Journal of Computer and Communications, **2013, 1, 26-31**
http://dx.doi.org/10.4236/jcc.2013.15005 Published Online October 2013 (http://www.scirp.org/journal/jcc)

LeaDen-Stream: A Leader Density-Based Clustering Algorithm over Evolving Data Stream

Amineh Amini, Teh Ying Wah

Department of Information Systems, Faculty of Computer Science and Information Technology, University of Malaya (UM), Kuala Lumpur, Malaysia.
Email: amini@siswa.um.edu.my

Received August 2013

ABSTRACT

Clustering evolving data streams is important to be performed in a limited time with a reasonable quality. The existing micro clustering based methods do not consider the distribution of data points inside the micro cluster. We propose LeaDen-Stream (Leader Density-based clustering algorithm over evolving data Stream), a density-based clustering algorithm using leader clustering. The algorithm is based on a two-phase clustering. The online phase selects the proper mini-micro or micro-cluster leaders based on the distribution of data points in the micro clusters. Then, the leader centers are sent to the offline phase to form final clusters. In LeaDen-Stream, by carefully choosing between two kinds of micro leaders, we decrease time complexity of the clustering while maintaining the cluster quality. A pruning strategy is also used to filter out real data from noise by introducing dense and sparse mini-micro and micro-cluster leaders. Our performance study over a number of real and synthetic data sets demonstrates the effectiveness and efficiency of our method.

Keywords: Evolving Data Streams; Density-Based Clustering; Micro Cluster; Mini-Micro Cluster

1. Introduction

Mining data stream became more prominent in many applications, including real-time detection of anomalies in computer network traffic, web searches, monitoring environmental sensors, social networks, sensor networks, and cyber-physical systems [1]. In these applications, data streams arrive continuously and evolve significantly over time. Mining data streams is related to extracting knowledge structure represented in streams information. Clustering is a significant data streams' mining task [2-6]. However, clustering in data stream environment needs some special requirements due to the data stream's characteristics such as clustering in limited memory and time with single pass over the evolving data streams and further handling noisy data [7-9].

There are various methods of clustering in the literature such as partitioning and hierarchical, which are developed to find spherical-shape clusters. One of the important classes in clustering is density-based clustering which can discover the clusters of non-spherical shape and filter out the outliers. The density-based clustering algorithms can find non-spherical shape clusters and are useful for identifying the noise. Some typical examples of density-based algorithms include DBSCAN [10], OPTICS [11], and DENCLUE [12]. The main idea in these algorithms is to consider the dense area of points in the data space as clusters, which are separated by low-density area (noise). Another method of clustering is grid-based which has fast processing time and is independent from the number of data points. Moreover, some algorithms are developed based on the integration of grid and density based termed as density grid based clustering algorithms [13].

Micro clustering is a remarkable method in stream clustering to compress data streams effectively and to record the temporal locality of data [6]. The micro-cluster was first proposed in [14] for large data sets, and subsequently adapted in [5] for data streams. The micro-clustering method for clustering data streams has attracted considerable attention in literature [5,15-20].

In [21], a two-level hybrid DBSCAN algorithm, L-DBSCAN, is proposed. First, it searches each point in dataset and finds out the coarse leaders at a coarse level in order to reduce time complexity. Then, it uses these leaders to determine density-based clusters in a finer level to reduce the deviation of the result. Furthermore, L-DBSCAN is developed into rough-DBSCAN in [22].

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 introduces basic definitions. In Section 4, we explain the LeaDen-

Stream algorithm in details. We conduct experimental study of LeaDen-Stream on real-world and synthetic data sets in Section 5 and conclude the paper in Section 6.

2. Related Work

Algorithms on clustering data streams are categorized as one-scan and evolving approaches. The one-scan approaches cluster the data streams by scanning only once under the assumption that the data arrives in chunks [7,23]. In evolving approaches, the behavior of data streams is defined based on certain time window. Fading window model and sliding window model are widely adopted in stream mining [5,9,15,17,24-26].

Most of clustering algorithms over evolving data streams have two phases firstly introduced by CluStream [5]. CluStream has online and offline phases. The online phase keeps summary information, and the offline phase generates clusters based on synopsis information. However, CluStream, which is based on the k-means approach, finds only spherical clusters. Density-based clustering can overcome this limitation. Therefore, recently density-based clustering is extended in two phase clustering [9,17,24,27,28].

Den-stream [17] is a clustering algorithm for evolving data stream. The algorithm extends the micro cluster [5] concept, and introduces the outlier and potential micro clusters to distinguish between real data and outliers. Den-Stream is based on fading window model in which the importance of micro-clusters is reduced over time if there are no incoming data points.

MR-Stream [9] is an algorithm, which has the ability to cluster data streams at multiple resolutions. The algorithm partitions the data space in cells and a tree like data structure, which keeps the space partitioning. The tree data structure keeps the data clustering in different resolutions. Each node has the summary information about its parent and children. The algorithm improves the performance of clustering by determining the right time to generate the clusters.

D-Stream [24] is a density grid-based algorithm in which the data points are mapped to the corresponding grids and the grids are clustered based on their density. It uses a multi-resolution approach to cluster analysis.

We compared the time complexity and the clustering quality of DenStream, MR-Stream, and D-Stream algorithms. The results are shown in **Figures 1** and **2**. In terms of time complexity, D-Stream has the lowest time complexity; however, it has low quality since the clustering quality depends on the granularity of the lowest level of the grid structure. DenStream has a higher time complexity compared to D-Stream; however, it has a better memory usage and quality. MR-Stream has the highest time complexity and memory usage while it has good quality.

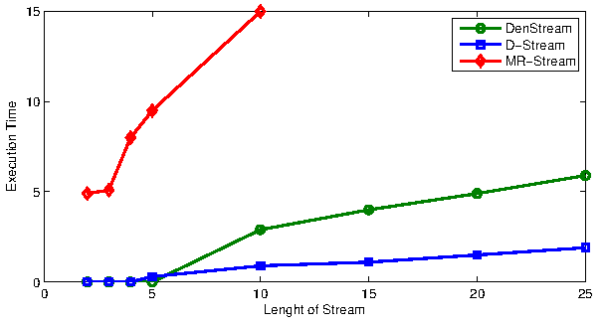

Figure 1. Data stream clustering algorithms time execution comparison.

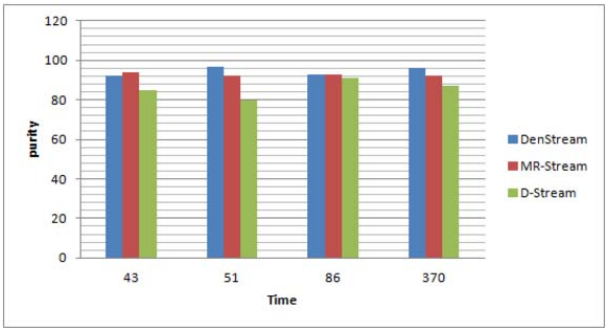

Figure 2. Data stream clustering algorithms quality comparison.

In this paper, we introduce a new algorithm which we call it LeaDenStream with good quality while its time complexity is as low as D-Stream. We introduce new concepts, which are called Mini Micro Leader Cluster and Micro Leader Cluster. We present a new method in which we have to define the granularity of Micro Leaders based on their inside data distribution (which is not considered in any of the existing algorithms). For example, in Den-Stream only the center of potential micro clusters are sent to its offline phase. However, if the data points are not distributed uniformly inside the micro cluster, sending only one representative point for each micro cluster leads to less accuracy. Therefore, using Mini Micro Leader Cluster keeps the quality and Micro Leader-Cluster decreases the time complexity. **Figure 3** shows the Mini Micro Leader Cluster and Micro Leader Cluster in the micro cluster. The situation is compared with DenStream. We also used Mahalanobis distance instead of Euclidean distance for identifying correct cluster center, which increases the quality of clustering as well.

3. Basic Definitions

In this section, we introduce the basic definitions, which form LeaDenStream algorithm.

Definition 1. The Decaying Function:

The fading function [29] used in LeaDen-Stream is

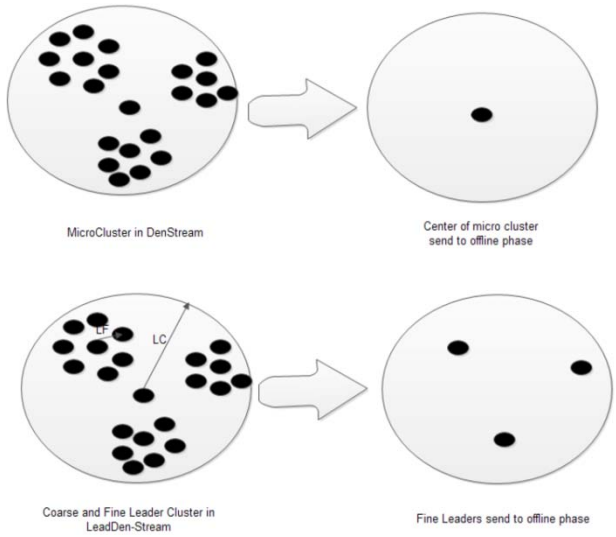

Figure 3. Mini micro and micro leader clusters.

defined as $f(t) = 2^{-\lambda t}$, where $0 < \lambda < 1$. The weight of the data stream points decreases exponentially over time, *i.e*. the older a point gets, the less important it gets. The parameter λ is used to control the importance of the historical data of the stream.

Definition 2. MiniMicroLeaderCluster (MMLC):

A MMLC for a group of data points $p_{i1}...p_{in}$ with time stamp at time *t*, $T_{i1}...T_{in}$, is defined as $\{CF^1, CF^2, W_{mm}, C_{mm}, L_{mm}\}$

- $CF^1 = \sum_{j=1}^{n} f(t-T_j)p_j$: weighted linear sum of the points,
- $CF^2 = \sum_{j=1}^{n} f(t-T_j)p_j^2$: weighted squared sum of the points,
- $W_{mm} = \sum_{j=1}^{n} f(t-T_j)$: MMLC weight,
- $C_{mm} = \frac{CF^1}{W_{mm}}$: MMLC center,
- $L_{mm} = \sqrt{\frac{CF^2}{W_{mm}} - (\frac{CF^1}{W_{mm}})^2}$: MMLC radius

Definition 3. MicroLeaderCluster (MLC):

A MLC for a number of its mini-micro leader is defined as MLC = {{MMLC},W_m,C_m,L_m}

- {MMLC}: MiniMicroLeaderClusters list
- $W_m = \sum_{i=1|}^{\{MMLC\}|} W_{mm}$: MicroLeaqdercluster weight
- $C_m = \frac{\sum_{i=1}^{|\{FLC\}|} C_{mm}}{|\{MMLC\}|}$: MicroLeaderCluster Center
- $L_m = \varepsilon$: MicroLeaderCluster radius which is equal to DBSCAN radius threshold

Definition 4. Dense Mini Micro Leader Cluster (DMMLC):

A MMLC with a weight more than maximum threshold. $W_{mm} > \frac{h_{mm}}{1-2^{-\lambda}} = D_{mm}$

Definition 5. Sparse Mini Micro Leader Cluster (SMMLC):

A MMLC with a weight less than maximum density threshold. $W_{mm} \le \frac{k_{mm}}{1-2^{-\lambda}} = S_{mm}$ h_{mm} and k_{mm} control the threshold since the density cannot exceed $\frac{1}{1-2^{-\lambda}}$ (according to Lemma 1).

Definition 6. DenseMicroLeaderCluster (DMLC):

A MLC, which is in {MMLC} and is dense. DMLC = {{DMMLC}}

Definition 7. SparseMicroLeaderCluster (SMLC):

A MLC, which all its mini-micro leaders are sparse. SMLC = {{SMMLC}}

Definition 8. Leader List Centers ($LL_{centers}$):

A set of centers of DMMLCs and DMLCs, which are sent to the offline phase:

$LL_{centers}$ = {DMMLC} ∪ {DMLC}

Definition 9. Mini Micro Leader Cluster (MMLC) Maintenance: If we have a MMLC at a time *t* and a point *p* arrives in *t*+1 then the statistics become

$$MMLC_{t+1} = \{2^{-\lambda 1}.CF + p, 2^{-} .W_{mm} + 1\}$$

Lemma 1. The maximum weight of the Mini Micro Leader Cluster (MMLC) is $\frac{1}{1-2^{-\lambda}}$

Proof. If we assume that the data point in the data stream adds to the same Mini Micro Leader Cluster (MMLC), the weight is equal to $W_{mm} = \sum_{t'}^{t} 2^{-\lambda(t-t')}$ which can be converted to the following equation:

$$W_{mm} = \sum_{t'}^{t} 2^{-\lambda(t-t')} = \frac{1-2^{-\lambda(t+1)}}{1-2^{-\lambda}}$$

The maximum weight is defined when $t \to \infty$, therefore the maximum is defined as follows:

$$W_{mm_{maximum}} = \frac{1}{1-2^{-\lambda}}$$

Lemma 2. The minimum time for converting DMMLC to SMMLC and vice versa is: $t_{\min\lambda} = \log^{(\frac{k_{mm}}{h_{mm}})}$

Proof. It is given in [17] and [24].

4. LeaDen-Stream Clustering Algorithm

We describe the key components of LeaDen-Stream outlined in **Algorithm 1**. In LeaDen-Stream, when a new data record x arrives, it is added to the Mini-Micro or Micro leader cluster based on the distribution of data in AdjustingLeader-Clusters (**Algorithm 2**). Then, we periodically and in every gap time, which is the minimum

Algorithm 1. LeaDen-Stream(DS, ε, L_m, L_{mm}).

1: Input: a data stream
2: Output: arbitrary shape clusters
3: t = 0;
4: while not end of stream do
5: Read data point *x* from Data Stream
6: AdjustLeaderClusters(x,Lm,Lmm);
7: if t mod t_{min} == 0 then
8: PuringLeaderClusters(MMLC,MLC);
9: end if
10: t=t+1;
11: end while
12: if the clustering request is arrived then
13: — Generate clusters
14: end if

Algorithm 2. Adjust leader clusters (*x*, L_m, L_{mm}).

1: Input: a data point from data stream
2: Output: list of Mini Micro Leader Clusters and Micro Leader Clusters {{MMLC,MLC}}
3: find the nearest MLC center C_m to x
4: if Distance(x, C_m) < l_m then
5: find the nearest MMLC center C_{mm} to x
6: if distance(x, C_{mm}) < l_{mm} then
7: Merge x to the MMLC;
8: else
9: create a new MMLC with x;
10: MMLC = MMLC ∪ { x } ;
11: end if
12: else
13: create a new MLC by x;
14: MLC=MLC ∪ { x } ;
15: end if

time for converting a dense mini-micro leader to a sparse, convert sparse mini-micro leader clusters to dense and vice versa. We remove the sparse mini micro and micro leader clusters in PuringLeaderClusters (**Algorithm 3**).

Our clustering algorithm is divided into two phases:

- Online phase: keeping Mini-Micro and Micro leader clusters
- Offline phase: generating final clusters

4.1. Keeping Mini-Micro and Micro Leader Clusters

This phase is triggered when a data point arrives from data streams. The procedure is described as follows (**Algorithm 2**, Adjust Leader Clusters):

1) We try to find the nearest micro leader cluster to the data point

2) If we find such a micro leader cluster, we try to find nearest mini-micro leader cluster to the data point.

(a) If there is such a mini-micro cluster leader then merge the data point to the nearest mini-micro cluster leader.

(b) Otherwise, form a new mini-micro cluster with *x* as the center of new mini-micro cluster.

3) Otherwise, there is not such micro leader cluster, form a new micro leader cluster with *x* as the center of new micro leader cluster.

Algorithm 3. Puring leader clusters ({MMLC}, {MLC}).

1: Input: list of Mini Micro Leader Clusters and Micro Leader Clusters {MMLC},{MLC}
2: Output: List of centers {$LL_{centers}$}
3: for all {MLC} do
4: check all its mini micro leader clusters {MMLC};
5: if all the {MMLC} are sparse then
6: delete the MLC;
7: end if;
8: if all the {MMLC} are dense then
9: add the MLC center C_m to the $LL_{centers}$
10: $LL_{centers} = LL_{centers} \cup \{ C_m \}$
11: else
12: if some of {MMLC} are dense and some sparse then
13: add all the DMMLC center Cmm to the $LL_{centers}$
14: $LL_{centers} = LL_{centers} \cup \{ Cmm \}$
15: Remove the SMMLCs
16: end if
17: end if
18: end for

Furthermore, we prune the mini-micro and micro leader clusters in the gap time in **Algorithm 3**, Puring Leader Clusters. In the pruning time, all the micro leader clusters and their Mini Micro Cluster Leaders are checked. Micro and mini-micro leader clusters are kept in the tree structure to make it easier for searching and updating. Based on different kinds of Mini Micro Cluster inside micro cluster different decisions are made for pruning, which are described as follows:

- All the mini-micro leader clusters are dense: micro leader cluster center is kept for the offline phase
- All the mini-micro leader clusters are sparse: mini micro leader clusters are removed as well as their micro leader cluster.
- Some of mini-micro leader clusters are dense and some of them are sparse:

1) Remove the sparse mini-micro leader clusters

2) Keep the center of the dense mini-micro leader clusters for the offline phase

4.2. Generating Final Clusters

The online phase maintains micro and mini-micro leaders clusters. However, we need to use a clustering algorithm to get the final clusters. When a clustering request arrives, DBSCAN algorithm is used on the micro and mini-micro leader cluster centers to get the final results. Each mini-micro and micro leader center is used as a virtual point to be used for clustering.

5. Experimental Evaluation

We implemented LeaDen-Stream in Massive Online Analysis (MOA)[1] [30] (**Figure 4**). In order to evaluate the clustering quality and scalability of the LeaDen-Stream algorithm, both real and synthetic data sets are

[1]http://moa.cms.waikato.ac.nz/

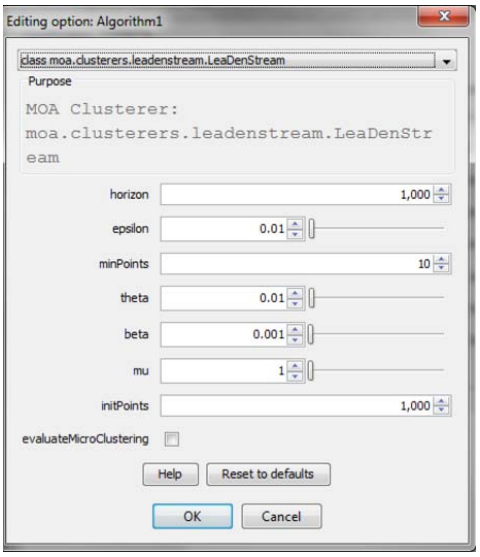

Figure 4. LeaDenStream in MOA.

used. The synthetic data set is depicted in **Figure 5**. The real data set is the KDD CUP99 Network Intrusion Detection data set (all 34 continuous attributes out of the total 42 available attributes are used). Using MOA framework, the clustering quality of LeaDen-Stream algorithm is evaluated and compared with CluStream and Den-Stream based on purity [31]. The efficiency is measured by the execution time. The quality of LeaDen-Stream is higher than CluStream with lower execution time. The LeaDen-Stream clustering quality is equal to DenStream while it runs faster than DenStream.

6. Conclusion

In this paper, we have proposed LeaDen-Stream, an algorithm for density-based clustering of evolving data stream using leader clustering. The algorithm runs in two phases. The method determines data points for offline clustering based on the distribution of the data inside the micro leader clusters. If the data is uniformly distributed, it only sends the micro leaders' centers. However, if the data is non-uniformly distributed, instead of micro leader centers their dense mini-micro leader cluster centers are kept for the offline phase. The pruning strategy is designed to eliminate the sparse mini-micro and micro leader clusters and to keep the dense ones for the offline phase.

Mini-micro and micro leader clusters are used in terms of increasing cluster quality and decreasing the time complexity. Using more than one representative point in cases that some of the mini-micro leader clusters are dense and some sparse, improves the quality of clustering. On the other hand, in cases that all of the mini-micro leader clusters are dense, sending only the micro leader cluster's center is enough for the offline phase, which in

Figure 5. Synthetic data set.

turn saves the time complexity.

Experimental results on a real-world data set as well as a synthetic data validates the design goals and shows that LeaDen-Stream significantly improves over DenStream and Clustream in terms of both clustering quality and time. As a future work, we want to automate the parameters of LeaDen-Stream and examine our algorithm in a sliding window model.

REFERENCES

[1] J. Han, M. Kamber and J. Pei, "Data Mining: Concepts and Techniques," 3rd Edition, Morgan Kaufmann Publishers Inc., San Francisco, 2011.

[2] L. OCállaghan, A. O. Meyerson, N. Mishra and S. Guha, "Streaming Data Algorithms for High-Quality Clustering," *International Conference on Data Engineering*, Los Alamitos, IEEE Computer Society, 2002, pp. 685-694.

[3] S. Guha, A. Meyerson, N. Mishra, R. Motwani and L. O'Callaghan, "Clustering Data Streams: Theory and Practice," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 15, No. 3, 2003, pp. 515-528. http://dx.doi.org/10.1109/TKDE.2003.1198387

[4] D. Barbará, "Requirements for Clustering Data Streams," SIGKDD Explor. Newsl.3, 2002, pp. 23-27. http://dx.doi.org/10.1145/507515.507519

[5] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, "A Framework for Clustering Evolving Data Streams," *Proceedings of the* 29*th International Conference on Very Large Data Bases*, VLDB Endowment, 2003, pp. 81-92.

[6] C. C. Aggarwal, "Data Streams—Models and Algorithms," Springer, 2007. http://dx.doi.org/10.1007/978-0-387-47534-9

[7] S. Guha, N. Mishra, R. Motwani and L. O'Callaghan, "Clustering Data Streams," *Proceedings of the* 41*st Annual Symposium on Foundations of Computer Science*, Washington DC, IEEE Computer Society, 2000, p. 359. http://dx.doi.org/10.1109/SFCS.2000.892124

[8] P. Kranen, I. Assent, C. Baldauf and T. Seidl, "The Clustree: Indexing Micro-Clusters for Anytime Stream Mining," *Knowledge Information System*, Vol. 29, No. 2, 2011, pp. 249-272. http://dx.doi.org/10.1007/s10115-010-0342-8

[9] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu and K. Zhang, "Density-Based Clustering of Data Streams at Multiple Resolutions," *ACM Transactions Knowledge Discovery*

Data, Vol. 3, No. 3, 2009, pp. 1-28. http://dx.doi.org/10.1145/1552303.1552307

[10] M. Ester, H. P. Kriegel, J. Sander and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proceedings of* 2*nd International Conference on Knowledge Discovery and Data Mining* (KDD), Portland, Oregon, AAAI Press, 1996, pp. 226-231.

[11] M. Ankerst, M. M. Breunig, H. P. Kriegel and J. Sander, "Optics: Ordering Points to Identify the Clustering Structure," SIGMOD Record, Vol. 28, 1999, pp. 49-60. http://dx.doi.org/10.1145/304181.304187

[12] A. Hinneburg and D. A. Keim, "An Efficient Approach to Clustering in Large Multimedia Databases with Noise," KDD, 1998, pp. 58-65.

[13] A. Amini, W. Teh Ying, M. R. Saybani and S. R. Aghabozorgi, "A Study of Density-Grid Based Clustering Algorithms on Data Streams," 8*th International Conference on Fuzzy Systems and Knowledge Discovery* (*FSKD*11), Shanghai, IEEE, 2011, pp. 1652-1656.

[14] T. Zhang, R. Ramakrishnan and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," In: J. Widom, Ed., *Proceedings of the* 1996 *ACM SIGMOD International Conference on Management of Data*, ACM Press, 1996, pp. 103-114.

[15] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, "A Framework for Projected Clustering of High Dimensional Data Streams," *Proceedings of the Thirtieth International Conference on Very Large Data Bases*, Vol. 30, VLDB, 2004, pp. 852-863.

[16] C. C. Aggarwal, J. Han, J. Wang and P. S. Yu, "On High Dimensional Projected Clustering of Data Streams," *Data Mining and Knowledge Discovery*, Vol. 10, 2005, pp. 251-273. http://dx.doi.org/10.1007/s10618-005-0645-7

[17] F. Cao, M. Ester, W. Qian and A. Zhou, "Density-Based Clustering over an Evolving Data Stream with Noise," *SIAM Conference on Data Mining*, 2006, pp. 328-339.

[18] A. Forestiero, C. Pizzuti and G. Spezzano, "A Single Pass Algorithm for Clustering Evolving Data Streams Based on Swarm Intelligence. Data Mining and Knowledge Discovery," 2011.

[19] A. Amini and W. Teh Ying, "A Comparative Study of Density-Based Clustering Algorithms on Data Streams: Micro-Clustering Approaches," In: S. I. Ao, O. Castillo and X. Huang, Eds., *Intelligent Control and Innovative Computing*, Vol. 110 of Lecture Notes in Electrical Engineering, Springer, 2012, pp. 275-287.

[20] A. Amini and W. Teh Ying, "Density Micro-Clustering Algorithms on Data Streams: A Review," *International Conference on Data Mining and Applications* (*ICDMA*), Hong Kong, 2011, pp. 410-414.

[21] P. Viswanath and R. Pinkesh, "l-dbscan: A Fast Hybrid Density Based Clustering Method," *Proceedings of the* 18*th International Conference on Pattern Recognition*, Vol. 01, ICPR'06, Washington DC, IEEE Computer Society, 2006, pp. 912-915.

[22] P. Viswanath and V. Suresh Babu, "Rough-Dbscan: A Fast Hybrid Density Based Clustering Method for Large Data Sets," *Pattern Recognition Letters*, Vol. 30, No. 16, 2009, pp. 1477-1488. http://dx.doi.org/10.1016/j.patrec.2009.08.008

[23] J. Gao, J. Li, Z. Zhang and P. N. Tan, "An Incremental Data Stream Clustering Algorithm Based on Dense Units Detection. Lecture Notes in Computer Science 3518," 2005.

[24] Y. Chen and L. Tu, "Density-Based Clustering for Real-Time Stream Data," *Proceedings of the* 13*th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, New York, ACM, 2007, pp. 133-142. http://dx.doi.org/10.1145/1281192.1281210

[25] A. Zhou, F. Cao, W. Qian and C. Jin, "Tracking Clusters in Evolving Data Streams over Sliding Windows," *Knowledge and Information Systems*, Vol. 15, 2008, pp. 181-214. http://dx.doi.org/10.1007/s10115-007-0070-x

[26] A. Amini and W. Teh Ying, "DENGRIS-Stream: A Density-Grid Based Clustering Algorithm for Evolving Data Streams over Sliding Window," *International Conference on Data Mining and Computer Engineering* (*ICDMCE*), Bangkok, 2012, pp. 206-210.

[27] L. Tu and Y. Chen, "Stream Data Clustering Based on Grid Density and Attraction," *ACM Transactions on Knowledge Discovery Data*, Vol. 3, No. 3, 2009, pp. 1-27. http://dx.doi.org/10.1145/1552303.1552305

[28] A. Amini and W. Teh Ying, "On Density-Based Clustering Algorithms over Evolving Data Streams: A Summarization Paradigm," *International Conference on Information Technology and Management Innovation* (*ICITMI*), Guangzhou, 2012, pp. 2234-2237.

[29] W. Ng and M. Dash, "Discovery of Frequent Patterns in Transactional Data Streams," *Transactions on Large-Scale Data- and Knowledge-Centered Systems II*, Vol. 6380 of Lecture Notes in Computer Science, Springer, Berlin/ Heidelberg, 2010, pp. 1-30.

[30] A. Bifet, G. Holmes, B. Pfahringer, P. Kranen, H. Kremer, T. Jansen and T. Seidl, "Moa: Massive Online Analysis, a Framework for Stream Classification and Clustering," *Journal of Machine Learning Research* (*JMLR*), Vol. 11, 2010, pp. 44-50.

[31] Y. Zhao and G. Karypis, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," *Machine Learning*, Vol. 55, 2004, pp. 311-333. http://dx.doi.org/10.1023/B:MACH.0000027785.44527.d6

