DMM-Stream: A Density Mini-Micro Clustering Algorithm for Evolving Data Streams

Amineh Amini, Hadi Saboohi, Teh Ying Wah, and Tutut Herawan

Faculty of Computer Science and Information Technology
University of Malaya (UM)
50603 Kuala Lumpur, Malaysia

Abstract. Clustering real-time stream data is an important and challenging problem. The existing algorithms have not considered the distribution of data inside micro cluster, specifically when data points are non uniformly distributed inside micro cluster. In this situation, a large radius of micro cluster has to be considered which leads to lower quality. In this paper, we present a density-based clustering algorithm, DMM-Stream, for evolving data streams. It is an online-offline algorithm which considers the distribution of data inside micro cluster. In DMM-Stream, we introduce mini-micro cluster for keeping summary information of data points inside micro cluster. In our method, based on the distribution of the dense areas inside the micro cluster at least one representative point, either micro cluster itself or its mini-micro clusters' centers, are sent to the offline phase. By choosing a proper mini-micro and micro center, we increase cluster quality while maintaining the time complexity. A pruning strategy is also used to filter out the real data from noise by introducing dense and sparse mini-micro and micro cluster. Our performance study over real and synthetic data sets demonstrates effectiveness of our method.

Keywords: Density-based Clustering, Micro Cluster, Mini-Micro Cluster

1 Introduction

Recently, a huge amount of data have been generated from various real-time applications such as monitoring environmental sensors, social networks, sensor networks, and cyber-physical systems [14]. 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 [17, 12, 6, 2, 1]. However, clustering in data stream environment needs some special requirements due to data stream characteristics such as clustering in limited memory and time with single pass over the evolving data streams [13, 15, 19].

Traditional clustering algorithms can not deal with evolving data streams. In last few years, many data stream clustering algorithms have been proposed [20, 2, 5, 3, 4, 7, 8, 10, 19, 13, 12].

T. Herawan et al. (eds.), *Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013)*, Lecture Notes in Electrical Engineering 285, DOI: 10.1007/978-981-4585-18-7_76,

In this paper, we propose a density-based clustering algorithm over evolving data streams which considers distribution of data inside micro clusters. The algorithm, named as DMM-Stream, uses fading window model to deal with cluster evolution. DMM-Stream introduces the mini-micro cluster concept which is similar to micro cluster with a smaller radius (which was introduced in DenStream [10]). Our algorithm's features are described as follows. DMM-Stream:

- Considers the distribution of data points inside micro cluster.
- Increases quality by sending multiple representative points to the offline phase.
- Introduces dense and sparse mini-micro clusters to recognize real data from noise.
- Uses mahalanobis distance instead of Euclidean distance for identifying correct cluster center. This increases the quality of clustering as well.

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 introduces basic definitions. In Section 4, we explain in details the DMM-Stream algorithm. Section 5 shows our experimental results. We conclude the paper in Section 6.

2 Related Work

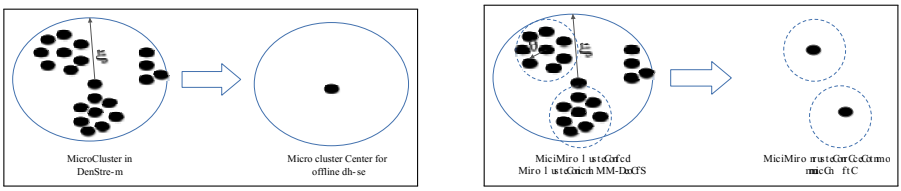
Most of the clustering algorithms over evolving data streams have two phases that was firstly introduced by CluStream [2]. CluStream has online and offline phases. The online phase keeps summary information, and the offline phase generates clusters based on synopsis information. However, CluStream, which is based on the k-means approach, finds only spherical clusters. Density-based clustering can overcome this limitation. Therefore, density-based clustering is extended in two-phase clustering [11, 10, 19, 18].

DenStream [10] is a clustering algorithm for evolving data stream. The algorithm extends the micro cluster [2] concept, and introduces the outlier and potential micro clusters to distinguish between real data and the outliers. DenStream 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 [19] 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 [11] is a density grid-based algorithm in which the data points are mapped to their corresponding grids and the grids are clustered based on the density. It uses a multi-resolution approach to cluster analysis.

We compare time complexity and clustering quality of D-Stream, DenStream and MR-Stream algorithms. In terms of time complexity, D-Stream has the lowest time complexity; however, it has low quality since the clustering quality



(a) DenStream (b) DMM-Stream

Fig. 1: Distribution of data inside microcluster

depends on the granularity of the lowest level of the grid structure. DenStream has a higher time complexity in comparison with 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.

We present a new method in which the granularity of micro cluster based on the distribution of data inside it by introducing new concepts called mini-micro cluster and micro cluster. This has not been considered in any of aforementioned algorithms. For example, in DenStream only the center of potential micro clusters are sent to offline phase. However, if the data points are not distributed uniformly inside micro cluster, sending only one representative point for each micro cluster leads to a low accuracy. Therefore, using mini-micro cluster increases the quality and using micro cluster prevents high time complexity (Figure 1, in which ϵ and θ are the radius of mini-micro and micro clusters respectively). We also use 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 DMM-Stream algorithm.

Definition 1. ***The Decaying Function****: The fading function [16] used in DMM-Stream is 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. ***MiniMicroCluster (mmc)****: A mmc for a group of data points with time stamp at time t is defined as* $\{CF^1, CF^2, W_{mmc}, C_{mmc}, r\}$*.* $CF^1 = \sum_{j=1}^{n} f(t - T_j)p_j$*: weighted linear sum of the points,*$CF^1 = \sum_{j=1}^{n} f(t - T_j)p_j$*: weighted linear sum of the points,* $W_{mmc} = \sum_{j=1}^{n} f(t-T_j)$*: mmc weight,* $C_{mmc} = \frac{CF^1}{W_{mmc}}$*: mmc center,*$r_{mmc} = \sqrt{\frac{CF^2}{W_{mmc}} - (\frac{CF^1}{W_{mmc}})^2}$*: mmc radius,* $r_{mmc} < \theta$*.*

Definition 3. ***MicroCluster (mc)****: A mc for a number of its mini-micro is defined as* $mc = \{\{mmc\}, W_{mc}, C_{mc}, r_{mc}\}$*.* $\{\{mmc_0, mmc_1, \ldots, mmc_n\}\}$ *mini-micro Clusters list,* $W_{mc} = \sum_{i=1}^{|\{mmc\}|} W_{mmc}$ *Microcluster weight,*
$C_{mc} = \frac{\sum_{i=1}^{|\{W_{mc}\}|} C_{mmc}}{|\{mmc\}|}$ *MicroCluster radius,* $r_{mmc} < \epsilon$ *MicroCluster radius.*

Definition 4. ***DenseMiniMicroCluster(DMMC)****: a mmc with a weight more than maximum threshold.* $W_{mm} > \frac{h_{mm}}{1-2^{-\lambda}} = D_{mm}$

Definition 5. ***SparseMiniMicroCluster(SMMC)****: a mmc with a weight less than maximum density threshold.* $W_{mm} \leq \frac{k_{mm}}{1-2^{-\lambda}} = S_{mm}$

h_{mm} and k_{mm} are controlling the threshold since the density cannot exceed $\frac{1}{1-2^{-\lambda}}$ (according to Lemma 1).

Definition 6. ***DenseMicroCluster(DMC)****: a mc which any of its* $\{mmc\}$ *is dense.* $DMC = \{\{DMMC\}\}$

Definition 7. ***SparseMicroCluster(SMLC)****: a mc which all its mini-micro clusters are sparse.*$SMC = \{\{SMMC\}\}$

Definition 8. ***CenterList****: set of centers of* $DMMC$ *and* DMC *which are sent to the offline phase:* $CenterList = \{DMMC\} \cup \{DMC\}$

Definition 9. ***MiniMicroCluster(MMC) Maintenance****: if we have a* MMC *at a time* t *and a point* p *arrives in* $t+1$ *then the statistics become* $MMC_{t+1} = \left\{2^{-\lambda}.CF^1 + p, 2^{-\lambda}.W_{mm} + 1\right\}$

Lemma 1. *The maximum weight of the mini-microCluster(mmc) is* $\frac{1}{1-2^{-\lambda}}$

Proof. If we assume that the data point in the data stream is added to the same mini-microCluster(mmc), the weight is equal to $W_{mmc} = \sum_{t'}^{t} 2^{-\lambda(t-t')}$ which can be converted to the following equation: $W_{mmc} = \sum_{t'}^{t} 2^{-\lambda(t-t')} = \frac{1-2^{-\lambda(t+1)}}{1-2^{-\lambda}}$ the maximum weight is defined when $t \to \infty$ therefore the maximum is defined as follows: $W_{mmc_{maximum}} = \frac{1}{1-2^{-\lambda}}$

Lemma 2. *The minimum time for converting the DMMC to SMMC and vise versa is:* $t_{min} = \log_{\lambda}^{(\frac{S_{mmc}}{D_{mmc}})}$

Proof. proof is shown in [10] and [11].

4 DMM-Stream Clustering Algorithm

We now describe the key components of DMM-Stream outlined in Algorithm 1. When a new data record x arrives, we add it to the mini-micro or micro cluster depending on the distribution of data in merging algorithm. Then, we periodically prune in every gap time (which is the minimum time for converting a dense mini-micro cluster to sparse mini-micro cluster and vise versa). We remove the sparse mini micro and micro clusters in pruning algorithm.

Our clustering algorithm is divided into two phases: a) Online phase: keeping mini-micro and Micro clusters, b) Offline phase: generating final clusters.

Algorithm 1 DMM-Stream(DS, ϵ, θ)

```
Input: a data stream
Output: arbitrary shape clusters
t=0;
while not end of stream do
    Read data point x from Data Stream
    Merge(x,ϵ,θ);
    if t mod t_min == 0 then
        Pruning(MMC,MC);
    end if
    t=t+1;
end while
if the clustering request is arrived then
    Generate clusters
end if
```

4.1 Keeping mini-micro and Micro clusters

When a data point is arrived from data streams. The procedure is described as follows (Algorithm 2: Merge):

1. we try to find the nearest micro cluster to the data point
2. if we find such a micro cluster we try to find nearest mini-micro cluster to the data point. if there is such a mini-micro cluster then merge the data point to the nearest mini-micro cluster, otherwise form a new mini-micro cluster with x as a center of new mini-micro cluster.
3. otherwise, if there is not such micro cluster, form a new micro cluster with x as a center of new micro cluster.

Furthermore, we prune the mini-micro and micro cluster in the gap time in Algorithm 3: Pruning. In the pruning time, all the micro clusters are checked. We keep the list of micro and mini-micro cluster in the tree structure to make it more easier for searching and updating. For each micro cluster, its mini-micro cluster lists are checked.

We have three different situation for the mini-micro cluster list:

- if all of the mini-micro clusters are dense: micro cluster's center is kept for offline phase
- if all of the mini-micro clusters are sparse: mini micro clusters are removed as well as the micro cluster.
- if some of the mini-micro clusters are dense and some of them are sparse:
  - remove the sparse mini-micro clusters
  - keep center of the dense mini-micro clusters for offline phase

Initialization: we apply DBSCAN algorithm to the first initial points to initialize the online phase. we initialize the group of micro and mini-micro clusters by scanning data points. For each data point, if the total weight in its θ neighborhood is above its threshold , then we create a mini-micro clusters and

Algorithm 2 Merge(x, ϵ, θ)

Input: a data point from data stream
Output: list of MicroClusters MC with their MiniMicros
$mc = \left\{\left\{mmc_0^1, mmc_1^1, \ldots, mmc_n^1\right\}, ..., \left\{mmc_0^n, mmc_1^n, \ldots, mmc_n^n\right\}\right\}$
find the nearest microcluster center C_{mc} to x
if $Distance(x, C_{mc}) < \epsilon$ **then**
    find the nearest mmc center C_{mmc} to x
    if $distance(x, C_{mmc}) < \theta$ **then**
        Merge x to the mmc;
    else
        create a new mmc with x;
    end if
else
    create a new mc by x
end if

remove data point from data point list. furthermore, we check the aggregation of mini-micro cluster weights in the ϵ neighborhood of a microcluster. If it is above its threshold then a micro cluster is formed for these minis.

4.2 Generating final clusters

The online phase maintained micro and mini-micro 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 cluster centers to get the final results. Each mini-micro and micro cluster center is used as a virtual point to perform clustering.

5 Experimental Evaluation

We implemented DMM-Stream in MOA [9]. We use the KDD CUP'99 Network Intrusion Detection data set and compare the performance and quality of the DMM-Stream with DenStream. The efficiency is measured by the execution time. The clustering quality is evaluated by the average purity of clusters. We make different sizes of data set from KDD CUP'99, and evaluate the clustering quality. In most of the cases the quality is better than DenStream. However the best answer is the KDD Cup 99 sub dataset with 17843 numerical records with 6 different classes. Our result only 1% improved the cluster quality with same time complexity in DenStream. We are trying to improve purity more than this value with bigger data set. We also evaluate our algorithm on the simple synthetic data set with arbitrary shapes. In this situation we get better purity but with higher time complexity.

6 Conclusion

In this paper, we have proposed DMM-Stream, an algorithm for density-based clustering of evolving data stream. The algorithm has two phases. The method

Algorithm 3 Pruning($\{mmc\}, \{mc\}$)

Input: list of MiniMicroClusters and MicroClusters$\{MMC\}, \{MC\}$
$mc = \left\{\left\{mmc_0^1, mmc_1^1, \ldots, mmc_n^1\right\}, \ldots, \left\{mmc_0^n, mmc_1^n, \ldots, mmc_n^n\right\}\right\}$
Output: Center List$\{CL_{centers}\}$
for all microclusters $\{mc\}$ **do**
  check all its minimicros $\{mmc\}$;
  $\{mmc_{initial}\} = \{mmc\}$
  for each mmc **do**
    if mmc is sparse **then**
      remove mmc from its mc list
    end if
  end for
  if $\{mmc\} = \{\}$ **then**
    remove its related microcluster mc
  end if
  if $\{mmc_{initial}\} = \{mmc\}$ **then**
    add mc center C_{mc} to center list
    $CenterList = CenterList \cup \{C_{mc}\}$
  else
    add all the mmc center C_{mmc} to the CenterList
    $CenterList = CenterList \cup \{C_{mmc}\}$
  end if
end for

determines the centers for offline clustering based on the distribution of the data inside the micro clusters. If the data is uniformly distributed, it only sends the micro cluster centers. However, if the data is non uniformly distributed instead of micro cluster center, its dense mini-micro cluster centers are kept for the offline phase. The pruning strategy is designed to delete the sparse mini-micro and micro clusters and to keep the dense one for the offline phase. Mini-micro and micro clusters are used in terms of increasing cluster quality and decreasing the time complexity. As a future work we want to automate the parameters of DMM-Stream and examine our algorithm in a sliding window model.

Acknowledgement

This paper is supported by High Impact Research (HIR) Grant, University of Malaya No UM.C/625/1/HIR/196.

References

1. Aggarwal, C.C. (ed.): Data Streams – Models and Algorithms. Springer (2007)
2. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th international conference on Very large data bases. pp. 81–92. VLDB Endowment (2003)
3. Amini, A., Teh Ying, W.: Density micro-clustering algorithms on data streams: A review. In: International Conference on Data Mining and Applications (ICDMA). pp. 410–414. Hong Kong (2011)

4. Amini, A., Teh Ying, W.: A comparative study of density-based clustering algorithms on data streams: Micro-clustering approaches. In: Ao, S.I., Castillo, O., Huang, X. (eds.) Intelligent Control and Innovative Computing, Lecture Notes in Electrical Engineering, vol. 110, pp. 275–287. Springer US (2012)
5. Amini, A., Teh Ying, W.: DENGRIS-Stream: A density-grid based clustering algorithm for evolving data streams over sliding window. In: International Conference on Data Mining and Computer Engineering (ICDMCE). pp. 206–210. Bangkok, Thailand (2012)
6. Amini, A., Teh Ying, W.: Requirements for clustering evolving data stream. In: 2nd International Conference on Power Electronics, Computer and Mechanical Engineering (ICPECME). Cambodia (2013)
7. Amini, A., Teh Ying, W., Saybani, M.R., Aghabozorgi, S.R.: A study of density-grid based clustering algorithms on data streams. In: 8th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD11). pp. 1652–1656. IEEE, Shanghai (2011)
8. Amini, A., Wah, T.Y.: Adaptive density-based clustering algorithms for data stream mining. In: Third International Conference on Theoretical and Mathematical Foundations of Computer Science. pp. 620–624. IERI (2012)
9. Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., Seidl, T.: Moa: Massive online analysis, a framework for stream classification and clustering. In: Journal of Machine Learning Research (JMLR). vol. 11, pp. 44–50 (2010)
10. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: SIAM Conference on Data Mining. pp. 328–339 (2006)
11. Chen, Y., Tu, L.: Density-based clustering for real-time stream data. In: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 133–142. KDD '07, ACM, New York, NY, USA (2007)
12. Guha, S., Meyerson, A., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams: Theory and practice. IEEE Transactions on Knowledge and Data Engineering 15(3), 515–528 (June 2003)
13. Guha, S., Mishra, N., Motwani, R., O'Callaghan, L.: Clustering data streams. In: Proceedings of the 41st Annual Symposium on Foundations of Computer Science. p. 359. IEEE Computer Society, Washington, DC, USA (2000)
14. Han, J., Kamber, M., Pei, J.: Data Mining: Concepts and Techniques Third edition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (2011)
15. Kranen, P., Assent, I., Baldauf, C., Seidl, T.: The clustree: indexing micro-clusters for anytime stream mining. Knowl. Inf. Syst. 29(2), 249–272 (2011)
16. Ng, W., Dash, M.: Discovery of frequent patterns in transactional data streams. In: Transactions on Large-Scale Data- and Knowledge-Centered Systems II, Lecture Notes in Computer Science, vol. 6380, pp. 1–30. Springer Berlin / Heidelberg (2010)
17. OĆallaghan, L., Meyerson, A., Motwani, R., Mishra, N., Guha, S.: Streaming-data algorithms for high-quality clustering. In: International Conference on Data Engineering. pp. 685–694. IEEE Computer Society, Los Alamitos, CA, USA (2002)
18. Tu, L., Chen, Y.: Stream data clustering based on grid density and attraction. ACM Transactions on Knowledge Discovery Data 3(3), 1–27 (2009)
19. Wan, L., Ng, W.K., Dang, X.H., Yu, P.S., Zhang, K.: Density-based clustering of data streams at multiple resolutions. ACM Transactions Knowledge Discovery Data 3(3), 1–28 (2009)
20. Zhou, A., Cao, F., Qian, W., Jin, C.: Tracking clusters in evolving data streams over sliding windows. Knowledge and Information Systems 15, 181–214 (May 2008)