Density Micro-Clustering Algorithms on Data Streams: A Review

Amineh Amini, Teh Ying Wah

***Abstract*—Data streams are massive, fast-changing, and infinite. Applications of data streams can vary from critical scientific and astronomical applications to important business and financial ones. They need algorithms to make a single pass with limited time and memory. Mining data streams is concerned with extracting knowledge structures represented in models and patterns in non-stopping data streams. Clustering is a prominent task in mining data streams, which group similar objects in a cluster. Several clustering algorithms have been introduced in recent years for data streams that are based on distance, so they can find only spherical shapes. Therefore, density-based clustering algorithms are adopted for data streams with ability for not only discovering the arbitrary shape clusters, but also for providing protection against the outliers. In fact, in density-based clustering algorithms, dense areas of objects in the data space are considered as clusters, which are segregated by low density area (noise). However, in the clustering data streams, due to certain characteristics, it is impossible to record all the data. Micro-clusters are a technique in stream clustering that maintains the compact information about the data objects in data streams. Micro-cluster is a temporal extension of the cluster feature, which compresses the data effectively. In this paper, we intend to review the outstanding density-based clustering algorithms on data streams using micro-clusters. We will explore algorithm characteristics and analyze their merits and limitations.**

***Index Terms*—Data streams, Density-based clustering, Micro cluster**

I. Introduction

IN recent years, mining data streams has become prominent as sensors and the other devices that generate data streams are used extensively. Mining data streams relate to extracting useful pattern from data streams. Data streams need to be processed as they arrive. Therefore, real-time analysis and mining of data streams have attracted substantial amount of researches [1], [2], [3], [4], [5]. With the applicability of data streams, clustering data streams have received more attention in data mining research. In clustering data streams, the aim is to cluster the data streams continuously so there is an up-to-date clustering of all objects seen so far.

Clustering data streams posed additional challenges [6] such as:

- **Single pass clustering**: data arrive contentiously so the clustering has to be done in a single pass over the data
- **Limited time**: the algorithm has to handle the speed of data streams for clustering
- **Limited memory**: since data streams are infinite, the storing of all data streams is impossible
- **Number and size of clusters**: according to data streams characteristics, the number and the shape of clusters are unknown in advance.
- **Evolving data**: the algorithm has to consider that the data streams may change over the time
- **Noisy data**: the clustering algorithm has to handle the noise in data, which affects clustering.

Some of the clustering methods that have been developed are distance-based. As such, they are suitable to find ellipsoid-shaped clusters, or at best convex clusters. However, for non-convex clusters, these methods have trouble finding the true clusters, since two points from different clusters may be closer than two points in the same cluster. Nevertheless, within one cluster, there must be enough intermediate points such that we can reach from one end of the cluster to another [7].

Density-based clustering algorithms are developed based on density notion of clusters. They are designed to discover clusters of arbitrary shape and to handle outliers. In fact, in these clustering algorithms the high density area is separated from the low one. Density is defined as the number of points within a specified radius [8]. A density-based cluster is a set of density-connected objects that is maximal with respect to density-reachability. Every object not contained in any cluster is considered to be noise [9].

In clustering data streams, we cannot save all the incoming data objects due to limited memory. Micro-clusters are a method in stream clustering, which is used to record summary information about the data objects in the streams. Several algorithms are such as [10], [6], [11] and [12] developed that are used micro-clusters for their clustering.

These algorithms are distance- or density-based. However, the ability of density-based clustering to discover clusters of arbitrary size and shape, together with the fact that it satisfies the requirements for data streams clustering, made us choose the density-based clustering, which used micro-clusters for recording compact information about the clusters.

The main objective of this paper is to review the density-based clustering algorithms specially developed for data streams, as well as using micro-clusters for saving synopsis information about the clusters.

The rest of the paper is organized as follows. Section II surveys related work. Section III provides an overview of the density micro-clustering based algorithms on data stream. Section IV presents discussion, and Section V concludes our study.

II. Related Works

A number of density-based clustering algorithms were proposed at the end of the previous decade, including DBSCAN [8], DENCLUE [13], OPTICS [14], and CLIQUE [15].

A. Amini is a PhD student in the Department of Information Science, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603 Kuala Lumpur, Malaysia.
e-mail: amini@siswa.um.edu.my

Dr. T. Ying Wah is a senior lecturer in the Department of Information Science, Faculty of Computer Science and Information Technology, University of Malaya (UM), 50603 Kuala Lumpur, Malaysia.
e-mail: tehyw@um.edu.my

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

DBSCAN[1] has been designed for the clustering of large noisy datasets on spatial data. DBSCAN introduced the concept of neighborhoods as a region of given radius (i.e. a sphere) and containing a minimum number of data points. Connected neighborhoods form clusters, thus departing from the notion of spherical cluster. It grows clusters according to a density-based connectivity analysis.

DENCLUE[2] is also based on neighborhoods, focusing on high-dimensional multimedia databases. Within the multidimensional data space, DENCLUE computes the impact of a data point upon its neighborhood and uses it to assign data points to the clusters. It clusters objects based on a set of density distribution functions.

OPTICS[3] is not a clustering algorithm; rather, it contributes in identifying the clustering structure by ordering points and reachability distances in a way that can be exploited by a density-based algorithm. It extends DBSCAN to produce a cluster ordering obtained from a wide range of parameter settings. Like DBSCAN and DENCLUE, it observes density as a regional phenomenon.

CLIQUE[4] combines a density-based and a grid-based approach to find clusters embedded in subspaces of a high-dimensional data space. It partitions each dimension like a grid structure and determines whether a cell is dense, based on the number of points it contains. It partitions the data space into rectangular units that are considered dense if they contain a minimum number of points [16].

However, these algorithms are not applicable for data streams. They are developed for large or spatial database. Several density-based clusterings have been developed in recent years, which adopted these algorithms for data streams. On the other hand, some algorithms use micro-cluster for saving summary information about the clusters that are not density-based, like [10], [6], but they designed for data streams. We intend to review density-based clustering algorithms based on data streams. Besides that, they used micro-clusters concept for their clustering.

III. Density Micro-Clustering Algorithms

In this section, we will introduce and analyze the outstanding density-based clustering algorithm based on micro-clusters. One of the well-known designs for clustering data stream is two-phase clustering, which Aggarwal et al. [10] introduce. The two-phase clustering separates the clustering process into online and offline components. In this online-offline way, the online phase captures synopsis information from the data stream, and the offline phase generates clusters on the stored synopsis information.

Density-based clustering has the ability to discover clusters in any shape. It defines the clusters by separating dense area from sparse ones. Among the density-based algorithms that are explained earlier in this paper, DBSCAN is used in the offline phase for clustering algorithms on data streams.

In clustering data streams, it is impractical to save all the incoming data objects. Micro-clusters are a popular technique in stream clustering, which maintain the compact representation of the clustering. Micro-cluster, which was first introduced in [10], is a temporal extension of cluster feature vector. According to [17] and [10], we can define the cluster features and the micro-cluster as follows:

Definition 1: **Cluster Feature (CF)**: Cluster feature [17] is a triple summarizing, which is maintained about a cluster. It is a triple vector, which include the number of the data points, the linear sum of data points, and the squared sum of them.

$$CF = (N, \overrightarrow{LS}, SS)$$

- N is the number of data points in a cluster.
- $\overrightarrow{LS} = \sum_{i=1}^{N} \overrightarrow{X_i}$ is the linear sum of the N data points.
- $SS = \sum_{i=1}^{N} {X_i}^2$is the square sum of data points.

The most important characteristic of a cluster feature is the additive. If $CF_1 = (N_1, \overrightarrow{LS}_1, SS_1)$ and $CF_2 = (N_2, \overrightarrow{LS}_2, SS_2)$ are two disjoint clusters, the CF vector of the merged cluster of two clusters will be:

$$CF_1 + CF_2 = (N_1 + N_2, \overrightarrow{LS}_1 + \overrightarrow{LS}_2, SS_1 + SS_2).$$

The additive property makes them very useful for data streams cluster analysis. Furthermore, a large number of micro-clusters can be maintained without using a great deal of memory. Therefore, based on *cluster feature* definition, the micro-cluster is defined as follows:

Definition 2: **Micro-Cluster**: Micro-cluster extends the CF vector by summing up the timestamps. A micro-cluster for a set of *d-dimensional* points defines as $(\overrightarrow{CF2^x}, \overrightarrow{CF1^x}, CF2^t, CF1^t, n)$. $\overrightarrow{CF2^x}, \overrightarrow{CF1^x}$ are equivalent to SS and LS respectively. The timestamps entries are defined as follows:

- $CF2^t$: The sum of squares of timestamps $T_{i1} \ldots T_{in}$.
- $CF1^t$: The sum of timestamps $T_{i1} \ldots T_{in}$.

The micro-cluster for a set of points C denote by $\overrightarrow{CFT(C)}$.

The micro-clusters framework in density-based clustering is shown in Fig. 1 [18].

In the following sections, four remarkable algorithms (*DenStream*, *C-Denstream*, *rDenStream*, and *SDStream*) are explored, and the pros and cons of them are discussed in a separate section. *DenStream* will be explained in more detail in the next section, since all the other algorithms are based on it.

A. DenStream

DenStream [12] is a density-based clustering solution for data streams. It extends the micro-cluster concept [10]. Instead of using the number of points that are in the neighborhood as density concept like DBSCAN, micro-cluster density is based on weighting areas of points in the neighborhood. It uses the *fading function* [19] and aggregates the weights in micro-clusters. *DenStream* has two phases -online-offline- and is based on CluStream [10] framework.

In evolving data streams, the role of clusters and outliers frequently exchange. Therefore, the structure of *p-micro-cluster* [12] and *o-micro-cluster* [12] are introduced for keeping the difference among potential clusters and outliers. The main differences between them are their different constraints on their weights. In addition, *outlier buffer* is defined for

[1]Density-Based Spatial Clustering of Applications with Noise
[2]DENsity-based CLUstEring
[3]Ordering Points to Identify the Clustering Structure
[4]CLustering InQUEst

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

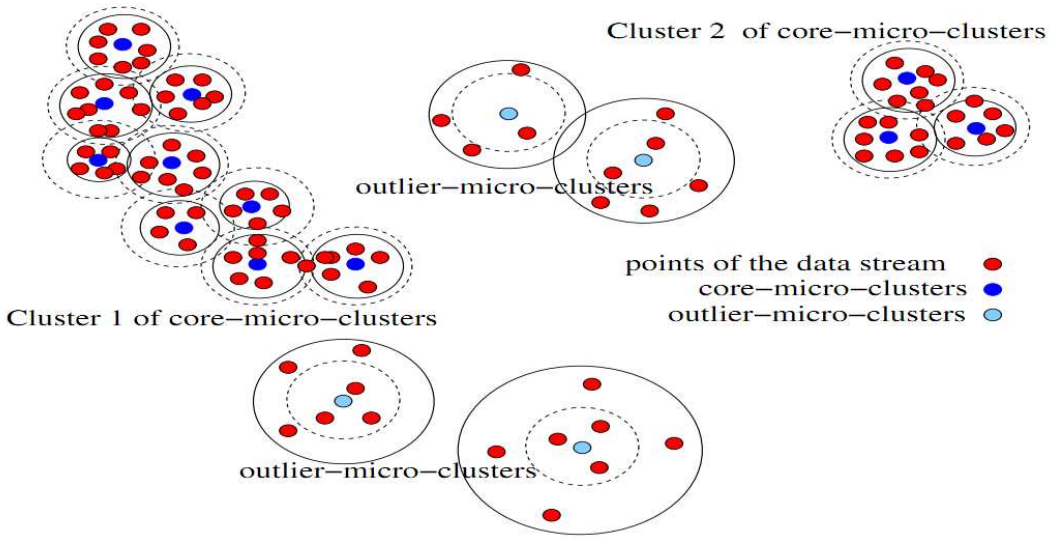

Fig. 1. Micro-Clusters framework in density-based clustering [18]

separation of the process of the *p-micro-clusters* and *o-micro-clusters*. In [12] the *p-micro-cluster* and *o-micro-cluster* are defined as follows:

The *p-micro-cluster* is actually a potential *c-micro-cluster*, so first we need to define the *c-micro-cluster* concept. $f(t)$ a *fading function*, used in the following definitions, explained in [19].

Definition 3: **core-micro-cluster (c-micro-cluster)**: A *c-micro-cluster* defines as $CMC(w, c, r)$ for a group of close points $p_{i1} \dots p_{in}$ with timestamps $T_{i1} \dots T_{in}$.

- $w = \sum_{j=1}^{n} = f(t - T_{ij})$, is the weight and $w \geq \mu$.
- $c = \frac{\sum_{j=1}^{n} f(t-T_{ij}) p_{ij}}{w}$ is the center.
- $r = \frac{\sum_{j=1}^{n} f(t-T_{ij}) dist(p_{ij}, c)}{w}$, $r \leq \epsilon$ is the radius. $dist(p_{ij}, c)$ is *Euclidean distance* between point p_{ij} and the center c.

Definition 4: **potential c-micro-cluster (p-micro-cluster)**: *p-micro-cluster* at the time t for a group of close points $p_{i1} \dots p_{in}$ with timestamps $T_{i1} \dots T_{in}$ defines as $(\overrightarrow{CF^1}, \overrightarrow{CF^2}, w)$.

- $w = \sum_{j=1}^{n} f(t - T_{ij})$, is the weight and $w \geq \beta\mu$. β is the parameter to determine the threshold of the outlier relative to *c-micro-clusters* ($0 < \beta < 1$).
- $\overrightarrow{CF^1} = \sum_{j=1}^{n} f(t-T_{ij}) p_{ij}$, is the weighted linear sum of the points.
- $\overrightarrow{CF^2} = \sum_{j=1}^{n} f(t - T_{ij}) p_{ij}^2$ is the weighted squared sum of the points.

Definition 5: **outlier micro-cluster (o-micro-cluster)**: The definition of *o-micro-cluster* is similar to *p-micro-cluster*. It is defined as

$$(\overrightarrow{CF^1}, \overrightarrow{CF^2}, w, t_0)$$

However, it considers $t_0 = T_{i1}$, relate to the creation time of *o-micro-cluster*. It is used to determine the life span of the *o-micro-cluster*. In addition to, it considers as an outlier since the micro-cluster weight is below the threshold of outlier weight $w \leq \beta\mu$.

DenStream divides the process of clustering into two parts. First, an online maintenance step of micro-clusters is carried out followed by an offline step generates the final clusters using the clustering algorithm DBSCAN.

Online Phase (micro-cluster upkeep): The *p-micro-clusters* and *o-micro-cluster* are maintained in an online way. All the *o-micro-clusters* are maintained in a separate memory space, which is called *outlier-buffer*. When a new point arrives, it will be merged with the existing micro clusters as follows:

1) Merge a new point with the nearest *p-micro-cluster*,
   - if the new radius of *p-micro-cluster* after adding the new point is below or equal to the threshold, which is considered for radius of *c-micro-cluster* .
   - Otherwise, merge the new point with the nearest *o-micro-clusters*
2) The weight of new *o-micro-cluster* is checked,
   - if it is higher than the threshold, it means that it grows into *p-micro-cluster*, so it will be removed from the *outlier buffer*, and a new *p-micro cluster* is created.
   - Otherwise, the new *o-micro-cluster* is created by new point and put in *outlier buffer*.

The algorithm puts each new point in *outlier buffer* because the new point cannot fit in any clusters.Therefore, it may be an outlier or the seed of new cluster.

Offline Phase (generating clusters): After capturing the *p-micro-clusters* in the online phase, a variant of DBSCAN algorithm is applied to *p-micro-clusters* to get the final result of clustering. Each *p-micro-cluster* is regarded as a virtual point for clustering. Weight is an important parameter in this variant DBSCAN. For any decision, the weight factor is checked, which should be higher than the threshold.

Furthermore, it has a strategy for distinguishing between the real outliers, and potential ones, which will change to potential micro-clusters. In special time intervals, the algorithm checks the weights of *p-micro-clusters* and*o-micro-clusters* and make a decision based on their weights. If the weight of *p-micro-clusters* is lower than threshold, it will be omitted. It means that *p-micro-clusters* change to outliers, since it does not receive any data for a long time. In addition, it checks

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

the *o-micro-clusters* weight if it is below the lower limit of threshold, it will omit since it is real outliers. Otherwise, the *o-micro-clusters* are considered as a potential one, which will change to *p-micro-clusters*.

The execution time of *DenStream* grows linearly as the stream proceeds. The execution time is evaluated on data stream with various dimensionally and different number of natural clusters. The empirical result shows that the algorithm is more effective than CluStream.

B. SDStream

The *SDStream* algorithm [20] is an offline-online algorithm, which has ability to discover the clusters with arbitrary shapes over sliding windows [21]. In this algorithm, the distribution of the most recent data stream is considered and the data points that are not accommodated in sliding window length are discarded.

SDStream algorithm is based on *SWClustering* algorithm [22] and *DenStream*. It modifies the *c-micro-cluster*, *p-micro-cluster* and *o-micro-cluster* concepts by assigning a weight to each micro-cluster, based on the number of data points in it. The *p-micro-cluster* and *o-micro-cluster* are stored in the form of *Exponential Histogram of Cluster Feature (EHCF)*.

According to [22], the *TCF* and *EHCF* concepts are defined as follows:

Definition 6: **Temporal Cluster Feature (TCF)**: TCF is a temporal extension of *CF* with the timestamp of the most recent record for keeping cluster properties in the sliding window model . It is defined as a $(\overrightarrow{CF2^x}, \overrightarrow{CF1^x}, t, n)$, which is similar to *CF*. t is added as the timestamp of most recent record.

Definition 7: **Exponential Histogram of Cluster Feature (EHCF)**: *EHCF* data structure is proposed to construct cluster features based on sliding window model. In EHCF only the most resent N records are considered at any time. Every bucket in an *EHCF* is a *TCF* for a set of records.
The new data points will be added to the nearest micro cluster (*o-micro-cluster* or *p-micro-cluster*), if the new radius of micro cluster is below or equal to radius threshold. Otherwise, the new micro cluster will be created. As a result, two clusters have to be merged or one micro-cluster is deleted as follows:

- Merging two micro clusters: the two nearest micro clusters, which are *density-reachable* [8] from the other, are chosen and merged. The merging is completed by the merging of two *EHCF* [22].
- Omitting the micro cluster: The outdated micro cluster is deleted. The outdated micro clusters are defined by the value of time in *TCF*. If the value of t in *TCF* does not belong to the bound of N-length sliding window, the *TCF* is also deleted.

All the *density-connected* [8] *p-micro-clusters* form a cluster. Each *p-micro-clusters* views as a virtual point when the clustering request arrives. The final clusters of arbitrary shape generate, based on these virtual points using modified DBSCAN. The authors in this paper claim that *SDStream* has higher cluster quality compared to *CluStream* [10].

C. rDenStream

In[23], Liu et al. develop a three-step clustering algorithm based on *DenStream*. In *rDenStream* the discarded micro-clusters are kept in outside temporary memory, giving them a new chance to tend to clustering and improve the accuracy of clustering. This algorithm is usable for applications with large amounts of outliers.

The first two phases of *rDenStream* are similar to *DenStream*; however, it has one more phase, which is called the retrospect. In this phase, the algorithm will learn from discarded data to improve the accuracy of the algorithm. It gives a chance to misjudge points to relearn and improve the robustness of the clustering.

The experimental result shows that it has better performance than *DenStream* in the initial phase. The time complexity is more than *DenStream* since it has to process the historical buffer. In addition, the memory usage is more than *DenStream* because it saves the historical outlier buffer.

D. C-DenStream

Authors in [11] developed a density-based algorithm for data streams that includes domain information in the form of constraints. They extend the static semi-supervised clustering for streams and propose a method for the use of background knowledge in data streams. Semi-supervised clustering methods exploit background knowledge to guide the clustering process. Instance level constraints are a specific and popular form of background knowledge. They refer to instances that must belong to the same cluster (Must-Link constraints) and those that must be assigned to different clusters (Cannot-link constraints) [24]. Authors apply this semi-supervised approach to extend DenStream algorithm. C-DenStream is extended the notion of instance level constraints from static data to data streams, focusing on density-based clustering and cluster evolution in streams.

It is an extension of DenStream allowing the inclusion of domain information, at the same time as requirements for data stream algorithms are satisfied. They used the concepts of contrariness between micro clusters. They modify the *DenStream* algorithms as follows:

- Instead of using DBSCAN in the offline phase, they used C-DBSCAN [25] to include constraints.
- They put constraint in *DenStream* phase of their algorithms by using the concept of constraint between micro-clusters when they are created, removed and maintained [10].

They show that an infinite number of constraints cannot be stored; it needs effective management. Therefore, they showed how to transform the instance level constraints into micro-cluster level constraints.

IV. Discussions

In this section, we will discuss the density clustering algorithms that have been introduced in the previous sections.

The *DenStream* algorithm considers weights for data points, so the outdated data is eliminated, which is very useful in data streams. It also saves time since it does not merge data into a micro-cluster and then address the outliers. It determines the outliers before merging. *rDenStream* build on *DenStream*; however, it can handle the outliers very well with high accuracy. On the other hand, it has higher time complexity compared to *DenStream*, since it processes the

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

historical outlier buffer. In addition, it needs more memory space for saving the historical buffer.

SDStream uses the EHCF synopsis data structure that can better track the cluster evolution while consuming much less memory. EHCF provides a flexible framework for analyzing the cluster evolution. This algorithm is applicable for the applications in which the distributions of most recent data streams are important. *C-DenStream* uses background knowledge for guiding clustering and putting constraint on micro-cluster. It prevents the formation of the clusters that do not conform to the applications' semantics. For example, geographical objects, e.g. houses separated by a borderline or a river may, not be assigned to the same cluster, independently of their physical proximity. Therefore, it is applicable in real applications.

All of these algorithms have their pros and cons. They cluster data streams from different perspectives. While some of them emphasize the handling of outliers, others ignore it. Some are more accurate, though they have a high complexity. Choosing the time window for processing data stream is also different in these algorithms; while some of them consider the whole data stream, the others only use the most recent data streams. Consequently, choosing these algorithms depends on what we want from the algorithms, such as low time complexity, high accuracy, the distribution of our data, and so on.

V. Conclusions

Clustering data streams places additional constraints on clustering algorithms. Data streams require algorithms to make a single pass over the data with bounded memory and limited processing time, whereas the stream may be highly dynamic and evolve over time. Several clustering algorithms are introduced for data streams that are distance-based and cannot handle the interwoven clusters. Besides that, saving the data streams is impossible, due to the infinite characteristic. Consequently, micro-cluster is introduced to record a summary of data.

We explore four density-based clustering algorithms using micro-clusters. These algorithms utilize the density-based clustering because of their ability to find any shape clusters and micro-clusters as a general summarization of incoming data streams for solving data mining problems on streams.

The algorithms are two-phase, online and offline, in which the online phase maintains the micro clusters and offline phase generate the final clusters based on DBSCAN.

As a future work, we will implement all these algorithms and compare them based on the cluster quality on single dataset.

References

[1] C. Aggarwal, Ed., *Data Streams – Models and Algorithms*. Springer, 2007.

[2] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80.

[3] M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Data Stream Mining," *Data Mining and Knowledge Discovery Handbook*, pp. 759–787, 2010.

[4] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom, "Models and issues in data stream systems," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, ser. PODS '02. New York, NY, USA: ACM, 2002, pp. 1–16.

[5] M. M. Gaber, A. Zaslavsky, and S. Krishnaswamy, "Mining data streams: a review," *SIGMOD Rec.*, vol. 34, pp. 18–26, June 2005.

[6] P. Kranen, I. Assent, C. Baldauf, and T. Seidl, "The ClusTree: indexing micro-clusters for anytime stream mining," *Knowledge and Information Systems*, pp. 1–24, 2010.

[7] G. Gan, C. Ma, and J. Wu, *Data Clustering: Theory, Algorithms, and Applications (ASA-SIAM Series on Statistics and Applied Probability)*. SIAM, Society for Industrial and Applied Mathematics, May 2007.

[8] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, E. Simoudis, J. Han, and U. Fayyad, Eds. AAAI Press, 1996, pp. 226–231.

[9] J. Han, *Data Mining: Concepts and Techniques*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005.

[10] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams," in *VLDB '2003: Proceedings of the 29th international conference on Very large data bases*. VLDB Endowment, 2003, pp. 81–92.

[11] C. Ruiz, E. Menasalvas, and M. Spiliopoulou, "C-DenStream: Using domain knowledge on a data stream," in *Proceedings of the 12th International Conference on Discovery Science*, ser. DS '09. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 287–301.

[12] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *SIAM Conference on Data Mining*, 2006, pp. 328–339.

[13] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *KDD*, 1998, pp. 58–65.

[14] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "Optics: ordering points to identify the clustering structure," *SIGMOD Rec.*, vol. 28, pp. 49–60, June 1999.

[15] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," *SIGMOD Rec.*, vol. 27, pp. 94–105, June 1998.

[16] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, "Density-based semi-supervised clustering," *Data Min. Knowl. Discov.*, vol. 21, pp. 345–370, November 2010.

[17] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, J. Widom, Ed. ACM Press, 1996, pp. 103–114.

[18] D. K. Tasoulis, G. Ross, and N. M. Adams, "Visualising the cluster structure of data streams," in *Proceedings of the 7th international conference on Intelligent data analysis*, ser. IDA'07. Berlin, Heidelberg: Springer-Verlag, 2007, pp. 81–92.

[19] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for projected clustering of high dimensional data streams," in *Proceedings of the Thirtieth international conference on Very large data bases - Volume 30*, ser. VLDB '04. VLDB Endowment, 2004, pp. 852–863.

[20] J. Ren, R. Ma, and J. Ren, "Density-based data streams clustering over sliding windows," in *Proceedings of the 6th International Conference on Fuzzy systems and Knowledge Discovery (FSKD)*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 248–252.

[21] W. Ng and M. Dash, "Discovery of frequent patterns in transactional data streams," in *Transactions on Large-Scale Data- and Knowledge-Centered Systems II*, ser. Lecture Notes in Computer Science, A. Hameurlain, J. Kng, R. Wagner, T. Bach Pedersen, and A. Tjoa, Eds. Springer Berlin / Heidelberg, 2010, vol. 6380, pp. 1–30.

[22] 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, May 2008.

[23] L. Li-xiong, K. Jing, G. Yun-fei, and H. Hai, "A three-step clustering algorithm over an evolving data stream," in *Proceedings of IEEE International Conference on Intelligent Computing and Intelligent Systems (ICIS)*, 2009, pp. 160–164.

[24] K. Wagstaff, C. Cardie, S. Rogers, and S. Schrödl, "Constrained k-means clustering with background knowledge," in *Proceedings of the Eighteenth International Conference on Machine Learning*, ser. ICML '01. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001, pp. 577–584.

[25] C. Ruiz, M. Spiliopoulou, and E. Menasalvas, "C-DBSCAN: Density-based clustering with constraints," in *Proceedings of the International Conference on Rough Sets Fuzzy Sets Data Mining and Granular Computing*, vol. 4481. Springer-Verlag, 2007, pp. 216–223.

ISBN: 978-988-18210-3-4
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)