Data Stream Partitioning Algorithms for Big Data Analytics: A Review

DOI: 10.26573/2018.12.2.4
Volume 12*, Number* 2
May 2018*, pp.* 135-148

Hemant Kumar Singh
SMS Institute of Technology
(hemantbib@gmail.com)
Vinodani Katiyar
DSMNRU
*(*drvinodini@gmail.com*)*

As technology advanced, it opened new ways of continuous data gathering. In several applications like network monitoring, Walmarts are creating huge volume of data, so it is not possible to store such high volume, multidimensional data on physical storage medium i.e. available to analyze only once. Various existing mining techniques will not be efficient for such type of streaming data. So these existing data mining techniques need to be enhanced for processing big data streams. This paper takes a critical review of each of the four types of stream clustering algorithms and concludes with some critical discussions, advantages and disadvantages of each type of algorithm as well as gives some future research directions.

Keywords: Big Data, Data Stream, Stream Clustering, Map Reduce, IOT

1. Introduction

Today several companies have concluded that Big Data is not just a slogan. It has become a new reality of the big business life. In recent years data is coming from many dimensions which are huge, fast and in different formats. In year 2002 the 40 GB to 80 GB hard disk was large and sufficient in comparision to normal data storage but today data is generating in terabytes(2^{40}), Petabytes (2^{50}), exabytes (2^{60}), Zetabyte (2^{70}) and in yotabytes (2^{80}). Only the facebook generated data is more than 1000 TB a day. Therefore it is a challenge to store and process such large amount of data. Furthermore mobile phones call data record (CDR) is becoming popular in different aspects of analysis to significantly enhance our daily lives. Today it can be forecasted that Internet of Things (IOT) uses in near future will increase and it will generate unexpected data. In IOT persons and devices are loosely coupled with each other and all the services are mobile operated [1]. Suppose you are coming from a journey and it is very hot outside. Now you want to switch on the A.C. of your room 10 minutes before reaching home. IOT makes it possible to switch on A.C. of home using your mobile phone to make your room cool while driving your car. So with the enhanced use of IOT and CDR huge data will be generated and so we must be ready to handle such large data , to analyze such data and also to generate predictions for future to make living style more better and comfortable [2].Today we are having problem with system capacity, Algorithmic designs and business models. We need to work in this direction to minimize these issues in the near future.

2. Data Stream Clustering

Companies are often confused with what to do next whenever they look for large data. The solution is in data streaming. When huge amount of data that is not in rest needs to be analyzed in real time to extract hidden knowledge. The best answer is data streaming. Before this data was stored in operational data warehouses and analyzed offline. Therefore real-time predictions could not be generated through offline processing. If companies want to design new future strategies they need to analyze to large data sets to find out customer requirements. So companies must be aware with the leading factors of change and how will these factors affect the services and product that a company is planning to provide in the future. For example an insurance company can compare the number of accidents happening in a large geographic reason with weather conditions. Several companies are using big data analytics to find out new medicines. But the analytics should be quick and realistic. When the data volume is very large it has the following various mining challenges [1, 9]

1. The whole volume of data needs to be processed in single pass. So designed stream mining algorithms should process the data in one pass.
2. This consist of temporal locality that changes with time. Data may evolve with time. So straight forward adaption of single pass algorithm will not work better. Therefore Stream mining algorithms should be designed keeping it in mind that it should efficiently work on data evolving with time.
3. Data streams are mined in distributed way on cluster machines and furthermore individual processor has limited capacity of processing power and memory.
4. Clustering outcome are also affected by noise in data therefore the clustering algorithm will have to work well in case data is noisy.

In last few years a lot of data stream clustering algorithms have been proposed that are based on hierarchical clustering method, partitioning based clustering method, grid-based clustering method, density-based and model-based clustering methods [3, 4].

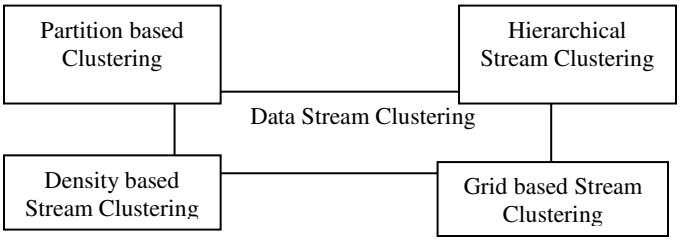

Figure 1 *Taxonomy of Stream Clustering*

The insight of big data in near future is health care sector that will help doctors in improving decision making for particular diseases like Cancer, Asthma, TB etc. Massive online analytical tools can be used for finding the insights of data. The utility of big data may be seen in health care monitoring. For example suppose a B.P. patient is running on a treadmill with his fitness wearing device. As his B.P. starts increasing device automatically informs the doctor about the status but person does not feel anything. Now patient receives the call from doctor to take the suggested medicine since his B.P. is high. Big data methods are by now being used to observe

babies in infancy and ill baby unit. By recording and examining each heart beat and breathing pattern of each baby, they can forecast infections 24 hours before any physical symptoms come into view. Big data analytics can tell the progress of infectious disease and its rate. Combining data from medical with social media, analytics enables us to supervise flu occurrence in real-time, simply by snooping to what people are saying. At present it is looking rubbish but it is reality of future

2.1 Partitioning based Data Stream Clustering Algorithms

Various partitioning methods like k-means, k-median has been proposed that are based on partitioning representatives. These methods define a cluster on a set of data points. Data point will be the member of which cluster, depends on distance from cluster. Data points will be the part of that cluster which is closest. These methods are iterative and need multiple passes. But in case of streaming the required data is already available for finding representative only in one pass. Following are the methods used for stream clustering based on partioning

2.1.1 Lsearch

It is considered that data arrived in large pieces (chunks) A_1, A_2, $A_{3, \ldots,}A_n$. A_i implies that it is a set of points that fits in primary memory. It is easy to convert a stream in to chunked stream by waiting till required point reaches in A_i.[5]

STREAM needs a simple, quick, constant-factor educated guess k-Median subroutine. It also offers flexibility in k. The LSEARCH algorithm does not straight forward solve k-Median but it could be utilized as a subroutine to a k-Median algorithm. LSEARCH clustering is useful to cluster high quality data stream.It gives better results than Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH).

2.1.2 Variations of K-Means Algorithm

K-means clustering is applied for cluster analysis.It divides n objects into k clusters on the basis of nearest mean. It means that object will be assigned to the cluster which has less mean difference. This way data space is partitioned in to bounded cells. Binary data streams are clustered using k-means algorithm. It's deviations are

1. On-line K-means
2. Incremental K-means
3. Scalable K-means.

These variants of k-means give better solutions in minimum time. Better solutions are located using Mean-based initialization and incremental learning. A easy set of adequate statistics and operations with sparse matrices constructs it quick. An on-line summary table of clusters is preserved. The K-means deviations are estimated on the basis of results quality and swiftness. These algorithms are very efficient to examine transactions [6].

2.1.3 CluStream

CluStream is a partitioning technique which clusters data streams in two steps that are called online micro clustering and offline macro clustering. Micro clustering summary statistics is obtained from the data stream then it is input to online micro clustering to perform the analysis.

The limitations of this clustering technique are as follows

- This algorithm is not able to obtain random sized clusters. Even k-means algorithm supports to find spherical clusters.
- It is difficult to find outliers and noise.
- It has better support for small data streams as the data stream size increases it's performance decreases. The reason is that is needs more than one pass.

Due to these limitations CluStream algorithm is mostly used to compress raw data streams to create micro clusters through online process and then it's result is given to offline phase [7, 8].

2.1.4 HP Stream

HP Stream is extension of CluStream algorithm. It is created to remove limitations of CluStream algorithm. Since CluStream algorithm does has better support for multi-dimensional data streaming even HP Stream performs well in case of high dimensional data [9,10].

2.1.5 SWClustering Algorithm

This SWClustering algorithm is able to discover clusters in data streams over the sliding window model. It uses exponential histogram cluster feature (EHCF) and it can incarcerate in-cluster evolution. This algorithm preserves aggregates over the sliding window model. It is capable of discovering clusters based on the synopsis formed by EHCF. The limitation of this algorithm is that it can not discover arbitrary shaped clusters and also does not has any provision for handling outliers [11, 12].

2.1.6 STREAMKM++

STREAMKM++ has a lot of similarity with k-means algorithm that is most suitable for clustering data streams from a Euclidean space. It generates a small weighted data stream sample. It uses k-means++ algorithm as a randomized seeding method to find the beginning values for the clusters. The small samples are generated using coreset generations using a coreset tree for fastness, As the cluster center increases it provided better quality outcome than Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) and Stream L Search, but time complexity of poor as compared to BIRCH in terms of running time is faster than STREAMKM++[13] .

Table 1 *Summary Table of Strong and Weak Points of Partitioning Algorithms*

Algorithm Name	Strong feature	Weak point
LSEARCH	LSEARCH clustering is useful to cluster high quality data stream. It gives better results than BIRCH	Does not have better support for evolving data over the time
variations of k-means algorithm	Variants of k-means gives better solutions in minimum time. Better solutions are located using Mean-based initialization and incremental learning	Unable to determine arbitrary shape clusters
CluStream	Does not has better support for multi-dimensional data streaming	Take no account of the attenuation of history data or the importance of recent data
HPStream	Has better support for multi-dimensional data streaming	Not suitable for handling irregularly distributed data streams, and interfered by noise easily

3. Hierarchical based Stream Clustering Algorithms

In hierarchical clustering technique a tree of clusters is formed with the given data that is being used for data summarization and visualization. This is based on binary-tree data structure and here tree is named as dendrogram. Once the dendrogram is created then right number of clusters can be chosen by dividing the tree at various levels to acquire diverse clustering solutions for the alike dataset and also the clustering algorithm need not to run again. Hierarchical clustering can be done in bottom up as well as top-down manner. Even both ways make use of concept of dendrogram while data clustering, they might produce totally different result sets depending on the principle used during the process of clustering . In this clustering once a merge or split is carried out, it can never be undoing. Various techniques like BIRCH, E-Stream, HUE- Stream are proposed hierarchical clustering for creating quality clusters. This algorithm creates random sized cluster and it needs two parameters that are radius and minimum number of data points in a cluster [14,15,16].

3.1 Birch

Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH)[17,18] creates the partition dynamically and progressively to multidimensional data points to produce quality partitions within available resources in a single scan. Initially BIRCH was not designed for data stream portioning and it is not very efficient for the data points that are rapidly changing. Key feature of BIRTCH is introduction of new data structure called a clustering feature or CF tree. CF is very efficient as well as sufficient for portioning whole data set.

Structure of CF is defined as triplet <NDP,LS,SS> where NDP is cluster data points, LS is sequential addition of NDP data points, SS is squared addition of data points.CF vector has two features

1. Inclusion of a data point x in cluster updates the sufficient statistics that is as follows

   $NDP_i \leftarrow NDP_i + 1$

   $LS_i \leftarrow LS_i + Z$

   $SS_i \leftarrow SS_i + Z^2$

This is Called Incrementality

2. Suppose $CF_1 = <NDP_1, LS1, SS1>$ and $CF_2 = <NDP_2, LS2, SS2>$ are two disjoint cluster's CF vector then merging of them will be equal to total of their parts. It allows merging of two or more sub clusters incrementally with no access of data set.

$$CF_1 + CF_2 = (NDP_1 + NDP_2, LS_1 + LS_2, SS_1 + SS_2)$$

CF tree is a height balanced tree shown in following Figure. It consists of hieratical clustering structure of whole data set. Here B is maximum number of CFs at each level

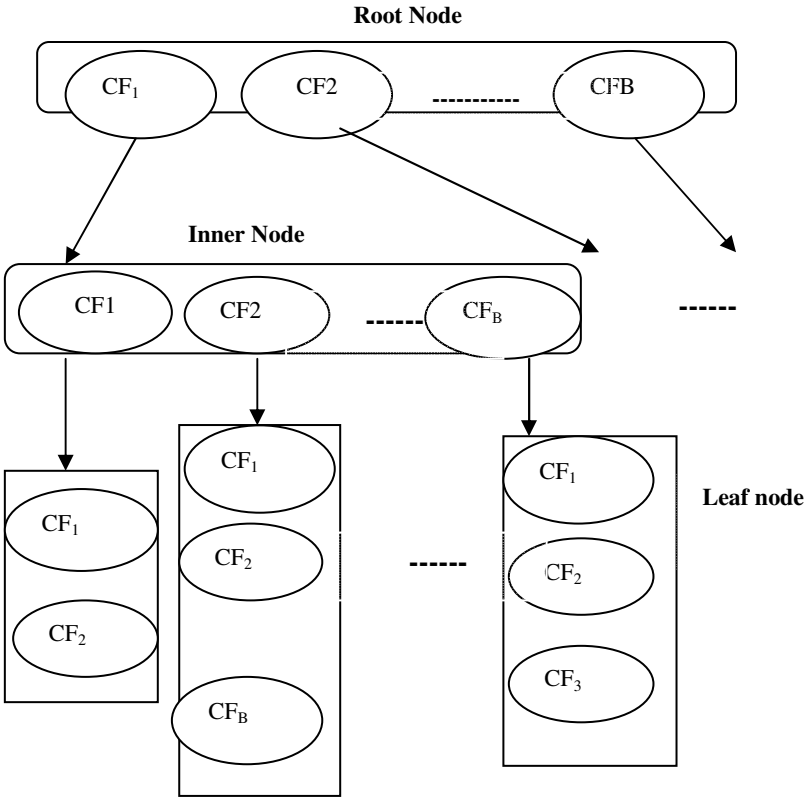

Figure 2 *CF Tree in Birch*

3.2 E-Stream

This is the evaluation based technique for clustering that supports monitoring and change detection of clustering. It divides data evolution in five parts i.e. disappearance, appearance, merge, split and self evolution. To store summary statistics it uses α bin histogram. Here each cluster is represented as fading cluster structure that uses α bin histogram of all features of data set. A cluster histogram data values are used to find out the cluster splits. Each bin range is found by (maximum – minimum) feature value divided by α. As maximum and minimum value varies new range is calculated and values of each range are updated by the intersection of old and new ranges. A histogram is drawn for each feature value of cluster and used to split active cluster.

The following function is used to calculate the closest active cluster. Suppose C is an active cluster and dp be a data point then cluster point distance calculated as

$$dist\ (C, dp) = \frac{1}{d}\sum_{j=1}^{d}\frac{\left|center_c^j - x^j\right|}{\left|radius_c^j\right|}$$

and cluster to cluster distance is calculated as

$$dist\ (C_a, C_b) = \frac{1}{d}\sum_{j=1}^{d}\left|center_{c_a}^{j} - center_{c_b}^{j}\right|$$

Here C_a and C_b are the two clusters.

E-Stream clustering starts as empty and each new point is included in existing cluster or new cluster is formed. If any cluster does not meet the defined density criteria, It is taken as inactive and isolated until it achieves required weight. Here weight of cluster is number of assigned data elements. The cluster that does not achieve required weight till a particular time, is considered as inactive and so is deleted from data space.

3.3 Hue Stream

Hue stream is extension of e-stream. It has the better support for uncertainty in heterogeneous data i.e. containing numerical and categorical attributes. This uncertainty creates double difficulty that has high volume and data uncertainty. This uncertainty is created due to reading error of sensors and other hardware. In some cases this error can be approximated using statistical tools like standard deviation, probability density functions etc. Umicro algorithm is proposed to deal with uncertain data stream that increases quality of micro cluster.

HClu stream enhances cluster feature vector to include categorical features and replaces the k-means clustering and is able to handle heterogeneous attribute's centroid plus discrete histogram attribute deployed to represent micro cluster as well as k prototype algorithm is used to build micro cluster and macro cluster.

3.4 Clus Tree

It is a single pass stream clustering algorithm with very less use of memory. It is parameter free clustering algorithm. It is very perfect clustering model. It also considers concept drift, novelty and outliers. It uses exponential time dependent decay function. It uses approach of apriori assumptions for deciding size of cluster but also adapts the size by self. It is any time algorithm and arranges micro cluster in a tree structure for accessing it fast and by self adapts size of micro cluster on varying data point.

4. Density Based Clustering

It is based on the linking between reasons and density functions. In density based clustering dense object locations in data space are taken as cluster that are separated by sparse density areas called noise. Den Stream [19] is a density based algorithm. It uses core-micro-cluster to summarize clusters. To retain and differentiate the outliers and potential clusters, this method presents core-micro-cluster and outlier micro-cluster structures. It uses two phase scheme, in first phase algorithm uses fading window model to generate summary of data. In second phase this summary of data is used to generate clustering results. This algorithm has the capability to handle the random shaped cluster but it has high running time. rDenStream is an improvement over DenStream which applies three phase clustering [20,21]

4.1 Den Stream

Den Stream [22] is for continuously increasing data set that also has the ability to handle the noise. The micro cluster concept was extended in the algorithm which is known as core micro cluster, potential micro cluster and outlier micro cluster to differentiate between real data and outlier. The purpose of core micro clustering is to summarize the arbitrary shape clusters in the data stream. Outlier as well as potential clusters are kept in the separate memory because they are required different processing. Den stream consists of online and offline components. In online phase it keeps real data and deletes micro cluster with noise then density based clustering is performed on real data. It uses fading data model in which weight of data point exponentially decreases with time t through the fading function f(t)= $2f(t) = 2^{-\lambda t}$ $f(t) = 2^{-\lambda t}$ here λ is greater than zero. If $= \sum_{j=1}^{n} f(t - T_{ij})$, where w is the weight greater than a threshold input parameter μ then the corresponding cluster is taken as core micro cluster where Ti1 ------Tin are timestamps of data points Pi1 ------Pin. If at the time t, if $w \geq \beta\mu$ then the micro cluster is called potential cluster otherwise it will be considered as outlier micro cluster where β is outlier threshold relative to core micro clusters ($0 < \beta < 1$). The micro cluster that are not getting recent point trend loses their weight continuously known as outdated micro cluster. The weight of the micro cluster is periodically calculated and based on the weight threshold the decision is taken to keep or remove it.

Whenever a new data point is received the method tries to insert it in nearest potential micro cluster depending on the new radius. If the insertion does not succeed then method tries to add data point in nearest outlier micro cluster. The cluster summary will be updated on successful addition else a new outlier micro cluster is created for that data point. Den stream supports pruning method in which weight of outlier clusters is frequently checked in outlier buffer to find the real outlier. The limitation of Den Stream clustering is non release of allocated memory whenever micro cluster is deleted or two micro clusters are merged as well as pruning phase for outlier removal is time consuming. In offline phase the output of online phase that is potential micro cluster are taken as pseudo point and given to variants of DBSCAN to find the ultimate cluster.

4.2 SOStream

SOStream [23] uses the principle of DBSCAN and Self Organizing Map (SOM) in which winner has effect on immediate neighborhood. It solves the problem of manually choosing the threshold value that has effect on making cluster unstable. It dynamically learns the threshold value for every cluster with the idea of having minimum number of clusters. A micro cluster set also represents the SOStream where for every cluster a cluster feature vector is stored given by a tuple $N_i = (n_i, r_i, c_i)$. Here ni implies data points in N_i . r_i denotes radius of cluster and C_i is the centroid. Whenever a new data point is received the nearest cluster is taken depending on the Euclidian distance between existing micro clusters and absorbed if found threshold is less than dynamically taken threshold. The centroids of cluster that is close to winning cluster is modified to reach closer to winning cluster centroids. It efficiently merges similar cluster and separates the different clusters. Changing, merging, dynamically adopting the threshold for every cluster is performed online method. Merging of clusters is performed whenever they overlap

with less than merge distance and maximum distance between cluster is taken as radius of final cluster to avoid loss of any data point. The exponential fading function is used by the SOStream to handle the non relevant data [24].

4.3 Support Vector Based Stream Clustering (SVStream)

It is support vector and support domain based clustering algorithm in which data space points are mapped in to high dimensional feature space with help of Gaussian kernel. A smallest sphere that encircle data image is selected in feature vector and mapped back to data space. There it builds contours which encircle the data point. These contours are selected as boundaries.

A one class classifier called Support vector domain description is one class classifier stimulated by support vector clustering. The main idea is application of kernel for projecting data in to feature spaces and after that calculate the sphere encircling the whole data excluding outliers. When volume of hyper sphere is reduced sufficiently, there are chances of rejection of some parts of training data point. The main limitation of SVDD is that resultant description is extremely sensitive to the selection of the trade-off parameter and also hard to guess practically. It affects outlier finding performance as well as its common properties radically [25].

Let us suppose group of M data points, q be the Gaussian kernel parameter and C be the trade-off parameter then the sphere structure S is given as

$S = \{SV, BSV, \| \mu\|2, RSV, RBSV\}$.

Where,

- SV implies support vector set.
- BSV implies bounded support vector set.
- $\| \mu\|^2$ implies squared length of the sphere center μ.
- RSV is sphere radius.
- RBSV implies max Euclidean distance of the bounded support vectors from the Center of sphere μ.

5. Grid based Stream Clustering

In grid based clustering data space is quantized in to definite number of cells that creates a grid shape and clustering is performed on these grids. In these clustering algorithms unknown number of data records in a data streams are mapped in to definite number of grids. For example D-Stream is a grid based algorithm which uses two phase scheme that consist of two components that are online and offline components. This clustering technique maps all input data record into a lattice/grid then density of grid is calculated and grids are clustered based on these density. This algorithm espouses a density decaying technique to detain the vibrant modifications in a data stream [26].

5.1 D Stream

D Stream algorithm is being used to cluster streaming data in real time. The algorithm is similar to density based algorithm in many ways but main difference at conceptual level is that clusters are generated using grids at place of micro cluster. Grid is similar to radius constrained micro cluster in Den Stream. So in Den Stream grid density constantly changes with time but it is not necessary to update decay based statistics either in micro cluster or in grid at each time instant. This is due to all

grids decay at similar proportional rate and lazy approach is used to perform updates. Updates are done only when grid density value changes when new data points are added. It consists of online and offline components. The job of online component is to process input stream and provide the summary statistics, offline part uses this summary statistics as input and produces the clusters. The online part maps the given data points in to grids then offline component computes the density of grid and clusters the grid relying on density. It finds the clusters of arbitrary shape and automatically adjusts the clusters without providing the specification of target time horizon as well as quantity of clusters. Density of grid g at specific time t is given as $D(g,t)=\sum_{x[E(g,t)} D(x,t)$

Where D (g,t) is sum of coefficient of whole data points mapped to g. E(g.t) is group of data points mapped to g at or before time t.

The main limitation of the approach is that significant number of grids to be discarded to keep the memory needs on track. The cluster quality reduces on discarding of these cells.

5.2 MR Stream

This algorithm [27] enhances the feat of density-based clustering algorithm by running the offline part at steady times. The algorithm finds out the accurate time for the users to create the clusters. MR-Stream divides the data space in to cells and a tree data structure that maintains the space division. Every time a dimension is divided in two and a cell can again be partitioned in to 2d parts where d is the dimensionality of dataset. In tree data structure every node contains the summary information about its successor and descendent. MR-Stream has two phases online and offline. As the data point arrives in the online phase, it is assigned to related cell of the grid. The offline phase creates the cluster of height defined by user. It finds all the accessible dense cells at a particular distance and marks as cluster. The clusters with noise are deleted on the basis of their size, density and density thresholds respectively. MR-Stream launches a memory sampling technique to decide when offline component to run that feat the cluster performance.

6. Discussion

In addition to algorithms mentioned in this paper, applying grid-based and density-based techniques, some hybrid algorithms have been developed by researchers for data streams known as density grid-based clustering algorithms [28,29]. These algorithms divides the data space is into tiny parts called grids. Each data point of data streams is assigned to a grid and after it grids are partitioned lying on their density. Density grid-based algorithms can find random shape clusters; it can find the outliers as well as it has fast processing time that is only dependent on the number of cells.

There are some other algorithms to cluster data streams like TECHNO STREAMS [2] based on artificial immune system that can find out unknown number of clusters including noise. Some authors applied called Flocking model to partition the stream bio inspired model. FADS in which multi agent algorithm used to find variance in data stream was proposed by A. Forestiero [30]. All the methods given in this paper advantages as well as some limitations given in following Table 2. These algorithms perform the clustering process by focusing on various aspects. Most of them consider

handing of noise and outliers. Some of them vary in their time complexity. Few algorithms consider whole data stream and few summary of data stream. So there is a need to develop such algorithm which consists of all the necessary features like creation of quality cluster, less time complexity, noise detection and concept drift [31, 32,33]. These big data clustering algorithms can be used for generating recommendations to meet the challenges of knowing customer emotional intelligence, understanding and optimizing business processes, financial trading etc.

Some other critical issues of big data stream clustering are as follows

- It is required to design some efficient tools for data streams since traditional systems are not capable of dealing with it.
- Designed algorithms will be used in wireless environment on mobile devices. So it must be energy efficient. For example it will be used in sensor networks that have very short battery life.
- There is need to design space efficient methods that needs only one look at incoming data stream.
- Designed space and time efficient schemes must produce acceptable results.
- After extracting knowledge and patterns locally. It is essential to transfer it to user. Therefore it must be possible this knowledge and patterns over the limited bandwidth channel.
- Some light weight preprocessing tools must be designed to provide better results and can be integrated with data mining techniques to automate the process.
- Some technological issues must be solved like tiny devices were not made for complex data computation. Now a day's emulators are used to perform this task. Therefore some hardware solutions are needed to deal with it.

Table 2 *Advantages/Disadvantages*

Method Name	Advantages	Limitations
Partitioning based Methods	• Easy Implementation • Iterative method applied to generate clusters.	• Necessary to define number of clusters by user. • Finds only spherical shape clusters.
Hierarchical based Methods	• Easy to find similarity and distance metric	• Uncertainty in termination criteria. • Complexity is more.
Density based Methods	• It can find arbitrary shape clusters • It can also handle noise	• Many parameter are needed to be predefined. • Performance reduces with multidensity data
Grid based Methods	• Quick processing time. • It can easily handle noise.	• Not efficient for high dimensional data. • Grid size should be known

7. Conclusions and Future Work

Today the data is generating in huge velocity in different area though sensors, digital cameras, Internet of Things, black box etc. This data in form of streams is also huge in size so needs to be processed in real time without storage. Therefore data stream

mining has emerged as an important and major research area. Various data stream mining algorithms have been proposed. This paper categorizes stream mining algorithms in four categories that are Partition based Clustering, Hierarchical Stream Clustering, Density based Stream Clustering and Grid based Stream Clustering. It emphasizes mainly on partioning based clustering and reviewed them. This paper also provides some introduction to remaining three categories. The summarized data stream clustering algorithms will be useful for researchers and practitioners working in the fields of big data stream clustering. They can go through the summary of clustering algorithms and can propose new efficient algorithms.

For future work it is aimed to develop efficient data stream clustering algorithm to provide the quality cluster in terms of its running time. The affect of high dimensionality reduces the effectiveness and precision of the algorithm. So it is also necessary to develop efficient dimensionality reduction technique. It is also important to develop algorithm which should deal with uncertain data effectively.

8. Acknowledgements

We would like to offer the sincere gratitude to the management and the University for providing the constant encouragement and support provided throughout the period of this research work. I would also like to thank our colleagues and all the people who created such a good atmosphere to complete the work.

9. References

1. Guha S, Meyerson A, Mishra N et al ”Clustering data streams: theory and practice”, in IEEE Transactions on Knowledge and Data Engineering, vol 15, pp 505–52,2003
2. Gianmarco, Albert Bifet et al “IoT Big Data Stream Mining”, in KDD August 13-17, 2016, San Francisco, CA, USA, 2016, DOI: http://dx.doi.org/10.1145/2939672.2945385
3. M. Kantardzic, Data mining: concepts, models, methods, and algorithms: John Wiley & Sons, 2011
4. T. White, Hadoop: The Definitive Guide Third Edition: O'Reilley, 2012
5. M. Minelli, M. Chambers and A. Dhiraj, Big Data, Big Analytics: Emerging Business Intelligence and Analytic Trends for Today's Businesses: Wiley, 2013
6. Dilpreet Singh,Chandan K Reddy "A survey on platforms for big data analytics" in Springer Journal of Big Data Research ,pp 1-20,2014, https://doi.org/10.1186/s40537-014-0008-6
7. C.L. Philip Chen , Chun-Yang Zhang” Data-intensive applications, challenges, techniques and technologies: A survey on Big Data” in the Elsevier journal of Information Sciences pp314–347 , 2014
8. Chun-Wei Tsai, Chin-Feng Lai et al "Big data analytics: a survey"," in Springer Journal of Big Data Research , pp 1-32, 2015, https://doi.org/10.1186/s40537-015-0030-3
9. C. Aggarwal, A Survey of Stream Clustering Algorithms: CRC Press, 2013.
10. O. Nasraoui, C. C. Uribe, C. R. Coronel, and F. Gonzalez, "Tecno-streams: tracking evolving clusters in noisy data streams with a scalable immune system learning model," in Third IEEE International Conference on Data Mining, pp. 235-242,2003.

11. C. C. Aggarwal, "Data Streams, Models and Algorithms," Springer, 2007. http://dx.doi.org/10.1007/978-0-387-47534-9
12. Ghesmoune, M., Lebbah, M. ,"State-of-the-art on clustering data streams" ,in Springer journal of Big Data Analytics vol 1, pp 1-13,2016, https://doi.org/10.1186/s41044-016-0011-3
13. L. O'Callaghan, N. Mishra, et al, "Streaming-data algorithms for high-quality clustering," Proceedings 18th International Conference on Data Engineering, San Jose, CA, 2002, pp. 685-694. doi: 10.1109/ICDE.2002.994785
14. Jonathan A. Silva, Elaine R. Faria, et al "Data stream clustering: A survey", in ACM journal of Computing Surveys. Vol. 46, pp 13-31, 2013
15. Maryam Mousavi1, Azuraliza Abu Bakar et al "Data Stream Clustering Algorithms: A Review", Int. J. Advance Soft Compu. Appl, Vol. 7, No. 3, 2015
16. A. Zhou, F. Cao, W. Qian, and C. Jin, "Tracking clusters in evolving data streams over sliding windows," Knowledge and Information Systems, vol. 15,pp. 181-214, 2008.
17. He, Y., Lee, R., Huai, Y., et al " RCFile: A Fast and Space efficient Data Placement Structure in Map Reduce-based Warehouse Systems." in IEEE International Conference on Data Engineering (ICDE), pp. 1199–1208, 2011
18. Ding, S., Wu, F., Qian, J. et al." Research on data stream clustering algorithms ", in An International Science and Engineering Journal Artificial Intelligence Review published by Springer, Vol 43, pp 593–600, 2013
19. M. R. Ackermann, M. Märtens, et al "StreamKM++: A clustering algorithm for data streams," Journal of Experimental Algorithmics , vol. 17, p. 2.4, 2012
20. MarjanKuchaki Rafsanjani, et al, "A survey of hierarchical clustering algorithms", in the Journal of Mathematics and Computer Science, vol.3, pp.229- 240,2012
21. Tian Zhang, Raghu Ramakrishnan, et al, "BIRCH: an efficient data clustering method for large databases", International Conference on Management of Data, In Proc. of 1ACM-SIGMOD Montreal, Quebec,1996
22. Hofmeyr, David, Pavlidis, et al "Divisive clustering of high dimensional data streams", in Springer journal of Statistics and Computing, Vol. 26, pp 1101–1120, 2016
23. Shifei Ding, Jian Zhang et al," An Adaptive Youn J., Choi J., et al "Partition-Based Clustering with Sliding Windows for Data Streams", in proceedings of 22nd International Conference on Database Systems for Advanced Applications, Suzhou, China, March pp 27-30, 2017
24. Amineh Amini, Teh Ying Wah, et al "On Density-Based Data Streams Clustering Algorithms: A Survey "journal of computer science and technology vol 116, 2014. DOI 10.1007/s11390-013-1416-3
25. Y. Chen, and L. Tu. Density-based clustering for real time stream data, ACMKDD Conference, 2007.
26. H.-P. Kriegel, P. Kroger,et al ". Density based subspace clustering over dynamic data" in SSDBM Conference, 2011.
27. Isaksson C, Dunham MH, et al "SOStream: Self organizing density-based clustering over data stream", In MLDM. Berlin: Springer Berlin Heidelberg; 2012. p. 264–78.

28. Wang C, Lai J, Huang D, Zheng W. SVStream: A support vector-based algorithm for clustering data streams. IEEE Trans Knowl Data Eng, 1410–24,2013
29. Chen Y, Tu L.," Density-based clustering for real-time stream data", In the Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, California, USA, p. 133–142, August 12–15, 2007.
30. A. Forestiero, "FADS: Flocking anomalies in data streams", in 6th IEEE International Conference Intelligent Systems (IS), pp. 461-466,2012
31. Sabeur Aridhia Engelbert Mephu Nguifo" Big Graph Mining: Frameworks and Techniques", in Big Data Research Volume 6, Pages 1-10,2016
32. Michael Hahsler, John Forrest et al. "Introduction to stream: An Extensible Framework for Data Stream Clustering Research with R", in the Journal of Statistical Software, vol. 76), pp 1–50, 2017
33. F. Xia, W. Wang, T. M. Bekele and H. Liu, "Big Scholarly Data: A Survey," in IEEE Transactions on Big Data, vol. 3, no. 1, pp. 18-35, 2017.
34. https://onlinecourses.nptel.ac.in/noc17_ma17/student/home. [Accessed: 25-July- 2017]
35. https://courses.cognitiveclass.ai/dashboard [Accessed: 28- Sept- 2017]

About Our Authors

Hemant Kumar Singh received PhD in Computer Science from Dravidian University Kuppam, Andhra Pradesh. He did MCA & M. Tech. (CSE) degree from U.P. Technical University Lucknow, U.P. India. He also earned M. Phil in computer Science and qualified UGC NET exam in Computer Sc. & Application. He has published 18 research papers in International/National Journal and conferences and attended various short term summer, winter courses and workshops. He attended one week workshop on Pervasive Computing in IIT Roorkee in 2012. He did one week workshop on Big Data & R Technology from DSMNRU Lucknow as well as successfully completed online NPTEL Elite certification in R Technology from IIT Kanpur.

Vinodani Katiyar has more than 16 years of experience in the field of Information Technology. She has rich and diverse experience in academia and has been visiting professor at various universities and colleges. She obtained her doctorate degree in computer science from U.P. Technical University in 2006. Presentaly she is working as head of the department of Information Technology at Dr. Shakuntala Misra National Rehabilitation University, Lucknow. She at present is Additional Proctor and Coordinator for faculty of Engineering and Technology.