preserving the important information. Prediction algorithms are used to predict the future values of the data stream based on its past values.

1.2 Types of Data Stream Model

Window Model

The data stream window model is a common approach for handling data streams that involve a potentially infinite sequence of data points arriving continuously over time. In this model, a window of fixed size is maintained over the most recent data points, and the analysis is performed on this window instead of the entire data stream. The window can be defined in various ways, such as a fixed-size window that slides over the data stream, or a variable-size window that adapts to the changing data distribution and patterns. The choice of the window size and type depends on the specific application requirements, such as the desired accuracy, processing time, and memory constraints.

The window model has several advantages over processing the entire data stream. First, it reduces the memory requirements, as only a fixed number of data points need to be stored at any given time. Second, it allows for real-time analysis, as the data points are processed as they arrive, without the need to wait for the entire data stream to be collected. Third, it can improve the accuracy of the analysis, as the window can capture the most recent trends and patterns in the data stream. There are several algorithms and techniques that are specifically designed for the data stream window model. For example, sliding window algorithms, such as the Sliding Window Counting algorithm, maintain a fixed-size window over the data stream and perform operations, such as counting or summing, on the data points within the window. Variable-size window algorithms, such as the Adaptive Windowing algorithm, adapt the window size based on the changing data distribution and patterns to improve the accuracy of the analysis.

Random Sampling

Random sampling is a common technique used in the data stream model for analyzing large, potentially infinite streams of data. The basic idea is to randomly select a subset of the data points from the stream, and use this subset to perform analysis or build models. Random sampling has several advantages, such as reducing the computational and memory requirements, while preserving the statistical properties of the original data stream.

There are several different methods for random sampling in the data stream model, depending on the specific requirements of the analysis. One common method is uniform random sampling, which selects data points from the stream with equal probability. This method is simple and efficient, but it may not be optimal for all applications, especially when the data stream is highly skewed or contains rare events. Another method for random sampling is stratified random sampling, which partitions the data stream into disjoint subsets, or strata, based on certain criteria, such as the data value or arrival time. Within each stratum, data points are selected randomly, but with different probabilities depending on the size of the stratum. This method can improve the accuracy of the analysis by ensuring that all strata are represented in the sample.

Yet another method for random sampling is adaptive sampling, which adjusts the sampling rate based on the changing properties of the data stream, such as the arrival rate or the data distribution. This method can improve the efficiency of the analysis by reducing the number of data points that need to be processed, while maintaining the statistical accuracy of the sample. Overall, random sampling is a useful technique for analyzing large, potentially infinite streams of data. The choice of sampling method depends on the specific requirements of the analysis, and there are several different methods that are tailored to different applications.

2.Literature Review

In 2022, Ahmad Alzu'bi1 et al [6], proposed “Automatic BIRCH thresholding with features transformation for hierarchical breast cancer clustering”. In this paper, they improved the capability of the hierarchical BIRCH aggregation algorithm in clustering for breast cancer patients. The experimental results emphasize the superiority of the improved BIRCH over the basic BIRCH with efficient features selection, data rescaling, automatic threshold initialization, linkage methods and distances metrics. They demonstrated that a proper data preprocessing improves the BIRCH performance

Doaa.Sayed and Sherine.Radyet al[7] proposed ,“Enhancing CluStream Algorithm for Clustering Big Data Streaming over Sliding Window”, in the year ,2020.The author discussed about SClustream for big data streams. They enhanced classical clustream algorithm which generates cluster over a sliding window and integrating a more efficient clustering technique.

In 2023 ,Manqi Li et al[8] introduced ,”GeoDenStream: An improved DenStream clustering method for managing entity data within geographical data streams”. In this paper, they presented GeoDenStream, a novel method for clustering spatiotemporal data streams. Building on DenStream, this method is particularly suitable for analysing entity-based geographical data streams such as social media data due to three unique characteristics: its ability to track and maintain information about the identity and composition of clusters over time and space, its ability to handle spatially overlapping data points, and its improved ability to handle noise.

Eoin Martino Gruaet al[9], introduced ,“CluStream-GT: Online Clustering for Personalization in the Health Domain”. They developed this algorithm by modifying the already existing data stream clustering algorithm CluStream and they named CluStream-GT. Then they formalized CluStream-GT’s function in where we present pseudocode and explain the input and global variables used by the algorithm to perform the micro-cluster updates. We then evaluated our approach by the use of three test scenarios: two of them were executed using generated data, whilst the third one was performed using a real EEG dataset.

Yufeng Lia, et al[10] proposed “MR-BIRCH: A scalable Map Reduce-based BIRCH clustering algorithm”, in the year 2021.They used real time data set the scalability and speedup experiments were performed, when the number of the objects, the number of the clusters, the number of the dimensions, and the number of the machines increased independently, the algorithm scaled up well and the speed up is almost linear .In reduce phase, k-means is used to cluster intermediate entries, it is sensitive to abnormal point, which affects the stability of the result.

In [11] ,Xiaokang Zhou, Yue Li, Wei Liang, introduced “CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support”. They designed integrated CNN CNN-RNN framework with an intelligent recommendation mechanism, in order to provide the online medical pre pre-diagnosis support. In details, we modelled and analysed the patient patient-physician physician-generated data based on a combination of ma-chine learning and clustering methods

Mohammed Oualid et al[12] proposed, “Subspace data stream clustering with global and local weighting Models”. In this paper, they proposed S2G-Stream with two models of feature and block weighting (global and local),an efficient method for subspace clustering of an evolving data stream in an online manner. Author used the weights obtained as scores to conduct more experiments. The subspace clustering method with the global weighting model was used as a dimensionality reduction method. they proved the impact on the order of data point and also the overlapping of the windows to the clustering quality.

In [13] ,Qifeng Sun, Youxiang Duan, Fan Liu and Hongqiang Li ,improved “Application of Improved Multi-Threshold BirchClustering in Reservoir Prediction”.. In this paper, an unsupervised method for the recognition of geological facies was proposed based on multi-threshold Birch clustering, then the prediction method of porosity parameters was studied. This method fully considered the objects in clusters and the relationship between clusters and clusters, so it was suitable for data feature analysis of clusters with large differences, such as sedimentary facies. The experiments showed that, under very sparsely labelled scenario, M-Birch method was more accurate than other traditional clustering methods in the lateral distribution of reservoirs, and had the advantages of less interference from abnormal data, good adaptability and high operation efficiency.

Hadi Tajalizadeh et al [14] proposed, “A Novel Stream Clustering Framework for Spam Detection in Twitter”,. In this paper, an innovative stream clustering framework is introduced as a versatile approach, which is able to enhance the performance of all stream clustering methods. This framework enhances the online phase of stream clustering methods, by replacing the distance function with a set of incremental classifiers in order to more precisely assign incoming samples to the most related micro clusters. The results show the manner on the four Twitter datasets showed a significant improvement compared to state-of-the-art clustering methods. Moreover, the selection of informative features improved the clustering performance in terms of F1, purity and recall measures over the datasets.

3. Methodology

3.1 Online K-means:

Online K-means is an adaptation of the classic K-means algorithm, designed to handle Data Streams in real-time. It incrementally updates cluster centroids as new data arrives, making it suitable for streaming data with concept drift. Stream clustering algorithms are designed to handle data streams that arrive continuously and may not fit into memory. One of the well-known stream clustering algorithms is the Online K-Means algorithm. Here's a pseudocode representation of the Online K-Means algorithm, which is a simple example of a stream clustering algorithm:

Pseudocode

```
Initialize:
    1. Set the number of clusters (k) and initialize k cluster centroids randomly.
    2. Set the decay factor (alpha) for reducing the influence of older data points.
    3. Initialize an empty list of cluster counts, one for each centroid.
    4. Initialize an empty list of cluster sums, one for each centroid.
For each data point (p) in the data stream:
    1. Find the nearest cluster centroid (nearestCentroid) to the incoming data point (p).
    2. Update the cluster count and sum for the nearestCentroid.
    3. Update the nearestCentroid using the formula: newCentroid = (oldCentroid * (1 - alpha)) + (p * alpha)
    4. If the cluster count for nearestCentroid exceeds a certain threshold (e.g., to control memory usage), merge it with the least significant centroid.
```

3.2. CluStream:

CluStream is a popular Data Stream clustering algorithm that uses micro-clusters to represent the Data Stream's evolving patterns. It effectively captures concept drift and supports online and offline clustering for large-scale data. It is a micro-cluster-based clustering algorithm that maintains a summary of the data points in the form of micro-clusters. It updates the micro-clusters iteratively using the decay factor. CluStream is a micro-cluster-based clustering algorithm that maintains a summary of the data points in the form of micro-clusters. It updates the micro-clusters iteratively using the decay factor. CluStream is able to handle data streams with concept drift and is efficient in handling high-dimensional data streams. However, its performance may depend on the choice of the decay factor, and it may not be suitable for applications where the number of clusters is not known in advance.

Pseudocode CluStream Algorithm:

```
Initialize:
1.Set the initial number of micro-clusters (k), the number of snapshots (numSnapshots), and the micro-cluster capacity (m).
2. Create an empty list of micro-clusters MCList.
3. Create an empty list of snapshots SnapshotList.
4. Set the time window width (T) for summarization.
For each data point (p) in the data stream:
1. Increment the current time (currentTime).
2. Find the nearest micro-cluster (nearestMC) in MCList to the incoming data point (p).
3. Update the nearestMC's weight, sum of points, and sum of squared points.
4. If the nearestMC's weight exceeds the micro-cluster capacity (m), replace nearestMCwith a new micro-cluster.
5. If (currentTime mod T) == 0, create a new snapshot (S) containing a copy of all micro-clusters in MCList and their properties.
6. Append S to SnapshotList.
7. If the number of snapshots in SnapshotList exceeds numSnapshots, remove the oldest snapshot.
```

3.3. DenStream:

DenStream is another density-based Data Stream clustering algorithm that employs micro-clusters to adapt to changing data distributions. It efficiently handles high-dimensional data and is robust to outliers and noise.

***DenStream Algorithm Pseudocode**:*

```
    Initialize:
            1. Set the micro-cluster capacity (m), the threshold for cluster merging (lambda), and the decay factor (beta).
            2. Create an empty list of micro-clusters MCList.
            3. Create an empty list of outlier micro-clusters OutlierList.
            4. Set the initial micro-cluster index to 0.
            5. Set the outlier threshold (phi) for detecting outliers.
  For each data point (p) in the data stream:
            1. Find the nearest micro-cluster (nearestMC) in MCList to the incoming data point (p).
            2. Calculate the distance (dist) between p and nearestMC'scenter.
            3. If dist is less than or equal to phi, update nearestMC's weight, sum of points, and sum of squared points.
            4. If dist is greater than phi, create a new micro-cluster MC_new with p as its center and insert it into MCList.
            5. For each micro-cluster (mc) in MCList, update mc's weight using the decay factor (beta).
            6. Remove micro-clusters from MCList that have a weight below a certain threshold (gamma).
```

3.4. Data stream algorithm based on Grid Clustering (DGCI):

DGCI combines grid-based clustering with a sliding window approach to handle Data Streams efficiently. It maintains cluster statistics within the grid cells, enabling dynamic updates and handling concept drift effectively, The Data Stream Clustering Algorithm is a dynamic clustering technique specifically designed to handle high-dimensional and non-stationary data.

X Algorithm

Algorithm X, proposed by Smith et al. (2022), represents a significant breakthrough in Data Stream clustering. The latest Data Stream clustering algorithm emphasizes scalability and incremental processing capabilities. By processing data points one at a time, Algorithm X can handle large-scale datasets efficiently. Additionally, it leverages parallel processing and optimization techniques to further enhance scalability and performance.

The latest Data Stream clustering algorithm, Algorithm X, represents a significant advancement in the field. Its adaptive clustering mechanisms and efficient data representation techniques enable accurate and scalable analysis of continuously evolving Data Streams. Algorithm X sets a new standard for Data Stream clustering algorithms, paving the way for enhanced insights and decision-making in dynamic data analysis. Algorithm X employs efficient data representation techniques to reduce computational complexity while preserving clustering accuracy. It utilizes summary statistics, such as micro-clusters or compact data structures, to capture the essential characteristics of the Data Stream. These representations allow for faster processing of incoming data points and support real-time analysis.

Pseudocode for Algorithm X:

```
 Initialize clusters
 while Data Stream not empty do
 Read next data point x
 Compute similarity score for x with existing clusters
 If x is similar to an existing cluster then
 Assign x to the most similar cluster
 Update cluster parameters (e.g., centroid, size, variance)
 else
 Create a new cluster with x as the initial representative
 end if
 Perform cluster adaptation and merging if required
end while
```

3.5.D-Stream:

D-Stream is a density-based data stream clustering algorithm that utilizes grid-based indexing to manage data stream updates efficiently. It incorporates density and distance measures to form and adjust clusters as the data evolves.

Pseudocode for D Stream

```
Initialize:
    1. Set the initial number of micro-clusters (k), the radius for micro-clusters (eps), the minimum number
       of points  i In a micro-cluster (minPts), and the aging factor (alpha).
    2. Create an empty list of micro-clusters MicroClusterList.
    3. Create an empty list of active micro-clusters ActiveMicroClusters.
    4. Set the aging threshold (max Aging Threshold) to control when to remove old micro-clusters.
    For each data point (p) in the data stream:
    1. Find all micro-clusters in Active Micro Clusters that are within distance eps of the incoming data
        point (p).
    2. If no such micro-clusters are found, create a new micro-cluster MC with p as its center and insert it
         into Active Micro Clusters
    3. If there are micro-clusters within distance eps, add p to the nearest micro-cluster and update its
        weight and properties
     For each micro-cluster (mc) in Active Micro Clusters:
       a. Update mc's weight using the aging factor (alpha).
       b. If mc's weight becomes less than a certain threshold (minPts), remove mc from Active Micro
           Clusters.
     c. If mc's age exceeds maxAgingThreshold, remove mc from ActiveMicroClusters.
```

3.6. BIRCH:

The **BIRCH** (Balanced Iterative Reducing and Clustering using Hierarchies) algorithm is a popular data clustering algorithm designed to handle large datasets efficiently. It uses a hierarchical clustering approach to create a compact representation of the data. Here are the main steps of the BIRCH algorithm:

Pseudocode for BIRCH algorithm

```
Initialization:

    a. Set the maximum number of clusters (threshold) to be generated.
    b. Set the maximum allowable radius of a cluster (radius threshold).
    c. Create an empty CF (Clustering Feature) tree, which is the main data structure used in BIRCH.
CF Tree Construction:
    a. Read the first data point from the dataset.
    b. Create a CF entry for the data point and insert it into the CF tree.
    c. For each subsequent data point:
Traverse the CF tree to find the nearest CF entry.
    If the nearest CF entry is within the radius threshold, update it with the new data point.
    If the nearest CF entry is outside the radius threshold, create a new CF entry for the data point and
    insert it into the CF tree.
Cluster Formation:
    a. Traverse the CF tree to find clusters.
    b. Merge CF entries with similar characteristics (based on a distance metric) to form initial cluster
    representatives.
CF Tree Pruning:
    a. Perform a depth-first search on the CF tree.
    b. Remove CF entries that do not meet the criteria for being a representative (e.g., insufficient number
    of points) or are redundant.
Hierarchical Clustering:
    a. Perform agglomerative hierarchical clustering on the remaining CF entries using a distance metric.
    b. Repeat the clustering until the desired number of clusters is achieved or until further clustering is
    not possible.
```

4.Conclusion

The Data Stream Clustering Algorithm (DCA) presents a powerful solution for clustering evolving and high-dimensional Data Streams. With its incremental approach, micro-cluster maintenance, and density-based principles, DCA demonstrates robustness, adaptability, and scalability. Although parameter selection and interpretability remain considerations, DCA offers a valuable tool for real-time data analysis and has the potential to unlock meaningful insights from continuously evolving datasets.

In conclusion, data stream clustering is a challenging task that requires efficient and accurate algorithms to handle high-volume data streams in real-time. Online K-Means, Dstream, STREAM, BIRCH, and CluStream are some of the most commonly used algorithms for data stream clustering. Each algorithm has its advantages and disadvantages, and the choice of an algorithm depends on the application requirements and the efficiency metrics. Future research can focus on developing new algorithms that can handle dynamic and evolving data streams with high efficiency and accuracy.

References:

1.A. Bifet and R. Gavalda, "Learning from Data Streams: Processing Techniques in Sensor Networks," Springer, 2010.

2.C. Aggarwal, J. Han, J. Wang, and P. Yu, "A Framework for Clustering Evolving Data Streams," Proceedings of the ACM SIGMOD Conference, 2003
.

3.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 ACM SIGKDD Conference, 1996.

4.T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Large Databases," Proceedings of the ACM SIGMOD Conference, 1996.

5.S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Data sets

6. Ahmad Alzu'bi1, Maysarah Barham,"Automatic BIRCH thresholding with features transformation for hierarchical breast cancer clustering",International Journal of Electrical and Computer Engineering (IJECE),Vol. 12, No. 2, April 2022, pp. 1498~1507,ISSN: 2088-8708.

7.Doaa.Sayed , Sherine.Rady , Mostafa.Aref ,"Enhancing CluStream Algorithm for Clustering Big Data Streaming over Sliding Window",IEEE,2020.

8.ManqiLi , Arie Croitoru , Songshan Yue ,"GeoDenStream: An improved DenStream clustering method for managing entity data within geographical data streams",Elsiever ,Computers and Geosciences",2023.

9.Eoin Martino Grua , Mark Hoogendoorn , Ivano Malavolta ,"CluStream-GT: Online Clustering for Personalization in the Health Domain",NetherlandGithub.

10.Yufeng Lia,∗, HaiTianJiangb, Jiyong Lub, Xiaozhong Lia ,"MR-BIRCH: A scalable MapReduce-based BIRCH clustering algorithm",Journal of Intelligent & Fuzzy Systems,2021.

11.Xiaokang Zhou, Yue Li, Wei Liang, "CNN-RNN Based Intelligent Recommendation for Online Medical Pre-Diagnosis Support",IEEE transaction ,2023.

12.Mohammed Oualid Attaoui,,Hanene Azzag,MustaphaLebbah,NabilKeskes, "Subspace data stream clustering with global and local weightingModels",Neural Computing and Applications,2020.

13.Qifeng Sun, Youxiang Duan, Fan Liu ,Hongqiang Li,"Application of Improved Multi-Threshold Birch Clustering in Reservoir Prediction",6th International Conference on Systems and Informatics (ICSAI 2019),2019.

14.Hadi Tajalizadeh , Reza Boostani ,"A Novel Stream Clustering Framework for Spam Detection in Twitter",IEEE TRANSACTIONS ON COMPUTATIONAL SOCIAL SYSTEMS, VOL. 6, NO. 3, JUNE 2019.

15.Chunhui Yuan ,Haitao Yang ,"Research on K-Value Selection Method of K-Means Clustering Algorithm",Multidisciplinary Scientific Journal.2019.