STM JOURNALS

Recent Trends in
Parallel Computing

ISSN: 2393-8749
Volume 12, Issue 2, 2025
May–August
DOI (Journal): 10.37591/RTPC

https://journals.stmjournals.com/rtpc

Research RTPC

Efficient Clustering Techniques for Data Stream Mining

Abhishek Kumar Saxena[1], Rashi Umar[2,*]

Abstract

Data mining mainly works on a massive database for storing heavy amount of data. It is generally essential for extracting the meaning insights from the massive, continuously growing database. The traditional method often struggles with sheer volume and the dynamic nature of the modern data. Data stream mining allows for the real-time analysis, means insights are generated as the data arrives, and not after the long batch process. This continuous flow processing is essential for the staying ahead of dynamic changes and then making the timely decision. A major difficultly is, data is not static, the pattern within the data changes overtime and this is called "concept drift". Basically, the data keep changing, so we have to change our method for getting accurate and the relevant result. Data streams often lack labels or due to absence of pre-labeled data in many streams unsupervised methods are crucial for pattern identification. Clustering algorithms are very useful for the data stream mining, because these group similar types of data points together for revealing the hidden structure and trends that might get unnoticed. This helps in understanding the behavior of the underlaying of the data stream. The aim of this research is to provide a clear understanding of diverse challenges and the problem definition in the data stream mining. It will deeply study how to deal with the data which changes overtime, as it is a big problem for maintaining clustering accuracy. This research also analyzes how to make an algorithm that uses less memory and time as data stream has these limits. This study will compare the different clustering methods, mainly focusing on basic idea and rules they use. The main goal of this study is to create and test the better clustering method that will be best for the data stream, and making them more accurate and faster. Data mining has various challenges but one of its major challenges is the phenomenon of concept drift, where properties of data changes overtime. Data stream mining algorithms should be operated under the resource constraints, like processing capabilities and limited memory. Clustering algorithm is mostly same as stream mining, because both of them group similar data together so that relationship and the trends are shown clearly. This study compares several categories of the clustering techniques, and this comparison is done based on their ability to handle the noise, adaption towards concept drift, scalability towards large dataset and many more. This study allows us to analyze ever-evolving data stream in the more accurate and timely manner.

***Author for Correspondence**
Rashi Umar
E-mail: rashiumarjnp@gmail.com

[1]Head, Department of Information Technology, Bansal Institute of Engineering and Technology, Lucknow, Uttar Pradesh, India
[2]Student, Department of Information Technology, Bansal Institute of Engineering and Technology, Lucknow, Uttar Pradesh, India

Received Date: May 09, 2025
Accepted Date: May 19, 2025
Published Date: June 4, 2025

Citation: Abhishek Kumar Saxena, Rashi Umar. Efficient Clustering Techniques for Data Stream Mining. Recent Trends in Parallel Computing. 2025; 12(2): 26–32p.

Keywords: Data mining, data stream mining, clustering, unsupervised learning, real time analysis

INTRODUCTION

The continuous growth of data across various applications is leading to the significant constraint for data storage capacity. The standard data mining tools do not work well with the large amount of data. So, we need to use data mining techniques or come up with new quicker solutions. They should have the ability of processing the data dynamically and to extract activable insights, because data is growing fast and storages are very limited. For the data stream, data is always flowing, and as data is

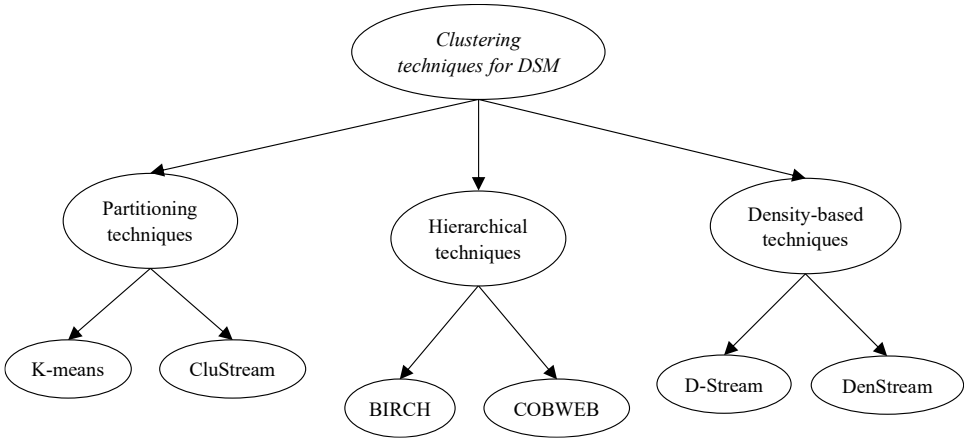

Figure 1. Clustering techniques for data streaming mining.

flowing, we have some challenges: Firstly, we get to look at the data only once, and secondly, the data keeps changing, so patterns also change. For handing these challenges, we use different methods like storing data, finding connection and grouping similar data. Many researchers have worked on this problem, mainly on concept drift [1, 2].

For making data representation smaller, researcher are actively working on various methods, process new data step-by-step and quickly to find unusual points [3]. In clustering data stream there are two main problems: first is concept change and second one is only being able to see the data only once. In concept change, we need to detect the changes as soon as they happen; it handles both normal and sudden changes. It tells the difference between the real changes and the random noise. When we see changes then we have to discard the old data and clusters, but it is important to keep some historical information. For being efficient, we need really fast algorithm because we get to see the data only once. Data stream is like a river; it keeps flowing and we cannot go back to see the data again [4]. Figure 1 illustrates the main categories of clustering techniques used for data stream mining.

This flowchart describes the categories of the clustering techniques which are applied in Design-Clustering Matrix. Clustering techniques for DSM have three main fundamental classifications:

- *Partitioning techniques*: They divide the dataset into pre-determined number of non-overlapping subgroups or clusters. It has two main algorithms, K-means and CluStream.
- *Hierarchical techniques*: This technique generates a hierarchical structure of the cluster, by using agglomerative clustering to unite smaller clusters or divisive clustering to separate large clusters. It also has two algorithms, BIRCH and COBWEB.
- Density-Based techniques group data points which are closely packed together forming outlier points which lie in low-density region. It also has two algorithms, D-Stream and DenStream.

LITERATURE REVIEW

System design needs more alteration, as we have spent lots of time for improving the algorithm which is used for finding patterns and grouping the data in stream, but we have not paid attention on how the whole system should be built. So, lack of focus made it harder to create the complete solution.

Many existing algorithms struggle for grouping the data without requiring much help from the experts. Specifically, they often need to know the things like expected number of the clusters and how much dense they are, which is not really possible in real world scenarios. They mostly do not easily remember and reuse the patterns that repeat overtime. The quality of the clustering result is often affected by the problem with how tightly the data is packed and the cluster separation. Furthermore, it is difficult to detect the changes in the data, and speed of processing is a critical factor in handling efficiency challenges. There are many methods which are not very good at finding unusual data point or outliers.

In practical application, normal statistical methods are limited by the insufficient data; so, a new approach is required for handling data stream, and to do it quickly and correctly. Also, we need to handle the clusters of arbitrary shapes also made up of different things. With recent advancements in getting better at grouping the data stream, we need efficient way for checking if our result is right or not. There are many methods but they work on only one specific method, and it is very difficult to find such a method which can work on all types of jobs. Beyond dealing with time and changes in data i.e. concept drift, space complexity is a very crucial part, especially in sensor network. Sensor network does not need large amount of memory for storing less data. So, it becomes necessary to develop a new clustering method that addresses these constraints.

In image processing, similarity analysis, customer preference clustering, network intrusion detection and time series data, a high-dimensional dataset is processed which presents a significant challenge. When the number of dimensions increases, the complexity of these datasets also gets increased which requires specialized data stream process techniques.

PRIMITIVE CLUSTERING METHODS

BIRCH is considered as the one of the older ways for grouping the data [5]. It is made with the help of regular databases, but because lot of data can be handled with this, people started using this in data streaming too. It mainly uses two techniques: first is micro clustering and second one is macro-clustering; these techniques help in handling huge amount of data and can find some problem with older grouping methods, like alteration is not possible once changes are done. It first looks at the data and tries to build tree (maps) which shows where the groups are, and in second step it clears the tree (map) by getting rid of data points which are far away (outlier), which helps in making the group clearer. The main problem or the disadvantage is that, it has limits on how much data a certain part can hold. Also, it is not suitable for groups because it uses techniques of radius or diameter for deciding where groups end [4].

STREAM is the method which is specifically made for grouping the data stream [6]. K-median version is used for grouping data which is based on how much error is present. In the first step, it groups all the data and tries to find the center of each group, and gives each center a weight based on how much data points are present in it. In next step, the center of each group is grouped again until it reaches to the top. It finds difficulty in handling time and data changes.

There is way for grouping the data stream by choosing the most relevant parts of the data [7]. Whenever new data arrives, it is kept in the closest group, even if some data are missing. It then decides which part is more important for grouping and removes the less useful data. This method is far much better than STREAM for complicated data, but it does not handle the data which changes overtime like STREAM does.

Clustering helps in fixing the problem of STREAM [8]. Clustering uses the ideas of both BIRCH and STREAM, with micro-clustering and macro-clustering, which is done in mainly two parts: first part is done when the data comes in, i.e. online component, and second part is done when it works later, i.e. offline component. Changes overtime can be tracked in this method; with the help of this method, we can answer the question about the past data. If data is uncertain, it does not give accurate result. So, the other method is introduced for fixing this and that method is UMicro [9]. K mean and k median is used by many grouping methods, but both have problems with outlier. K mean is very sensitive to outlier. It does not work efficiently with oddly shaped groups or group of different sizes. Number of groups needed will also be mentioned, several methods are there for making k mean faster [10, 11]. For complex text data, SPKMEANS was made [12]. The SPKMEANS algorithm can be explained using a statistical model of figuring out the likely group, based on the specific kind of distribution [13].

There are several issues with k-mean and people have tried to fix those issues. For best possible grouping of documents, a new method is introduced i.e. HKA [14]. HKA is an algorithm which uses a

technique called harmony search which helps in finding the best way to group data. Its work can be proved by using the math's idea 'Markov chain' [15]. For grouping gene data, a well-defined changed version of k-mean is present and it helps in avoiding bad grouping results. It builds groups in small-part using the center of the groups. But this method is very complicated and takes a lot of time. Fractal Clustering tries to group data just by looking things that are similar to themselves [16]. One by one it adds data point to groups putting them from where they change groups' "fractal" pattern least, so that it can find group of any shape. It handles the data points very efficiently which is very different from the rest (outlier); it can be used for the data which has lots of different kinds of information although it was not initially made for data streams. It does not work with the text data [17]. A new algorithm has been proposed for handling the data stream which is based on the fractal concepts. It groups data into arbitrary shapes. It can manage noisy and high dimensional data. It finds difficulty in handling different levels of shape complexity, automatically adjusting the setting for changing data, and the best way for finding the outliers.

DATA STREAM CLUSTERING METHODS

The aim of COBWEB method is to find the patterns which can be easy to understand and can group data step by step [18]. A tree structure is built by using the system and that tree represents the data groups. Each branch describes the characteristics of the data point in the group. It handles the outlier and manages the structure of the tree that can add complexity [19]. A cluster web page proposed the density-based clustering method, which gives better performance than COBWEB. It is a totally CluStream based framework that evolves to cluster text and expresses facts on massive scale [20].

Net Fuzz: A Neural community primarily based set of rules for Cloud assault Detection and evaluation is used for clustering records of the window of information using the ground truth and it summarizes the data of over 65500 points within a time body. Their test results show a completely affordable performance.

Information move uses totally Density-based clustering [21]. This method gets rid of drawbacks of totally k-means based strategies. This approach entails mapping every individual input information into a grid earlier than calculating the density of each grid and making use of a density-based clustering set of rules to the ensuing grids. The clustering should detect a cluster, no matter the form, and identify many outliers across the original clusters. It also cannot process specific and text statistics [22], and this version can be expanded to the textual content statistics streams and introduce a generalized semantic smoothing model. It additionally offers two on-line clustering algorithms OCTS and OCTSM which are two extended version models so that they may be applied in the clustering of massive-scale text data flow. Fully geared up with simultaneous characterization of live semantics of text facts Streams, these algorithms present a new sort of cluster information called cluster profile which now not only facilitates determine out the contents of relevant streams, but can also accelerate the clustering manner. In addition, they offer experimental effects that display the approach [23]. Clustering in facts circulate: A divide and triumph over technique for HAC. That being said, it is not a terrific alternative for apps that need to speed up what they are doing. Some guidelines are not of abecedaria to classify and outstrip. Dataset [24]: A degree divide-and-triumph over clustering set of rules of a statistics set, 2000 statistics factors. A frontrunner algorithm is first used to cluster the unique data into a large range of clusters [25]. Hierarchical clustering is applied to cluster the representatives obtained from those clusters. D-important: a course-Centralized k-manner with dynamic and incremental clustering the usage of Divide and overcome: Experimental consequences indicated were able to partition excessive exceptional and excessive-performance agencies of gadgets specially with high dimensionality items [26, 27].

There are few proposed techniques primarily based on graph concept. Recently 59 have offered a connectivity-primarily based methodology to cluster records streams. It has an awful performance, but the accuracy of their research is incredible. A brand-new factor could be processed through a repository of antique statistics and it cannot get motivation history in extraordinary time scale algorithm for

Weighted Fuzzy C Means facts circulate [28]. This approach is a variant of the same set of rules, FCM, where it most effectively makes a specialty of weighting on cluster and data factor iteratively, thus weighted clusters steadily cluster with the following data circulate analyzing chunk. Just enjoy is not taken under consideration, neither is the effect of outliers on the weighting procedure.

As mentioned, detecting outliers is one of the challenges in statistics circulate clustering problems. In reality, it is difficult to detect outliers among moving statistics [29]. Regardless of whether they are outliers or not, more attention should be given to the potential outlier points, and in the next incoming bite, let them persist instead of discarding spam outliers by means of checking modern-day chunk. It maintains the most suitable candidate outliers. We cannot fit the whole movement in a few bodily reminiscences, so in fact, the data flow ignores the parts of nearby region that are safe and do not include outliers; and those loose reminiscence are utilized in processing the subsequent set of information.

Excessive dimensionality is one of the essential apparent aspects in records complexity. This, in turn, makes it possible to automate the taking of a big number of measurements by using generation. Many researches have been performed for this reason. There are two important streams of researchers that are being highlighted at this time. At the same time as the previous institution is concerned with static high dimensional facts units [30–34], the latter institution shifts to the clustering of high dimensional records streams. HPStream [35] is possibly the most recognized one, which proposed a projected clustering to group excessive dimensional movement facts. It selects only a few of the capabilities and particular subspaces to cluster the records into it. The other maximum hard aspect of this set of rules is to pick out the proper subspace.

Comparison Table

Table 1 provides a comparison of key characteristics of different clustering algorithms.

ANALYSIS AND RESULT

CluStream

This algorithm has limited noise handling ability but it can adapt to Concept Drift. It is suitable for Spherical cluster shapes and has high scalability. For high dimensional data it has moderate suitability.

Den Stream

This algorithm can handle noise and it can adapt to Concept Drift. It is suitable for Arbitrary cluster shapes and has moderate scalability. For high dimensional data it has low suitability.

D-Stream

This algorithm can handle noise and it can adapt to Concept Drift. It is suitable for Arbitrary cluster shapes and has moderate scalability. For high dimensional data it has low suitability.

StreamKM++

This algorithm has no noise handling ability and has limited adaptation for Concept Drift. It is suitable for Spherical cluster shapes and has high scalability. For high dimensional data it has moderate suitability.

Table 1. Comparison of clustering algorithm.

Algorithm	Handles noise	Adapts to concept drift	Scalability	Cluster shapes	Suitable for high-dim data
CluStream	Limited	Yes	High	Spherical	Moderate
Den Stream	Yes	Yes	Moderate	Arbitrary	Low
D-Stream	Yes	Yes	Moderate	Arbitrary	Low
StreamKM++	No	Limited	High	Spherical	Moderate
HPStream	No	Limited	Moderate	Spherical	Yes

HPstream

This algorithm has no noise handling ability and has limited adaptions to Concept Drift. It is suitable for Spherical cluster shapes and has moderate scalability. For high dimensional data it has high suitability.

CONCLUSION

In this study we have discussed about the main problem which is faced when we try to group data in the data stream. Many data streams are high dimensional or have lots of different parts of information and a new method is required for this purpose. As data streams change all the time, so we require improved methods that can keep up with the change. Besides, we can say that data stream is like river, which is flowing in and out, and we can see the data once, so we have to create efficient and fast algorithm which can be helpful in finding the hidden pattern inside it.

REFERENCES

1. Tsymbal A. The problem of concept drift: definitions and related work. Computer Science Department, Trinity College Dublin. 2004 Apr 29; 106(2): 58.
2. Wang H, Fan W, Yu PS, Han J. Mining concept-drifting data streams using ensemble classifiers. In Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining. 2003 Aug 24; 226–235.
3. Barbará D. Requirements for clustering data streams. ACM sIGKDD Explorations Newsletter. 2002 Jan 1; 3(2): 23–7.
4. 35Khalilian M, Mustapha N. Data stream clustering: Challenges and issues. arXiv preprint arXiv:1006.5261. 2010 Jun 28.
5. Zhang T, Ramakrishnan R, Livny M. BIRCH: an efficient data clustering method for very large databases. ACM Sigmod Rec. 1996 Jun 1; 25(2): 103–14.
6. O'callaghan L, Mishra N, Meyerson A, Guha S, Motwani R. Streaming-data algorithms for high-quality clustering. In Proceedings IEEE 18th international conference on data engineering. 2002 Feb 26; 685–694.
7. Asbagh MJ, Abolhassani H. Feature-based data stream clustering. In 2009 Eighth IEEE/ACIS International Conference on Computer and Information Science. 2009 Jun 1; 363–368.
8. Aggarwal CC, Philip SY, Han J, Wang J. A framework for clustering evolving data streams. In Proceedings 2003 VLDB conference. Morgan Kaufmann. 2003 Jan 1; 81–92.
9. Aggarwal CC, Philip SY. A framework for clustering uncertain data streams. In 2008 IEEE 24th International Conference on Data Engineering. 2008 Apr 7; 150–159.
10. Pelleg D, Moore A. Accelerating exact k-means algorithms with geometric reasoning. In Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. 1999 Aug 1; 277–281.
11. Stoffel K, Belkoniene A. Parallel k/h-means clustering for large data sets. InEuro-Par'99 Parallel Processing: 5th International Euro-Par Conference Toulouse, France, August 31–September 3, 1999 Proceedings 5. Berlin Heidelberg: Springer; 1999; 1451–1454.
12. Dhillon IS, Modha DS. Concept decompositions for large sparse text data using clustering. Machine Learning. 2001 Jan; 42: 143–75.
13. Banerjee A, Ghosh J. Frequency-sensitive competitive learning for scalable balanced clustering on high-dimensional hyperspheres. IEEE Trans Neural Netw. 2004 May 10; 15(3): 702–19.
14. Mahdavi M, Abolhassani H. Harmony K-means algorithm for document clustering. Data Min Knowl Disc. 2009 Jun; 18(3): 370–91.
15. Bagirov AM, Mardaneh K. Modified global k-means algorithm for clustering in gene expression data sets. In ACM International Conference Proceeding Series. 2006 Dec 1; 246: 23–28.
16. Barbará D, Chen P. Using the fractal dimension to cluster datasets. In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. 2000 Aug 1; 260–264.
17. Kailing K, Kriegel HP, Kröger P. Density-connected subspace clustering for high-dimensional data. In Proceedings of the 2004 SIAM international conference on data mining. Society for Industrial and Applied Mathematics. 2004 Apr 22; 246–256.

18. Niu D, Dy JG, Jordan MI. Iterative discovery of multiple alternative clustering views. IEEE Trans Pattern Anal Mach Intell. 2013 Sep 23; 36(7): 1340–53.
19. Chehreghani MH, Abolhassani H, Chehreghani MH. Improving density-based methods for hierarchical clustering of web pages. Data Knowl Eng. 2008 Oct 1; 67(1): 30–50.
20. Aggarwal CC, Yu PS. A framework for clustering massive text and categorical data streams. In Proceedings of the 2006 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics. 2006 Apr 20; 479–483.
21. Chen Y, Tu L. Density-based clustering for real-time stream data. In Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 2007 Aug 12; 133–142.
22. Li X, Yang J, Wang Q, Fan J, Liu P. Research and application of improved k-means algorithm based on fuzzy feature selection. In 2008 IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery. 2008 Oct 18; 1: 401–405.
23. Guha S, Meyerson A, Mishra N, Motwani R, O'Callaghan L. Clustering data streams: Theory and practice. IEEE Trans Knowl Data Eng. 2003 May 13; 15(3): 515–28.
24. Wishart D. Number of clusters. Wiley StatsRef: Statistics Reference Online. John Wiley & Sons; 2014 Apr 14.
25. Duda RO, Hart PE. Pattern classification. New Jersey, United States: John Wiley & Sons; 2006.
26. Khalilian M, Boroujeni FZ, Mustapha N, Sulaiman MN. K-means divide and conquer clustering. In 2009 IEEE International Conference on Computer and Automation Engineering. 2009 Mar 8; 306–309.
27. Lühr S, Lazarescu M. Incremental clustering of dynamic data streams using connectivity based representative points. Data Knowl Eng. 2009 Jan 1; 68(1): 1–27.
28. Wan R, Yan X, Su X. A weighted fuzzy clustering algorithm for data stream. In 2008 IEEE ISECS International Colloquium on Computing, Communication, Control, and Management. 2008 Aug 3; 1: 360–364.
29. Elahi M, Li K, Nisar W, Lv X, Wang H. Efficient clustering-based outlier detection algorithm for dynamic data stream. In 2008 IEEE Fifth International Conference on Fuzzy Systems and Knowledge Discovery. 2008 Oct 18; 5: 298–304.
30. Fraley C, Raftery AE. Model-based clustering, discriminant analysis, and density estimation. J Am Stat Assoc. 2002 Jun 1; 97(458): 611–31.
31. Alijamaat A, Khalilian M, Mustapha N. A novel approach for high dimensional data clustering. In 2010 IEEE Third International Conference on Knowledge Discovery and Data Mining. 2010 Jan 9; 264–267.
32. Thangavel K, Pethalakshmi A. Dimensionality reduction based on rough set theory: A review. Appl Soft Comput. 2009 Jan 1; 9(1): 1–2.
33. Peng J, Tang CJ, Yang DQ, Zhang J, Hu JJ. Similarity computing model of high dimension data for symptom classification of Chinese traditional medicine. Appl Soft Comput. 2009 Jan 1; 9(1): 209–18.
34. Xu R, Wunsch D. Clustering. New Jersey, United States: John Wiley & Sons; 2008 Nov 3.
35. Aggarwal CC, Han J, Wang J, Yu PS. A framework for projected clustering of high dimensional data streams. In Proceedings of the Thirtieth international conference on Very large data bases. 2004 Aug 31; 30: 852–863.

