ADV MATH SCI JOURNAL
Advances in Mathematics: Scientific Journal **9** (2020), no.9, 7451–7462
ISSN: 1857-8365 (printed); 1857-8438 (electronic)
https://doi.org/10.37418/amsj.9.9.91

ANALYTICAL REVIEW OF EVOLVING DATA STREAM USING WEIGHTED CLUSTERING

ALAA FAREED ABDULATEEF[1], YUHANIS YUSOF, AND AZMAN YASIN

ABSTRACT. One of the data mining important tasks is clustering, and data stream clustering is a complicated process due to the infinite data arrival. Besides, the rapid growth of ICT, such as the deployment of the Internet of Things (IoT) in various real-world applications, has contributed to the overwhelming data stream. As data stream evolves over time, grouping the data into relevant clusters requires much of an attention. This paper discusses the various clustering algorithms used for the data stream and highlights their strength and weakness. To follow, proposed a new weighted function to cluster the evolving data stream. The main aim is to come up with a new cluster weight function that can enhance the evolving data stream clustering. The new function includes all the essential factors that affect data stream weight; some of these factors been missed by other studies.

1. INTRODUCTION

Lately, many applications facilitate collecting data in streams. Such as everyday life transactions, for example, using credit cards, online shopping, and browsing the Internet leads to generate massive data. Similarly, the flow of data can be mined online to extract meaningful information. When the underlying data is continuously moving over time, which is known as (data stream), it leads to several issues such as handling evolving data stream with the existence

[1]*corresponding author*

2010 *Mathematics Subject Classification.* 97R40.

Key words and phrases. data mining, data clustering, data stream, IoT, data stream weight.

of outliers. Studies on outlier detection and removal have been done extensively in the literature. In this context, it is essential to point out that there are many viewpoints of an outlier. In the area involving data classification, data that is close to the decision boundary or abnormal events is known as outliers. In literature, outlier detection is in trend. Ramasway et al. (2000) used a distance-based method for outlier detection by using on clustering algorithm to partitions the input data stream set into disjoint unique subsets, then to eliminate the outlier the portioning phase followed by pruning entire partitions that confirmed as an outlier cluster. The result is supported in another experiment in which the distance is used to rank data points from the closest neighbors. This study is followed by Breunig et al., who presented the concept of local outlier factor, which is termed as LOF [1].

Even though the above-mentioned studies are successful, they cannot be applied on data stream due to the usage of nearest distance neighbours. This is because the data stream is a long sequence of data and determining the neighbours for a data point is very difficult and subjective. Hence, Duan et al. proposed a new definition for outlier as he presented an algorithm termed as cluster-based outlier detection. This approach can detect both cases the single point outlier as well as a cluster-based outlier. Amini proposed an algorithm that can cluster and, at the same time excluding outliers from the data stream. This is achieved by using a grid-based density cluster as a buffer for isolating outliers from the data which will be formed as a final cluster without outliers. This paper includes an analytical study of clustering algorithms, based on the machine learning approach, focusing on the data stream and comparing the available algorithms that have met the data stream requirements. The findings suggest a weighted clustering to better detect outlier in the data stream.

2. CLUSTERING

In supervised machine learning algorithms, the primary function of the algorithms is to learn how to map the output data from input data. The obtained model will then be used on the untested data set. In contrast, unsupervised learning does not have any guidance, and only input data will feed the algorithm before finding the pattern from the input data. Generally, in an input data structure, there exists a certain pattern that occurs predominantly. From

this pattern, the algorithm then discovers the meaning of the data. In statistical analysis, this is known as density estimation. Clustering is an example of density estimation, where the function is to cluster or group the input. These groups are formed based on similar/nearby points. Figure 1 illustrates the clustering concept.

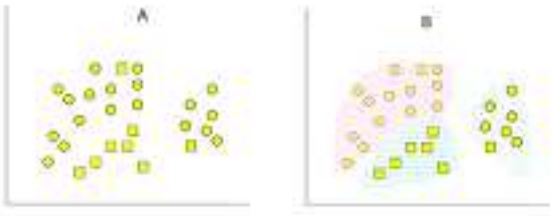

FIGURE 1. A: shows the original dataset before applying clustering, B: shows the clusters generated from dataset A

It should be mentioned at this time that none of the existing data stream clustering algorithms could solve all data stream challenges in a single algorithm such as high-speed data stream processing, one pass processing, dimensionality... etc. Therefore, the choice of the algorithm depends highly on the use case.

2.1. **Partitional Clustering Algorithms.** In partitional algorithms, the input data is divided into a defined number of partitions (non-overlapping subgroups) where each data point is mapped to exactly one specific cluster. Figure 2 illustrates the partitional clustering. Many researchers adopt partitioning algorithms for clustering problems, Yue et al., 2015 used, c-means and fuzzy c-means algorithms for clustering problem and evaluated the clustering quality by developing a new measure, called as the dual centre , this study followed by Jiang et al. 2016 used K-means algorithm for outlier detection, adopting weighted matching distance metric, to calculate the distance between two objects described by categorical attributes. Furthermore, Kumar and Reddy 2017, they improved the K-mean clustering algorithm as a clustering algorithm by proposing an efficient initial seed selection method. Moreover, Zhu and Ma 2018 were proposed a novel partitional based clustering method integrated with the enhanced classical K-means clustering algorithm, and this approach proposed a new variance-based clustering validity index to improve the optimal clustering number. In

addition to that, Chowdhury et al. 2019 used the K-means clustering algorithm to select an optimal cluster seed, which is not an outlier.

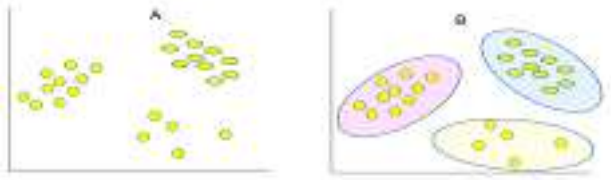

FIGURE 2. Partitional clustering

2.2. **Hierarchical Clustering.** In hierarchical clustering, nested clusters are used to map the data points. The data points will be arranged into a hierarchical tree representing a continuity of likeness and clustering. The main aim of the algorithm is to combine two nearest clusters in one cluster. These algorithms do not need a predefined number of clusters, but in some applications, the algorithm aims to segment the disjoined clusters similar to flat clustering, in this case, the hierarchy tree needs to be cut at some point. Many researchers used hierarchical clustering algorithms such as Bouguettaya et al. (2014) [2] improved the efficiency of the hierarchical clustering algorithm by using KnA method. This method applies clustering on group data points instead of individual data points. Later, Žurauskiene and Yau (2016) developed an agglomerative clustering algorithm (pcaReduce) to generate a cell state hierarchy where each cluster branch is associated with a principal component of variation that can be used to differentiate two cell states. Furthermore, Siless et al. (2017) [3] proposed AnatomiCuts algorithm for unsupervised hierarchical clustering of dMRI tractography data based on anatomical similarity measures and incorporated this measure into a hierarchical clustering algorithm and compare it to a measure that relies on Euclidean distance. Moreover, Cohen-Added et all (2019) proposed novel approach with design "good" fitness functions to determine both cases "similarity" and "dissimilarity" based on hierarchical clustering algorithm and then by analyzing the output performance results to decide which one is the suitable fitness function that bypasses the other algorithms in terms of speed, the following Figure 3 illustrates hierarchical clustering algorithm.

2.3. **Density-Based Clustering.** In density-based clustering, the cluster is formed and separated based on their density, where the low-density regions will

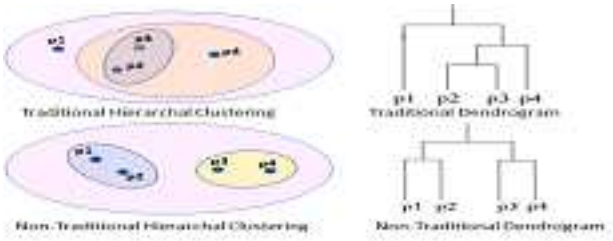

FIGURE 3. Hierarchical clustering

be separated from the high one. In the case of outlier data or abnormal data points, the algorithm will form irregular clusters from it. A cluster is defined as the greatest set of connected dense data as the algorithm can discover clusters of arbitrary shape clusters. Many researchers used density-based clustering algorithms such as Amini et al. 2016 [4] used grid and micro clustering algorithm to summarise the data stream and to prune the outliers, then used multi-density DBSCAN algorithm to form the final clusters. This study, followed by Bryant and Cios in 2017 [5], proposed density-based self-organising clustering for text stream by using a micro cluster method for the online phase and again used micro clustering for the offline phase to produce macro clusters. Figure 4. Illustrates the basic concept of density-based clustering.

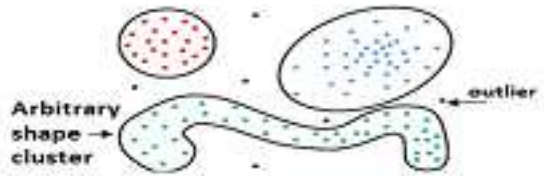

FIGURE 4. Density-Based Clustering

3. REQUIRED CHARACTERISTICS IN DATA STREAM CLUSTERING ALGORITHM

One of the main obstacles to the process data stream is that the data is continuously streaming and evolving over time, and the data cannot be kept for a long time as a new data point is always coming. Such data can be seen in the IoT applications where data is produced by sensors all the time. IoT generates fast and changing over time (evolving) with a huge data stream. Hence, there is a need to have a clustering algorithm that is tailored to operate on such data.

The following points elaborate on the characteristics required of a clustering algorithm to detect outlier:

- ***One pass***: the data processing has to be done with single-pass once, the data has to pass through a process which has to produce the result without having a second pass in clustering a data stream, the data should be processed once only with the assumption that data objects received in streams and been processed with an algorithm that can handle the data stream such as k-means.
- ***Evolving***: in data stream clustering algorithms, one pass methods of the cluster are formed over the entire data stream, but these data are continuously streaming and evolving over time. Hence, the clustering outputs may evolve (change) over time. In some methods that can handle the evolving feature of the data stream, the behaviours of data streams over time are considered as an evolving process and can be handled using different forms of window modelling. Mainly there are three types of window models illustrated in Figure 5.

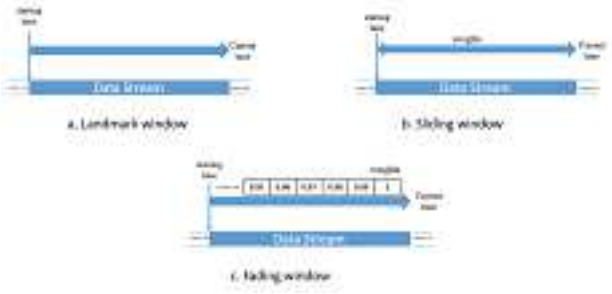

FIGURE 5. Windowing Models a. Landmark Window, b. Sliding Window c. Fading Window

The Landmark window model determines a window by deciding on a specific time interval (known as a landmark) and the present. It is utilized for mining over the whole history of the data streams (Figure 5a). The Sliding window model main idea is to achieve a detailed analysis over the historical data in the form of summarization and the most recent data points which the weight is the size of the window when the weight is too large that means there are outdated information and the accuracy of the model decreases. If the weight is small, the window may

have insufficient data, and the model over-fits and suffers from large variances (Figure 5b). On the other hand, The Fading (Damped) window model (Figure 5c) is usually used with an evolving data stream where each data point will be given weight. The weight is determined based on fading function, where the most recent data will be given larger weights while the weight for older data will decay gradually.

- ***Online-Offline***: the online-offline clustering algorithms are introduced by Aggarwal. The researchers came out with online-offline algorithms that mainly composed of two components; online and offline. The online component aims to summarise the data stream using statistical analysis while the purpose of the offline component is to form the clusters based on the summarization.
- ***High Dimensions***: the algorithm should also be able to operate on high dimension data. In some data stream, the distance between data points become indistinguishable and lose their usefulness, and that nearest and farthest neighbours become more and more similar. The problem is often caused by the increase in irrelevant dimensions, which make the analysis problem harder. To solve this problem for data stream clustering, many solutions can be used, such as subspace projection or shared nearest neighbour similarity. By applying clustering on subspaces, clusters can still be found. However, it is computationally high because all pairwise groupings of dimensions are formed as a cluster.
- ***Limited Time and Memory***: for real-time applications such as cyber-attack detection, the clustering algorithm should utilize less time and memory during the detection. This is because the data will be a nonstop and of a high-speed stream.
- ***Summarisation***: Data stream is a massive infinite data with space and time constraint which affect the computation process, so it is not possible to record and process the whole data. Despite that, data synopsis can be extracted from the data stream, yet the design depends on the problem design. One of the useful methods for data summarization is hierarchical clustering wherein groups are given data into a tree of clusters. According to [6] [7] , they confirmed that the most suitable summarisation method for density-based clustering algorithms is micro-clustering

and grid-based. Table 1 shows related data stream algorithms and their main features based on the above discussion.

Many algorithms have been developed for data stream clustering. The first attempt was to develop an algorithm that has two phases; online and offline. The online phase is used for data summarization, while the offline phase forms the final cluster. From the above-mentioned required characteristics, an appropriate algorithm is needed to properly handle evolving data streams that contain outliers. Many algorithms followed Aggrawal, such as the Incremental Data Stream Clustering Algorithm Based on Dense Units Detection (DUCStream) in 2005. As well as DStreamII and DenStream and many other data stream clustering algorithms Table 1 summarizes the most common data stream algorithms with their features.

TABLE 1. Summarise data stream clustering algorithms with their main features

Algorithm	**year**	**Limited time**	**Limited memory**	**Handling outliers**	**Handling Evolving data**	**Windowing**	**Processing**	**Summarisation**
DUCStream	2005				✓	landmark	Single	-
CluStream	2007			✓	✓	pyramidal	Online/offline	micro
DDStream	2008				✓	Fading	Online/offline	grid
DStreamII	2009		✓	✓	✓	Moving	Online/offline	grid
DenStream	2011	✓	✓		✓	Fading	Online/offline	micro
PKS-Stream	2012	✓	✓		✓	Sliding	Online/offline	grid/PKS-tree
DDenStream	2013	✓		✓	✓	Fading	Online/offline	micro
HDDStream	2014		✓	✓	✓	Fading	Online/offline	core-micro
MuDi-Stream	2016	✓	✓	✓	✓	Fading	Online/offline	Micro+cmc
EDMStream	2017		✓	✓	✓	-	online	micro
CEDAS	2017	✓	✓	✓	✓	-	online	micro
evoStream	2018	✓			✓	Fading	Online/offline	micro
BOCEDS	2019	✓	✓	✓	✓	-	online	micro

4. OUTLIER DETECTION FOR EVOLVING DATA STREAMS

The difference between evolving and not evolving data stream is an important issue. In order to handle the evolving data over time as well as the evolving features fading window with decay factor $f(t) = 2^{-\lambda t}$ [8] is commonly used. Many studies have been reported in assigning weights for evolving data stream. Some studies considered only the density factor, while others considered only

time intervals. Table 2 illustrates the relevant factors, along with the deployed equations. As can be seen in Table 2 many data stream clustering algorithms been used to process the data stream and each algorithm considered one or some of the important factors that effect on data stream weight. However, by excluding one of the factors which affect the data stream, the weight will be affected on the weight of the objects. As a result, the clustering quality will decrease as well as the misclassification between normal and outlier objects will increase. To better detection for the outliers, the new weighted function should be proposed to avoid the misclassification between outliers and normal data points.

TABLE 2. Weighted clustering for evolving data stream

No.	Year	Title	Authors	Methods	Cluster Weight
1	2014	DEC: Dynamically Evolving Clustering and Its Application to Structure Identification of Evolving Fuzzy Models	Baruah and Angelov	dynamically evolving clustering	$\frac{1-\gamma^{t+1}}{1-y}$
2	2014	Dynamically Evolving Clustering for Data Streams	Baruah et al.	Dynamically Evolving Clustering	γ^{t-t_k}
3	2015	Pattern Detection in Cyber-Physical Systems	Spezzano, Vinci	data stream clustering algorithm for pattern detection	$\frac{v}{1-2^{-\lambda}}$
4	2016	A fast density-based data stream clustering algorithm with cluster centres self-determined for mixed data [10]	Chen, He	A fast density-based data stream clustering algorithm (Str-FSFDP)	$2^{-\lambda(t_{-}\ t_{c})}$
5	2016	Clustering Data Streams Based on Shared Density Between Micro-Clusters	Hahsler, Bolanos	density-based stream clustering. (DBSTREAM)	$2^{-\lambda t_{gap}}$
6	2016	A new Growing Neural Gas for clustering data streams	Ghesmoune et al.	Growing Neural Gas over Data Streams. (G-Stream)	$\sum_{i=1}^{m} 2^{-\lambda_1(t-t_{i0})}$
7	2016	MuDi-Stream: A multi density clustering algorithm for evolving data stream	Amini et al.	MuDi-Stream algorithm	$\frac{1-2^{-\lambda(t+1)}}{1-2^{-\lambda}}$
8	2016	Self-organising anomaly detection in data streams	Forestiero	bio-inspired clustering algorithm	$\frac{v}{1-2^{-\lambda}}$
9	2017	Ensemble learning for data stream analysis: A survey	Krawczyk et al.	Survey	No specific equation
10	2018	evoStream – Evolutionary Stream Clustering Utilizing Idle Times	Carnein, Trautmann	evoStream	$2^{-\lambda(t-mc(t))}$
11	2018	Clustering Based on Correlation Fractal Dimension Over an Evolving Data Stream	Yarlagadda et al	Grid-based clustering algorithm (FractStream)	$\sum_{i=1}^{m} 2^{-\lambda_1(t-t_{i0})}$

4.1. **Proposed Weighted Cluster.** In data stream each data point (x), a weight coefficient will be considered and this weight decreases exponentially over time t. The initial value for the data point will be given (4.1). The weight of data point x at time t will be calculated based on Fading function $f(t) = 2^{-\lambda t}$, The parameter λ will be used to control the importance of the historical data of the stream and $\lambda > 0$,

$$f(t) = 2^{-\lambda t}, \tag{4.1}$$

$$w(x,t) = 2^{-\lambda t}, \tag{4.2}$$

(weight for data point x at time t).

The following are the input data sequence of samples $x1, x2, \ldots, xi, \ldots$, arriving at time stamps $T1, T2, \ldots, Ti, \ldots$. For each data point x, if x arrives at time t_c, whose time stamp is defined as$T(x) = t_c$, and weight coefficient $w(x,t)$ at time t is defined as:

$$w(x,t) = 2^{-\lambda(t-T_x)} = 2^{-\lambda(t-t_c)}. \tag{4.3}$$

Where $\lambda \in (0,1)$ is a constant value called decay factor. The weight for each cluster is the total weight for the data points inside the cluster,

$$w(c,t) = \sum w(x,t) = \sum_{x \in c} 2^{-\lambda(t-t_c)}. \tag{4.4}$$

The data points weight of any cluster is changing over time, However, to save the computational time is not necessary to update the weight value at every time stamp. Instead of that, the data points weight can be updated only when new data point join the cluster. In addition to that, to keep the historical information for the time stamp of the last data should be recorded, based on that, the cluster weight can be updated. The cluster weight is updated in t_c with the last updated value t_p as follows $(t_c > t_p)$:

$$w_c(t_p, t_c) = 2^{-\lambda(t_c - t_p)} * w_c(t_p) + 1. \tag{4.5}$$

While the weight of the data stream is a constant $w_s = \frac{v}{1-2^{-\lambda}}$ where v is the data stream speed i.e. the number of data points arrived in one time unit. The speed of the data stream effect on the weight of the data points as well as on the cluster weight where each cluster represents a mini stream. In case of all data points are mapped to one cluster, the weight for that cluster has the same weight for the data stream that represents the total weight for all data points in the stream.

$$w(x,t) = v * 2^{-\lambda(t-t_c)}. \tag{4.6}$$

We have:

$$w(x,t) = \sum_{x \in c} 2^{-\lambda(t-t_c)} = v \sum_{x \in c} 2^{-\lambda(t-t_c)}, \tag{4.7}$$

the weight for all data points in data stream can be transformed by using the sum formula for geometric series as:

$$w_c(t) = v \sum_{t=1}^{t} 2^{-\lambda(t-t')} = \frac{v(1-2^{-\lambda(t+1)})}{1-2^{-\lambda}}. \tag{4.8}$$

The maximum weight can be calculated as following:

$$w_c(t) = \lim_{t\to\infty} \frac{v(1-2^{-\lambda(t+1)})}{1-2^{-\lambda}} = \frac{v}{1-2^{-\lambda}}. \tag{4.9}$$

5. Conclusion

Studies on evolving data stream clustering have been reported in various means. The discussion was focused on the issue of handling evolving data stream which is the main feature of many emerging data sources such as Internet of Things. In this paper, an extended amendment in the clustering function is proposed based on the MuDi-Stream algorithm. A weighted function has been proposed to facilitate the process of clustering where the function includes factors gathered from the literature of data stream clustering. To demonstrate the effectiveness of this function, relevant experiment will later be performed on benchmark datasets.

References

[1] M.M. Breunig, H.-P. Kriegel, R.T. Ng, J. Sander: *LOF: Identifying Density-Based Local Outliers*, Proc. 2000 Acm Sigmod Int. Conf. Manag. Data, (2000), 93–104.

[2] A. Bouguettaya, Q. Yu, X. Liu, X. Zhou, A. Song: *Efficient agglomerative hierarchical clustering*, Expert Syst. Appl., **42**(5) (2015), 2785–2797.

[3] V. Siless, K. Chang, B. Fischl, A. Yendiki: *Hierarchical clustering of tractography streamlines based on anatomical similarity*, Neuroimage, 2017.

[4] A. Amini, H. Saboohi, T. Herawan, T.Y. Wah: *MuDi-Stream: A multi density clustering algorithm for evolving data stream*, J. Netw. Comput. Appl., **59**(2016), 370–385.

[5] A.C. Bryant, K.J. Cios: *SOTXTSTREAM: Density-based self-organizing clustering of text streams*, PLoS One, **12**(7) (2017), 5–7.

[6] C.C. Aggarwal, J. Han, J. Wang, P.S. Yu: *Chapter 2 ON CLUSTERING MASSIVE DATA STREAMS?: A SUMMARIZATION PARADIGM*, in Data Streams Models, Algorithms, **31**(2007), 9–38.

[7] J. LIN, E. KEOGH, W. TRUPPEL: *(Not) finding rules in time series: A surprising result with implications for previous, future research*, in Proceedings of the International Conference on Artificial Intelligence IC-AI 2003, **1**(2003), 55–61.

[8] S. MANSALIS, E. NTOUTSI, N. PELEKIS, Y. THEODORIDIS: *An evaluation of data stream clustering algorithms*, Stat. Anal. Data Min., **11**(4) (2018), 167–187.

[9] A. FORESTIERO: *Self-organizing anomaly detection in data streams*, Information Sciences, **373**(2016), 321–336.

[10] J.Y. CHEN, H.H. HE: *A fast density-based data stream clustering algorithm with cluster centers self-determined for mixed data*, Information Sciences, **345**(2016), 271–293.

SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA
MALAYSIA
Email address: alaafareed85@gmail.com

SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA
MALAYSIA
Email address: yuhanis@uum.edu.my

SCHOOL OF COMPUTING
UNIVERSITI UTARA MALAYSIA
MALAYSIA
Email address: yazman@uum.edu.my