A Survey on Clustering Algorithms for Data Streams

Article *in* International Journal of Computer Applications · October 2018

DOI: 10.5120/ijca2018918014

3 authors, including:

Neha Sharma
Shri Govindram Seksaria Institute of Technology and Science
5 PUBLICATIONS **3** CITATIONS

SEE PROFILE

A Survey on Clustering Algorithms for Data Streams

Neha Sharma
Shri Govindram Seksaria Institute of Technology and Science, Indore (M.P) India

Shraddha Masih
School of Computer Science, DAVV, Indore (M.P), India

Pawan Makhija
Shri Govindram Seksaria Institute of Technology and Science, Indore (M.P), India

ABSTRACT
Data stream mining is an emerging area for extracting useful information from continuous arriving data. Web click stream, weather monitoring, network traffic, shopping history, web log are some key resources of generating data stream. Clustering is one of the most useful technique for analsing stream data, as it does not require any predefined class labeling. Data stream mining is challanging as the data is massive and arriving continuously. The traditional clustering algorithms cannot be directly applied on the data streams. Data stream mining needs one scan algorithms to extract rich data in the form of data streams. In this paper we discuss various data stream clustering algorithms with their limitations and required data structures. This paper also provides a comparative study of these algorithms. Real world applications of data streams, data resources and publicly available softwares are also discussed.

Keywords
Data Mining, Data Stream, Clustering

1. INTRODUCTION
Recent advances, in hardware and software generates lot of continueous, unbounded, hetrogeneous data. These data is known as data stream. Extracting useful knowledge from data stream is quite challenging [1]. Traditional data mining techniques (Clustering, Classification, Pattern Identification), can not be applied directly on data stream, because it is not possible to scan data multiple times. Clustering is important in data stream mining because it is based on unsupervised learning. Existing clustering techniques work on static data and needs multiple scanning of data, hence modified clustering techniques are proposed by many researhers[2,3]. This paper presents various algorithms for clustering data streams. In our survey, we discuss details of algorithms and data structure for stream data.

2. DATA STREAM CLUSTERING METHODS
Data stream clustering methods are mainly classified into five categories : hierarchical methods, partitioning methods, density based methods, grid based methods and model based methods.

2.1 Hierarchical methods
Hierarchical methods are divided into two types namely- agglomerative and divisive. In agglomerative methods separate clusters are merged up until the distance between the two cluster reaches to the minimum required. In divisive a big cluster is divided into small clusters until the cluster cannot be split further. Some hierarchical algorithms are BIRCH, CURE, ROCK, ODAC, E-Stream, HUE-Stream.

2.1.1. BIRCH
BIRCH [4] was designed to mine large offline data. BIRCH can also be used for stream mining because of its growing nature. It uses two types of data structure named clustering feature vector (CF) and height balance tree (B+ tree) named CF tree. CF has mainly three component - number of data points(N), linear sum of data points(LS) and square sum of data points(SS). LS and SS are n-dimensional arrays. Clusters centroid, radius and diameter can be calculated using these three components. In CF tree each non leaf node contains a CF vector and a pointer to a child node. Every child node of a CF tree is a CF vector. Algorithm work in two steps: in first step data is scanned to build CF tree. Each leaf node has a maximum radius or diameter defined by user. The height of the tree is function of the diameter. In the second step, when the new data point arrives, the algorithm checks the tree from root to leaf. At each non leaf its closet CF vector is chosen. Either the new data point is absorbed by choosing CF vector or the new CF entry is created. BIRCH has a major drawback of its limited capacity of leaves, as it needs predefined size of tree by user.

2.1.2. CURE
In Clustering Using Representative (CURE) [5] algorithm, initially any constant number (C) from well scattered point are chosen of a cluster. These chosen scattered points gets shrink towards centroid of the cluster by the fraction alpha (a). These shrank scattered point then becomes representative of the cluster. The distance between the two clusters is measured as the distance between the closest pair of representative points. CURE uses sample data for clustering. The size of the cluster is predecided. Chernoff bound is used to define the boundary of clusters. CURE is more reliable and robust to outlier and easily identify clusters of non-spherical shape.

2.1.3. ODAC
Online divisive agglomerative clustering (ODAC) [6] is a clustering technique for time series data stream. Cluster are generated in the hierarchy of tree-shaped structure. These trees are constructed on top down manner. The leaf of the tree shows the cluster with the set of variables. This algorithm can handle concept drift. Closeness between time series is estimated by using Pearson correlation coefficient. The splitting of the cluster is performed when the diameter of the cluster crosses Hoeffding bound. Testing for splitting is performed when predecided data points arrive in clusters.

2.1.4. E-Stream
E-Stream is evaluation based technique [7] that has five points: appearance, self evolution, merge, split and disappearance. Initially the data point is considered as isolated cluster. Next upcoming data either join the existing cluster or form new cluster. Each cluster is represented by FCH (fading cluster histogram). The clusters can be merged if they show similar behaviour. Cluster distance is calculated between the center of the cluster and upcoming data point and then the merging is performed. The cluster can be split

into two clusters if it shows distinct behaviour. This behaviour is measured on the basis of radius_factor. Only the active clusters are split. Alpha-bin Histogram is used to summarize the cluster. The range of each bin has been calculated. When maximum and minimum value changes, new range is calculated and splitting is performed.

range=(Maximum feature value + mininmum feature value)/alpha

With time, cluster get faded and disappears. Fading function f(t) is used to confirm the disappearance of the cluster. fading function is calculated as :

$$f(t)= 2^{-\lambda \cdot t}$$

where Lambda is the decay rate and t is the elapsed time.

2.1.5. HUE-Stream

HUE-Stream [8] algorithm is an extended version of E-Stream. The uncertainty present in the heterogeneous data stream is handled by this algorithm. Uncertainty is handled by the distance function with probability distribution of two data points. It also merges and splited data like E-Stream. Histogram management is used to split the data. Maximum no of clusters are predefined here. In HUE-Stream the distance function is modified so that it can handle hetrogeneous data.

Distance between cluster and data point is calculated as:

Dist(C, Xi) = (∂) * numerical_dist(C, Xi) + (1- ∂) * categorical_dist(C, Xi)

Distance between two clusters is calculated as

Dist(C1, C2) = 1\n * ∑ | (center_C1 – center_C2) |.

2.2. Partition Based Methods

An algorithm based on partition method tries to find out k-partitions based on some measurements. K-Means and K-Meadins based methods come into this category. Stream LSearch, Incremental K- Means, CluStream, HPStream, SWClustering, STREAMKM++ are some partition based clustering algorithms.

2.2.1. Stream Lsearch

In Stream Lsearch [33], the data stream is considered in the form of chunks. The size of the chunks is such that it will fit into memory. The cluster is formed using LSEARCH algorithm. LSEARCH is based on the K-Median algorithm. CG algorithm [33] is used to define facility cost z of data points as initially solution (I, F) is derived. Here I belongs to N (facility location) and F is assignment function. Binary search is applied on z to find desire k to form clusters.

2.2.2. Incremental K-Means

There are too many variants of K-Means named as Scalable K-Means, online K-Means, Incremental K-Means [9]. Incremental K-Means algorithm is used to create clusters of binary data stream. The iteration is performed only when the convergence is found. Scalable K-Means and standard K-Means algorithms are slower than Incremental k- means. Sparse distance computation is used to enhance K-Means. After recieving the transaction, the distance between the non-null dimension is computed. K-dimensional vector is defined to calculate distance between center of cluster and null vector. Cluster centroids are not random seeds but global mean of data set. This centroid is incrementally maintained.

2.2.3. CluStream

CluStream [10] forms cluster in two steps: the first step is online or micro cluster. In micro cluster the detailed summary of data-statistics is stored. These are also defined as temporal extension of cluster feature vector. Microclusters are stored in main memory in the form of snapshot. These snapshot is taken using pyramidal time frame. For each dimension, square is stored in CF2 and sum is stored in CF1. CF3 and CF4 are used to the stores the sum and square-sum of time stamp.

Microcluster = (CF1, CF2, CF3, CF4,n) where n is number of transactions.

Each new upcoming data point is either appended in existing cluster or a new cluster is created. The second step is offline or macro cluster. It uses data summary generated by micro cluster. Weighted K-Means algorithm is applied to create macro clusters. CluStream uses fading cluster structure for removing the older microclusters.

2.2.4. HPStream

This algorithm is an extension of CluStream [11] . Here, in the name of the algorithm H stand for high dimensional and p stand for projection based clustering of the data stream. It uses a fading cluster data structure, so it is incremental updatable. Each data point has fading function f(t) associated. The value of f(t) is between [0,1]. The value of fading function decreases by decay rate with time t. Decay rate is defined as lambda = 1\t0, where t0 is half life of a data point. It is also highly scalable. It is better than CluStream if the data is of high dimensionality. Fading function, decay rate and range are same as CluStream.

2.2.5. SWClustering

In this algorithm, a new data structure is proposed [12]. They combine the exponential histogram with temporal cluster features. Exponential histogram is used for evaluation of cluster. The temporal cluster feature is used to distribute data points in clusters. By combining these two, the new data structure is proposed name as Exponential Histogram of Cluster Feature (EHCF). This algorithm uses sliding window model so it is named as SWClustering. Each bucket in the EHCF is temporal cluster feature (TCF) for data sets. TCF is a temporal extension of the cluster feature vector. TCF for d dimensional data is given by (CF2, CF1, t, n) where n is the number of records, CF2 is the square sum of n records, CF1 is a linear sum of n records and t is time stamped. EHCF is the collection of TCF's records. Center of EHCF is calculated by mean of TCF's. When the request of cluster arrives. The cluster is generated by the synopsis of EHCF.

2.2.6. STREAMKM++

STREAMKmeans++ algorithm [13] is also variation of K-Means algorithm. The main task is to develop a new coreset for eulicidean K-Means++, which is suitable to high dimensional data. Coreset tree data structure is used for sampling data points. Coreset is a small set p. Its clustering cost is approximate same as forming set of k-cluster centers. Approximation algorithm can be applied quickly on such sets. Standard Merge and reduce technique is used for sampling. After that Kmeans++ algorithm is used to generate clusters.

2.3 Density Based Methods

Data is separated into nonoverlapping cells also known as density connected regions. These regions are may be of different shapes and sizes. These methods can handle noise and requires only one pass to scan the data. They do not require the pre decided no of clusters. Density threshold is used to determine the connected regions. Some important density based clustering algorithms are – Incremental-DBSCAN, LDBSCAN, DenStream, rDenStream, D-Stream, MR-Stream, DSCLUE, OPClueStream.

2.3.1 Incremental-DBSCAN

DBSCAN[14] GDBSCAN[15] , OPTICS[16] are some density based algorithms which are useful to detect any shape cluster. These all are not suitable for stream mining because of multiple scanning of data is required. Incremental-DBSCAN is an extension of DBSCAN which is suitable for data stream mining. This algorithm [17] makes it possible to update data in the data warehouse. The experiment shows that it has similar results as DBSCAN on to updatable manner. At time of insertion and deletion of a new data point, the definition of desity reachable and boundry object get affected. Boundry object may become noise. New density connection may be established. Cluster membership can be changed by applying such changes on all the data points. Removing any data point may destroy some density connection but no new connection can be established. In such case some data points may get disconnected.

2.3.2 DenStream

This algorithm [18] work in two phases as CluStream. In first phase, online micro cluster is managed. Micro cluster are synopsis of data points. This uses the damped window model. The second phase uses the synopsis created by first phase. This algorithm can identify clusters of any shape. The online dense micro cluster also known as core-micro cluster is designed to manage a cluster of any shape. An outlier buffer is introduced to separate the outlier with core-micro cluster. It uses a fading function $f(t)= 2^{-\lambda \cdot t}$ where $\lambda>0$. Each data point is associated with some weight. Weight of data point is reduced with time using the fading function. Whenever a new data point arrives, it is clustered by using DBSCAN. Then we create a summary of these clusters called core-micro clusters.

2.3.3 *rDenStream*

This algorithm [19] is extended DenStream. It has three phases. Its initial two phases are comparable to denstream, but it has a new phase, called outlier retrospect. Here, rDenStream means DenStream with retrospect. In this phase, the discarded data are stored in memory. This discarded data get the chance to increase its weight and form cluster in the future. This is better than DenStream, but needs more time and memory. Initially size s of window is decided. The upcoming stream of size s is divided into two type of cluster, potential micro cluster and outlier micro cluster. This is same as DenStream. In DenStream outlier micro clusters are discarded while in rDenStream these are stored in historical buffer. In retrospect phase these outlier micro clusters are reidentified. If some cluster is misjudged then retrospect phase provide an opportunity to form potential micro clusters. If the depth of the outlier micro cluster is more than the threshold, then they are discarded.

2.3.4 D-Stream

In D-Stream [20] algorithm, the grid of a data point is generated online. After that, the density of this grid is found out in an offline manner. The clusters are generated on the basis of the density. A decay factor is used to identify the importance of data points. The result shows that it is better than CluStream. Stream data are considering of d-dimension, where each data point is stored in multidimensional gird in space s. Each data point x, is assigned a density coefficient D(x, t). if data point x arrives at time t_c then its time stamp $T(x)=t_c$. Then Density coefficient at time t is given as:

$$D(x,t) = \lambda^{t-T(x)}$$

The density of a grid g at time t is defined as:

$$D(g, t) = \text{sigma } D(x, t)$$

The characteristic vector of a grid is defined as a tuple (t_g, t_m, C, D, label) where t_g is time of last grid update, t_m is time of grid removal , C is a 2d vector that denotes the attraction between grids, D is the grid density and label is a grid class label. On the basis of density characteristic vector clusters are generated in an offline manner.

2.3.5 MR-Stream

In MR-Stream algorithm[21], The data stream is considered in the form of n dimensions. This data is stored in vector form. Tree like structure is used to store the data. Space is divided into cells. Further cell is divided into subcells. The division is limited by the user defined parameter. The divided cell is stored in the quad tree structure. At every time stamp tree pruning is performed. Some task is also performed in offline mode. Offline mode accesses a portion of the tree and perform clustering. It also removes clusters having noise. The height of a tree is limited by the user defined parameter H and each parameter of the data stream is divided into 2^H interval. The root of a tree contains overall synopsis information of space s, while each level $h < H$ of the tree contains synopsis information at granularity level h. Each data point x is assigned a weight value with time t. This weight value w(x,t) decreases with time. It is denoted by fading function $f(t) = \lambda^{-at}$, with $\lambda > 1$ and $a > 0$. Weight of a cell is the sum of all weight values of data point in one cell. Now the cell density D is computed by ratio of cell weight to cell volume. When the requirement of clusters occurs, a portion of a tree is chosen and clusters are formed on the basis of its density.

2.3.6 DSCLUE

DSCLUE [22] is derived by DenStream algorithm. It work in both online and offline modes. Online mode divides the data stream into dense micro clusters (DMC). DMC is defined as (CF1, CF2, W, T_u) where W at time t is $w = \text{sum } f(t - T_i) = \text{sum } 2^{-\lambda(t-T_i)}$ where Ti is arrival time. CF1 is a weighted sum of each data point p in DMC, defined as $CF1 = \text{sum } 2^{-\lambda(t-T_i)} p_j$. CF2 is weighted sum of square of each data point, $CF2 = \text{sum } 2^{-\lambda(t-T_i)} p_j 2$. T_u is last update time of DMC. When a new data point enters in the system, it tries to become the part of a dense micro cluster. For each cluster, time index ($T_{current} - t_u$) is calculated. If time index crosses predefined threshold, then the clusters are updated. The center of the cluster is calculated as CF1 / w. Two types of neighbour , generic neighbour Ng and special neighbour Ns are defined for DMC. Where Ns is the nearest cluster from DMC and Ng is furthest cluster in range r (radius). The offline mode is similar to DBSCAN. It merges the microclusters into more dense clusters. This mode is activated as per the requirement of the user.

2.3.7 OPClueStream

Order Points to CLustering data Stream algorithm [23] is

able to detect arbitrary shape clusters and overlapping clusters. There are mainly two parameters : Eps (Epsilon neighbours) and MinPts (minimum number of points). On the basis of these two parameters data points are divided into three groups : core point group, border point group and noises. A point p belongs to the core point group if MinPts > Eps. Point p belongs to the border point group if MinPts > Eps and at least one of its Eps is core point. If none of its Eps belongs to core point, then it is considered as noise. Two data points p and q are direct density reachable if p is Eps of q and q is core point. If both p and q are density reachable to any point o and o is a core point then p and q are density connected. It uses tree topology to link data points. This tree topology records the relationship between data points. In tree point p is father of point q if q is directly core reachable to p. Each tree farm on this basis is treated as separated cluster. When the new data points arrive it only affects to the limited area of a tree.

2.4 Grid Based Methods

It is a type of density based method in which grid structure is used to identify the density of a data point. The data is divided into cells. Number of data points in each cell define its density. Then the dense cell is aggregated into clusters. CLIQUE, WaveCluster, STING, GCHDS, DGClust are some algorithms that come into this category.

2.4.1 CLIQUE

This algorithm [24] is able to find clusters in high dimensional data. The clusters are generated in the form of DNF expressions. In this, the data points are divided into predefined grid cells. The tree structure is used to store the data.

2.4.2 WaveCluster

Here [25] the signal processing technique is applied on the data. This algorithm is based on the fact that the n-dimensional data can be stored in feature vector. High frequency part of data form the boundary of cluster. While the low frequency part of data forms the clusters. The first step of algorithm is quantization in which d- dimensional feature space data is divided into M interval. Then the data point is assigned into M on the basis of feature space. These quantized data Mj is transformed into discrete wavelet Tk. Connected component of Tk is grouped into clusters. The wavelet transform is created using the formula

$$S_i = \sum c_k \, S_{i+k \; - M\backslash 2}$$

where c_k is the signal coefficient, M is the number of coefficients in the filter and S is the result of convolution.

2.4.3 STING

STatistical Information Grid based [26] method is a region oriented cluster formation method. The whole area is divided into rectangular cells. Each cell's statistical information is calculated. This information is saved and utilized to answer the query. CLIQUE, WaveCluster and STING is not suitable for stream data. GCHDS, GSCDS and DGClust are used for clustering the data stream.

2.4.4 GCHDS

GCHDS [27] is the grid based clustering algorithm. It uses high dimensional data stream. They maintain grid structure in memory of a data stream. They extended the wavecluster. By analyzing the data distribution the high dimensional grid is used to form the clusters. This algorithm is suitable for stream data as it takes very less time to form grid structure. The data points in data streams are divided into chunks which will fit into memory. Minimum L_{min} and maximum L_{max} of data point in each direction is found out. Each dimension is divided into k intervals also called as cell C_i. Any data point x will be a member of C_i, if $L_{max} < C_i < h_i$. Only limited cells are stored in main memory.

2.4.5 GSCDS

GCHDS [28] forms cluster only in one dimension while the cluster may belong to any dimension. To overcome this problem, a new algorithm GSCDS is proposed. This is a grid based subspace clustering algorithm for the data stream. It handles high dimensional data. On some dimensions the data can not be differentiated. So the dimensions are divided into two types: spike dimension and smooth dimension. In spike dimension all data points have almost same value for such dimension. While in smooth dimension, the data points are uniformly distributed. Firstly, find out Ni of each dimension where Ni is the maximum of any data point on dimension i. Then arrange these Ni into descending order. After getting these dimensions clusters are formed. The grid data structure is used to form a synopsis of data. Top down approach is used to find subspace. Then for detecting cluster in subspace, a bottom up approach is applied.

2.4.6 DGClust

DGClust is a distributed grid based clustering algorithm [29]. This is mainly proposed for the data generated by the sensor network. By applying these algorithms every local sensor can store online discretization of the generated data. These data are clustered at a fixed interval. Every new upcoming data point triggers the grid formation.

2.5 Model Based Methods

In such methods a hypothesized model is run for every cluster and check which data will perfectly fit into the model. COBWEB, CluDistream, SWEM are some algorithms of this category.

2.5.1 COBWEB

COBWEB [30] is an incremental conceptual based method to form clusters. It is not designed for stream data, but its concept is used in stream data mining. It uses hierarchical sorting to form clusters. It generates classification tree by using the category function. Each node has a probabilistic description of a data point. It has the limitation of capacity of leaves.

2.5.2 CluDistream

CluDistream algorithm [31] is based on Expectation maximization and work well when the data is noisy or missing. This is for distributed data stream. It deals only when the problem is in landmark window. It combines Gaussian mixture model with test and cluster algorithm.

2.5.3. SWEM

SWEM [32] also uses expectation maximization but it uses time based sliding window. Its work in two parts. In the first part the synopsis of a data is generated in the form of microcomponent. Splitting and merging technique is used to form the cluster in a second step. It is able to handle noise and missing data. Table 1 below showing the comparative study of all the explained algorithms.

Table 1. Comparative study of algorithms

Algorithm	Data structure	Offline Clustering technique	Cluster shape	Outlier detection
BIRCH	feature vector and CF tree	k-means	hyper-shape	Density based
CURE	-	Hierarcical	non-spherical	less sensitive
ODAC	corelation matrix	Hierarcical	Hyper-elipsis	-
E-Stream	fading cluster strucure with histogram			-
HUE-Stream	fading cluster strucure with histogram			
Stream Lsearch	prototype array	k-median	hyper-sphere	
Incremental k- means	feature vector	k-means	hyper-sphere	
CluStream	feature vector	k-means	hyper-sphere	Statistical based
HPStream				
SWClustering	feature vector	k-means	hyper-sphere	-
STREAMKM++	Coreset tree	k-means	hyper-sphere	-
Incremental-DBSCAN	feature vector	k-means	hyper-sphere	
DenStream	feature vector	DBSCAN	arbitrary	density based
rDenStream	feature vector	DBSCAN	arbitrary	density based
D-Stream	grid	DBSCAN	arbitrary	density based
MR-Stream	feature vector	DBSCAN	arbitrary	density based
DSCLUE	grid	k-means	hyper-sphere	density based
OPClueStream	grid	DBSCAN	arbitrary and overlapping	density based
CLIQUE	array and tree	-	-	-
WaveCluster	feature vector	-	arbitrary	insensitive
STING	grid	DBSCAN	arbitrary	density based
GCHDS	incremental grid		arbitrary	density based
GSCDS	incremental grid		arbitrary	density based
DGClust	grid	k-means	hyper-sphere	
COBWEB	tree			
CluDistream		expectation maximization		
SWEM		expectation maximization		

3. REAL TIME DATA STREAM FRAMEWORKS AND TOOLS

3.1 Apache Spark

Apache Spark [34] is open source in-memory cluster computing framework. The spark is able to handle streaming. It takes data in small batches and perform transformation on it. It is useful when you don't have mapreduce installed or you want to work online.

3.2 Apache Flume

The flume is designed to transfer log data, but now it is used to transfer stream data. It has three pluggable components - Source, sink and channel. Source and sink, connect to outer systems while channel transfers data.

3.3 Apache Storm

Storm can be used for data stream computing. It can work with any programming language. It can also be used with Apache YARN and Flume. It works in a distributed manner with real time data

3.4 Apache Samza

Samza is a framework for mining stream data in a distributed manner. It is used for snapshoting the stream data. It stores states of data. This is useful when any one system fails. The data immediately transfer to another system.

3.5 Scribe

Scribe is used to aggregate streaming log data. Every local system runs scribe. These local scribes sends aggregated data to central scribe. If central scribe is not available, then the data are sent to local disk.

3.6 S4

S4 is a platform for processing stream data. It is decentralized, event driven, distributed and scalable platform. It contains Processing Elements(PE's) for computation. Data between PE's is sent in the form of event.

3.7 Amazon Kinesis

Kinesis is provided by Amazon for processing real time stream data on the cloud. Kinesis includes its own library named Kinesis client library(KCL). It strongly integrated with other amazon's services like S3, dynamoDB.

3.8 All-RiTE

All-RiTE is used for update of live data warehouse (DW). It provides a middle storage so that the live data on DW can be processed. It can integrate with other ETL tools.

3.9 R

R is programming language as well as a framework for statistical computing. R support almost all programming languages. It supports various statistical computing

techniques, data visualization techniques and analytics algorithms. Some enterprises also provide a stream processing tool like IBM InfoSphere Streams, Microsoft StreamInsight and Informatica Vibe Data Stream.

3.10 MOA

Massive Online Analysis (MOA) is a real time stream data processing framework. It supports classification, clustering, pattern recognition, regression and graph mining. It works on JAVA.

4. CONCLUSION

Data stream mining is a crucial task because of continuous arrival of data. The conventional clustering method is not useful in data stream mining because stream data needs one time scanning of the data. In this paper, we have discussed a variety of clustering algorithms for data stream mining. All the algorithms are useful with respect to its applications. We also discussed various online tool for mining stream data. A comparative study between various algorithm is also performed.

5. REFERENCES

[1] Gaber, Mohamed Medhat, Arkady Zaslavsky, and Shonali Krishnaswamy. "Mining data streams: a review." *ACM Sigmod Record* 34.2 (2005): 18-26.

[2] Mahdiraji, Alireza Rezaei. "Clustering data stream: A survey of algorithms." International Journal of Knowledge-based and Intelligent Engineering Systems 13.2 (2009): 39-44.

[3] Kavitha, V., and M. Punithavalli. "Clustering time series data stream-a literature survey." arXiv preprint arXiv:1005.4270 (2010).

[4] Zhang, Tian, Raghu Ramakrishnan, and Miron Livny. "BIRCH: an efficient data clustering method for very large databases." ACM Sigmod Record. Vol. 25. No. 2. ACM, 1996.

[5] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "CURE: an efficient clustering algorithm for large databases." ACM Sigmod Record. Vol. 27. No. 2. ACM, 1998.

[6] Rodrigues, Pedro Pereira, Joao Gama, and Joao Pedro Pedroso. "ODAC: Hierarchical clustering of time series data streams." Proceedings of the 2006 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2006.

[7] Udommanetanakit, Komkrit, Thanawin Rakthanmanon, and Kitsana Waiyamai. "E-stream: Evolution-based technique for stream clustering." International Conference on Advanced Data Mining and Applications. Springer, Berlin, Heidelberg, 2007.

[8] Meesuksabai, Wicha, Thanapat Kangkachit, and Kitsana Waiyamai. "Hue-stream: Evolution-based clustering technique for heterogeneous data streams with uncertainty." International Conference on Advanced Data Mining and Applications. Springer, Berlin, Heidelberg, 2011.

[9] Ordonez, Carlos. "Clustering binary data streams with K-means." Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. ACM, 2003.

[10] Aggarwal, Charu C., et al. "A framework for projected clustering of high dimensional data streams." Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. VLDB Endowment, 2004.

[11] Aggarwal, Charu C., et al. "A framework for projected clustering of high dimensional data streams." *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*. VLDB Endowment, 2004.

[12] Zhou, Aoying, et al. "Tracking clusters in evolving data streams over sliding windows." Knowledge and Information Systems 15.2 (2008): 181-214.

[13] Ackermann, Marcel R., et al. "StreamKM++: A clustering algorithm for data streams." Journal of Experimental Algorithmics (JEA) 17 (2012): 2-4.

[14] Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." Kdd. Vol. 96. No. 34. 1996.

[15] Sander, Jörg, et al. "Density-based clustering in spatial databases: The algorithm gdbscan and its applications." Data mining and knowledge discovery 2.2 (1998): 169-194.

[16] Ankerst, Mihael, et al. "OPTICS: ordering points to identify the clustering structure." ACM Sigmod record. Vol. 28. No. 2. ACM, 1999.

[17] Ester, Martin, et al. "Incremental clustering for mining in a data warehousing environment." VLDB. Vol. 98. 1998.

[18] Cao, F., Estert, M., Qian, W., & Zhou, A. (2006, April). Density-based clustering over an evolving data stream with noise. In Proceedings of the 2006 SIAM international conference on data mining (pp. 328-339). Society for Industrial and Applied Mathematics.

[19] Liu, Li-xiong, et al. "A three-step clustering algorithm over an evolving data stream." Intelligent Computing and Intelligent Systems, 2009. ICIS 2009. IEEE International Conference on. Vol. 1. IEEE, 2009.

[20] Tu, Li, and Yixin Chen. "Stream data clustering based on grid density and attraction." ACM Transactions on Knowledge Discovery from Data (TKDD) 3.3 (2009): 12.

[21] Wan, L., Ng, W. K., Dang, X. H., Yu, P. S., & Zhang, K. (2009). Density-based clustering of data streams at multiple resolutions. ACM Transactions on Knowledge discovery from Data (TKDD), 3(3), 14.

[22] Namadchian, Amin, and Gholamreza Esfandani. "DSCLU: a new Data Stream CLUstring algorithm for multi density environments." Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on. IEEE, 2012.

[23] Wang, Huan, et al. "A density-based clustering structure mining algorithm for data streams." Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications. ACM, 2012.

[24] Agrawal, Rakesh, et al. Automatic subspace clustering of high dimensional data for data mining applications. Vol. 27. No. 2. ACM, 1998.

[25] Sheikholeslami, Gholamhosein, Surojit Chatterjee, and Aidong Zhang. "WaveCluster: a wavelet-based clustering

approach for spatial data in very large databases." The VLDB Journal—The International Journal on Very Large Data Bases 8.3-4 (2000): 289-304.

[26] Lu, Yansheng, et al. "A grid-based clustering algorithm for high-dimensional data streams." Advanced Data Mining and Applications. Springer, Berlin, Heidelberg, 2005. 824-831.

[27] Sun, Yufen, and Yansheng Lu. "A grid-based subspace clustering algorithm for high-dimensional data streams." International Conference on Web Information Systems Engineering. Springer, Berlin, Heidelberg, 2006.

[28] Gama, Joao, Pedro Pereira Rodrigues, and Luís Lopes. "Clustering distributed sensor data streams using local processing and reduced communication." Intelligent Data Analysis 15.1 (2011): 3-28.

[29] Gama, Joao, Pedro Pereira Rodrigues, and Luís Lopes. "Clustering distributed sensor data streams using local processing and reduced communication." Intelligent Data Analysis 15.1 (2011): 3-28.

[30] Fisher, Doug. "Iterative optimization and simplification of hierarchical clusterings." Journal of artificial intelligence research 4 (1996): 147-178.

[31] 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.

[32] Dang, Xuan Hong, et al. "Incremental and adaptive clustering stream data over sliding window." International Conference on Database and Expert Systems Applications. Springer, Berlin, Heidelberg, 2009.

[33] Arthur, David, and Sergei Vassilvitskii. "k-means++: The advantages of careful seeding." Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. Society for Industrial and Applied Mathematics, 2007.

[34] Zhang, Pengfei, and Zonghuai Guo. "An Improved Speculative Strategy for Heterogeneous Spark Cluster." *MATEC Web of Conferences*. Vol. 173. EDP Sciences, 2018.