IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 12, Dec 2022**

Improving Performance of CluStream Algorithm

Ms. Neha Sharma
Assistant Professor
Sage university, Indore
Ne3haa@gmail.com

Dr. Shradhdha Masih
Associate Professor
SCSIT, DAVV Indore
Shraddhamasih@gmail.com

Abstract: The analysis of streaming data poses unique challenges due to the dynamic nature of the data and its potentially infinite size. Clustering algorithms designed for static datasets struggle to cope with streaming data. This research paper presents a comprehensive examination of the CluStream algorithm, a pioneering technique specifically developed for clustering data streams. The paper explores the key components of the algorithm, its underlying principles, advantages, and limitations. Additionally, it discusses the diverse applications of CluStream in various domains and highlights potential future research directions. This paper aims to offer researchers and practitioners a clear understanding of the CluStream algorithm and its implications for stream data clustering.

Keyword: CluStream, Data Stream, Clustering

1. Introduction

With the rise of big data, analyzing data streams has become crucial for extracting knowledge and insights in real-time. Traditional clustering algorithms, such as k-means and hierarchical clustering, assume a static dataset and struggle with the continuous arrival of streaming data. The CluStream algorithm addresses these challenges by providing an efficient and scalable approach for clustering data streams.

2. CluStream Algorithm

2.1 Data Stream Representation:

The CluStream algorithm represents a data stream as a sequence of time-ordered data points. Each data point consists of attribute values and a timestamp. The data stream is partitioned into fixed-size time intervals called micro-batches.

2.2 Micro-Clustering:

Micro-clustering is the initial step in the CluStream algorithm and focuses on summarizing the current micro-batch of data points. It aims to capture the statistical properties of the data efficiently. The micro-clustering process involves the following steps:

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12, Dec 2022

Each micro-cluster, denoted as Cj, maintains statistical measures including the number of data points (nj), the sum of attribute values (sum_j), and the sum of squared attribute values (sumsq_j). These statistics represent the characteristics of the data points within the micro-batch.

As new data points arrive in the micro-batch, they are assigned to the nearest micro-cluster based on a distance metric, typically the Euclidean distance. The assignment is performed by comparing the distance between the data point and the micro-cluster's center (computed as the mean of the micro-cluster's points) with a threshold value.

The statistical measures of the assigned micro-cluster are updated to reflect the inclusion of the new data point. This includes updating the number of data points, the sum of attribute values, and the sum of squared attribute values.

The micro-clusters are continuously updated as new data points arrive, allowing the algorithm to adapt to the changing micro-batch data distribution.

2.3 Macro-Clustering:

Macro-Clustering: Macro-clustering builds upon the micro-clusters and provides a summary of the entire stream history. It captures the evolution of clusters over time by combining information from multiple micro-clusters. The macro-clustering process involves the following steps:

Macro-clusters, denoted as CMj, store statistical measures including the number of micro-clusters in the macro-cluster (nmj), the weight of the macro-cluster (wj), and the center (cj) of the macro-cluster.

Periodically, the algorithm updates the macro-clusters by aggregating the statistical measures from the micro-clusters. This update process aims to maintain an accurate representation of the evolving data stream. The update involves combining the statistical measures from micro-clusters that have similar centers.

The macro-clustering step provides a global view of the clustering structure and allows for analysis of the entire stream history.

2.4 Cluster Evolution Monitoring:

Cluster Evolution Detection: Cluster evolution detection is a critical component of the CluStream algorithm that monitors the quality and evolution of the clusters. It utilizes statistical measures to detect concept drift or significant changes in the data distribution. The cluster evolution detection process involves the following steps:

Thresholds are defined for various statistical measures, such as the weight of macro-clusters, the distance between micro-clusters, or changes in the data stream characteristics.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **Volume 11, Iss 12, Dec 2022**

The algorithm periodically compares these statistical measures against the predefined thresholds to detect significant changes. If the measures exceed the thresholds, it indicates the need for cluster adaptation.

When cluster evolution is detected, the algorithm creates new macro-clusters to capture the changing data distribution. This involves merging existing micro-clusters or creating new ones to adapt to the evolving clusters.

3. Analysis

3.1 Silhouette Score

The silhouette score is a metric used to evaluate the quality and cohesion of clusters in a clustering algorithm. It provides a measure of how well each data point fits into its assigned cluster, taking into account both the distance to points within its cluster and the distance to points in neighboring clusters. A higher silhouette score indicates better clustering results. Here's a detailed explanation of the silhouette score:

Calculation of Intra-Cluster Distance: For each data point i in a given cluster, the average distance between i and all other data points within the same cluster is calculated. This value is denoted as a(i). The lower the value of a(i), the closer the data point is to the other points within its cluster.

Calculation of Inter-Cluster Distance: For each data point i in a given cluster, the average distance between i and all data points in the nearest neighboring cluster (i.e., the cluster to which i is not assigned) is computed. This value is denoted as b(i). The lower the value of b(i), the closer the data point is to the neighboring cluster.

Calculation of Silhouette Score: The silhouette score for each data point i is calculated using the formula:

silhouette(i) = (b(i) - a(i)) / max(a(i), b(i))

The silhouette score ranges from -1 to 1, where:

A score close to 1 indicates that the data point is well-matched to its assigned cluster and is distant from other clusters.

A score close to -1 indicates that the data point may be assigned to the wrong cluster, as it is closer to points in a neighboring cluster.

4. Result

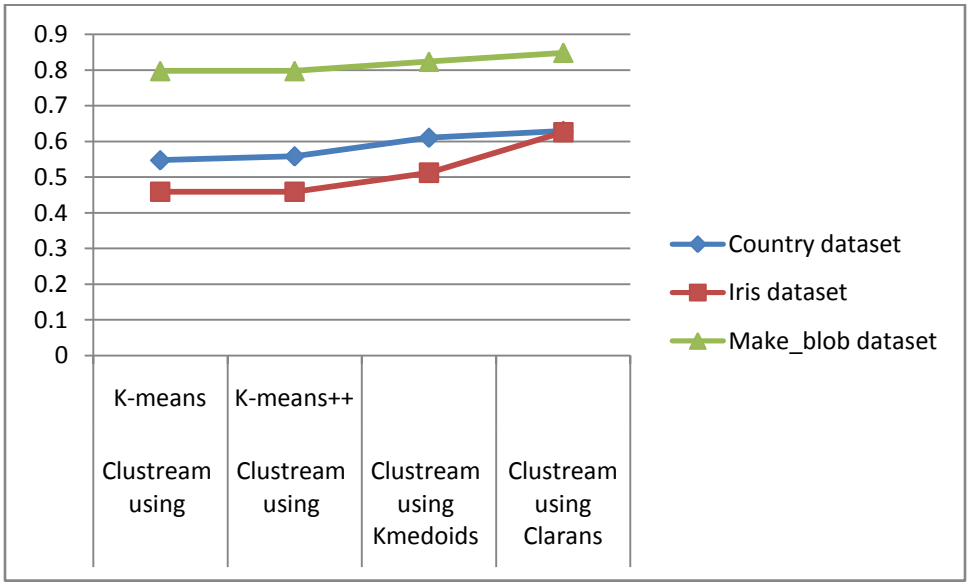
We perform CluStream algorithm on various data set (country dataset and iris dataset from kaggle) and measures the silhouette coefficient. CluStream uses K-means algorithm for cluster

ISSN PRINT 2319 1775 Online 2320 7876
Research paper Volume 11, Iss 12, Dec 2022

initialization. We uses K-mean++, Kmedoids and Clarans for as initial cluster analysis of CluStream algorithm. Experiment results are shown in table.

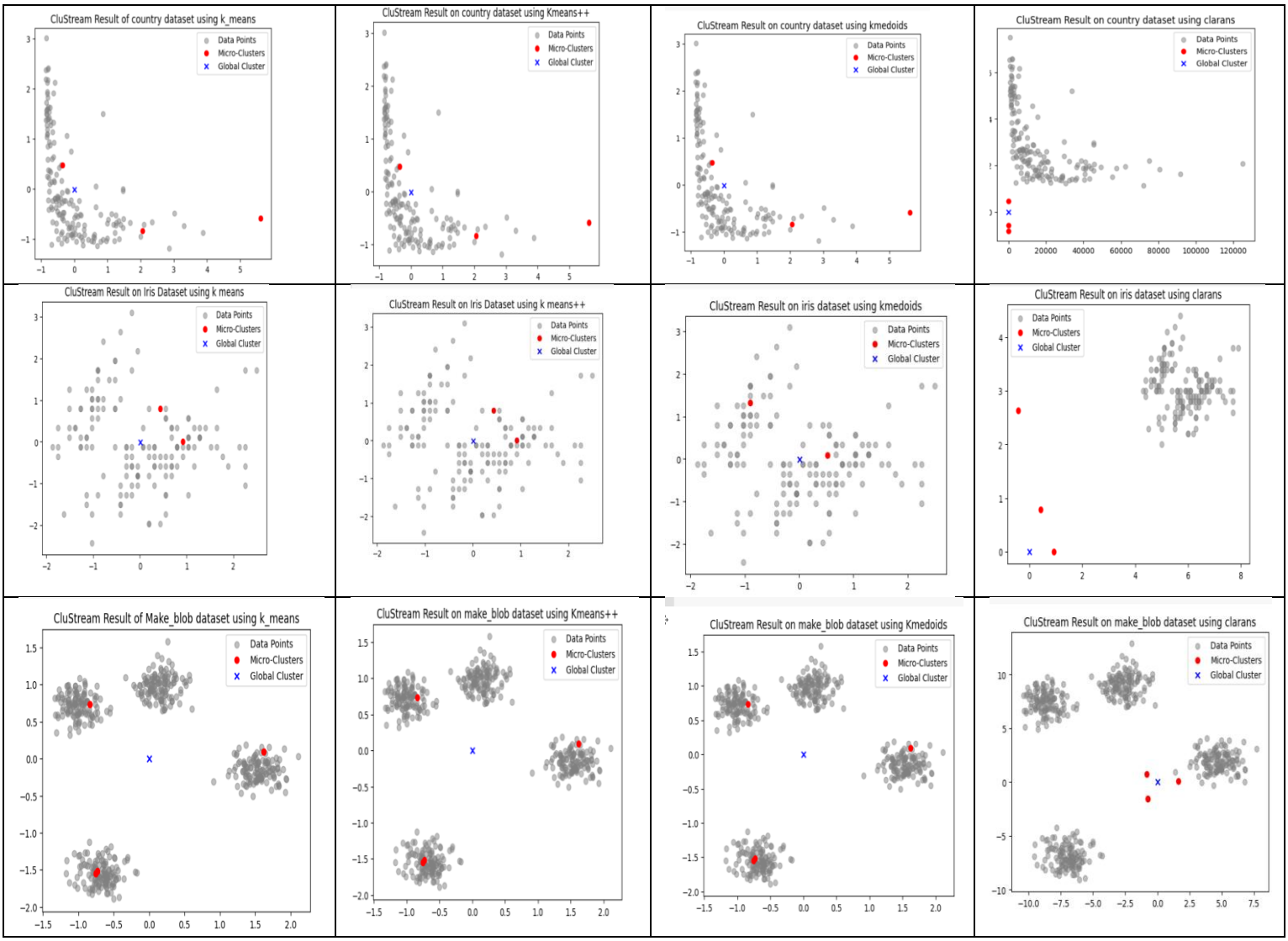
	Clustream using K-means	Clustream using K-means++	Clustream using Kmedoids	Clustream using Clarans
Country dataset	0.547	0.558	0.610	0.629
Iris dataset	0.459	0.459	0.512	0.625
Make_blob dataset	0.797	0.797	0.823	0.848

Table: Silhouette score



ISSN PRINT 2319 1775 Online 2320 7876

Research paper **Volume 11, Iss 12, Dec 2022**



5. Advantages of CluStream Algorithm

5.1 Scalability and Efficiency:

The CluStream algorithm is designed to handle large-scale data streams efficiently. By summarizing the data into micro-clusters and macro-clusters, it reduces the computational complexity and memory requirements compared to traditional clustering algorithms.

5.2 Real-Time Analysis:

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12, Dec 2022

CluStream enables real-time analysis of streaming data by maintaining up-to-date summaries of the data stream. This allows for timely insights and decision-making in dynamic environments.

5.3 Concept Drift Detection:

Due to its ability to monitor cluster evolution, CluStream is effective in detecting concept drift, which refers to changes in the underlying data distribution. The algorithm can adapt to these changes by creating new macro-clusters and maintaining accurate cluster representations.

5.4 Outlier Detection:

The CluStream algorithm provides a natural way to detect outliers in streaming data. Outliers are often represented by micro-clusters with significantly different statistical properties, making them identifiable during the clustering process.

6. Limitations of CluStream Algorithm

6.1 Parameter Sensitivity

The performance of the CluStream algorithm can be influenced by the proper selection of parameters, such as the decay factor and the threshold values for cluster evolution detection. Finding suitable parameter settings can be challenging and may require domain expertise or additional optimization techniques.

6.2 Handling High-Dimensional Data

CluStream may face difficulties when handling high-dimensional data due to the curse of dimensionality. As the number of dimensions increases, the effectiveness of distance measures and the interpretation of clustering results can be compromised.

6.3 Trade-Off Between Accuracy and Speed

To maintain efficiency, CluStream employs summarization techniques that can lead to a trade-off between clustering accuracy and processing speed. The algorithm sacrifices some level of precision to provide real-time analysis capabilities.

6. Applications of CluStream Algorithm

6.1 Network Traffic Analysis:

CluStream can be used for real-time monitoring and analysis of network traffic data. By clustering network packets, it enables the detection of network anomalies, intrusions, and DDoS attacks.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12, Dec 2022

6.2 Intrusion Detection:

The ability of CluStream to detect concept drift and adapt to evolving data distributions makes it suitable for intrusion detection systems. It can identify emerging attack patterns and detect anomalies in network traffic or system logs.

6.3 Social Media Analytics:

With the ever-increasing volume of social media data, CluStream offers an efficient solution for clustering and analyzing user-generated content. It can be used to identify trending topics, detect anomalies in user behavior, and personalize recommendations.

6.4 Sensor Data Processing:

In the Internet of Things (IoT) domain, CluStream can handle continuous streams of sensor data. It enables real-time monitoring, fault detection, and predictive maintenance in various applications, such as smart grids, environmental monitoring, and industrial systems.

7. Future Research Directions

7.1 Enhancing Accuracy and Robustness:

Further research is needed to improve the accuracy and robustness of the CluStream algorithm, especially in scenarios with complex data distributions and concept drift. Exploring advanced clustering techniques and novel distance measures could contribute to enhancing the algorithm's performance.

7.2 Handling Concept Drift in Evolving Data Streams:

Concept drift poses a significant challenge in stream data clustering. Future research should focus on developing mechanisms to efficiently handle concept drift, allowing the algorithm to adapt to changes in the data distribution and maintain accurate cluster representations.

7.3 Integration with Other Machine Learning Techniques:

Integrating CluStream with other machine learning techniques, such as anomaly detection or classification algorithms, can enhance its capabilities and extend its applications. Hybrid approaches could combine the strengths of different algorithms for more comprehensive data analysis.

7.4 Parallel and Distributed Implementations:

ISSN PRINT 2319 1775 Online 2320 7876
Research paper **© 2012 IJFANS. All Rights Reserved**, Volume 11, Iss 12, Dec 2022

As data streams grow larger and more complex, parallel and distributed implementations of the CluStream algorithm become essential. Research efforts should focus on designing efficient distributed algorithms that can handle high-velocity data streams across distributed computing platforms.

8. Conclusion

The CluStream algorithm provides an effective approach for clustering streaming data, addressing the limitations of traditional clustering algorithms in dynamic environments. Its ability to adapt to evolving data distributions, scalability, real-time analysis, concept drift detection, and outlier identification make it suitable for a wide range of applications. However, challenges remain in parameter selection, high-dimensional data analysis, and the trade-off between accuracy and speed. Future research should aim to enhance the algorithm's accuracy and robustness, develop strategies to handle concept drift, explore integration with other machine learning techniques, and design parallel and distributed implementations. By advancing the CluStream algorithm, we can unlock its full potential for stream data clustering and contribute to the field of data mining and knowledge discovery in the era of big data.

References

1. A.Hinneburg, E. Hinneburg, and D. A. Keim, “An efficient approach to clustering in large multimedia databases with noise,” in Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98). AAAI Press, 1998, pp. 58–65
2. Bifet, G. de Francisci Morales, J. Read, G. Holmes, and B. P Fahringer, “Efficient online evaluation of big data stream classifiers,” in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ser. KDD ’15.ACM, 2015, pp. 59–68.
3. Bezdek J. C. Pal N.R. Some new indexes of cluster validity. IEEE Trans. Syst.Man, Cyber. Part B 28 (3), 1998, pp. 301-315.
4. Gaber, Mohamed Medhat, Arkady Zaslavsky, and Shonali Krishnaswamy. "Mining data streams: a review." ACM Sigmod Record 34.2 (2005): 18-26.
5. Mahdiraji, Alireza Rezaei. "Clustering data stream: A survey of algorithms." International Journal of Knowledge-based and Intelligent Engineering Systems 13.2 (2009): 39-44.
6. Gao, Ming-ming, Chang Tai-hua, and Xiang-xiang Gao. "Application of Gaussian mixture model genetic algorithm in data stream clustering analysis."Intelligent Computing and Intelligent Systems (ICIS), 2010 IEEE International Conference on. Vol. 3. IEEE, 2010.
7. Zhou, Aoying, et al. "Distributed data stream clustering: A fast EM-based approach." Data Engineering, 2007. ICDE 2007. IEEE 23rd International Conference on. IEEE, 2007.