Jurnal Teknologi

Full Paper

IMPROVED DENSITY BASED ALGORITHM FOR DATA STREAM CLUSTERING

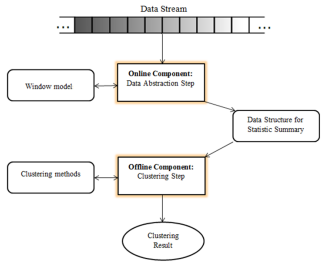
Maryam Mousavi*, Azuraliza Abu Bakar

Centre for Artificial Intelligence Technology, Faculty of Information Science and Technology, National University of Malaysia, 43600, Bangi, Selangor, Malaysia

Article history
Received
15 May 2015
Received in revised form
1 July 2015
Accepted
11 August 2015

*Corresponding author
maryam.mousavi2010@gmail.com

Graphical abstract



Abstract

In recent years, clustering methods have attracted more attention in analysing and monitoring data streams. Density-based techniques are the remarkable category of clustering techniques that are able to detect the clusters with arbitrary shapes and noises. However, finding the clusters with local density varieties is a difficult task. For handling this problem, in this paper, a new density-based clustering algorithm for data streams is proposed. This algorithm can improve the offline phase of density-based algorithm based on MinPts parameter. The experimental results show that the proposed technique can improve the clustering quality in data streams with different densities.

Keywords: Data streams; density-based clustering

1.0 INTRODUCTION

The process of data stream mining involves extracting valuable patterns in real time from dynamic streaming data in only a single scan, which can be very challenging. However, the process of data stream clustering has been the subject of much attention due to its effectiveness in data mining [1, 2]. Clustering involves processing data and partitioning the information or objects contained within it into subsets known as clusters. The aim of this process is to classify similar objects into the same cluster while objects in various clusters are dissimilar [3]. Essentially, clustering algorithms that are used to process huge data are basic methods that can be applied in data mining, pattern recognition, and machine learning. Streaming access performs better than random access for the huge volumes of data stored on hard disks or in data stream form, hence streaming algorithms are required to cluster such data [4]. However, due to the nature of the data stream, which is massive and evolves over time, traditional clustering techniques cannot be applied. Thus, it has become crucial to develop new and improved clustering techniques. Recently, various perspectives on and aspects of data stream clustering have been discussed and several algorithms and methods have been proposed. The existing clustering techniques are generally categorized into five major categories: hierarchical, partitioning, grid-based, density-based, and model-based [5].

Density-based techniques are the remarkable category in clustering data streams that possess quite a few significant advantages for data clustering such as i) the ability to detect arbitrary shaped clusters, ii) the ability to handle noises and iii) they require just the one time to scan raw data. Apart from that, such algorithms do not require prior knowledge of the number of clusters (*k*) unlike *k*-means algorithms that need to be given the number of clusters in advance [6, 7].

Among the existing density-based clustering methods for data stream, DenStream [8] is an outstanding algorithm. This algorithm is able to detect the clusters with arbitrary shapes and noises. However, one of the challenging problems of this algorithm is in detecting clusters with different densities where it can lead to reduce the clustering quality. For example, in Figure 1 DenStream can detect only one cluster instead of detecting three clusters with different densities.

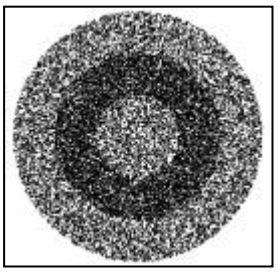


Figure 1 Clusters with different densities

To handle this problem, in this paper, a new density based clustering algorithm for data stream with various densities is proposed.

2.0 BACKGROUND

Density-based clustering techniques consider the dense areas of objects as clusters where they are separated with low density sparse areas in dataset. These techniques are able to recognize the clusters with arbitrary shapes and can handle noises. Also they do not require prior knowledge of the number of clusters.

A two-phase scheme density-based algorithm known as DenStream has been developed to cluster evolving data streams [8]. In the first phase, this algorithm uses the fading window model to create a synopsis of the data. Then, in the second phase, the synopsis of the data stored from first phase is utilized to provide the clustering results. This algorithm can handle arbitrary shaped clusters.

The D-Stream algorithm [9] has also been proposed, which is capable of making automatic and dynamic adjustments to the data clusters without user specification with regards to the target time horizon and number of clusters. This algorithm creates separated grids to map new incoming data. A decay factor is used with the density of each data point in order to determine which data are recent and which are less important (old). The D-Stream algorithm is incapable of processing very high-dimensional data; however, the DenStream algorithm has no difficulty in processing such data.

Similar to D-Stream, MR-Stream [10] creates cell partitions in the data space. Whenever a dimension is divided in half, a single cell goes through another division to form 2^d subcells, where d is the dimension of the data set. The division process can be set to a maximum limit by a user-defined parameter. The divided cells are stored on a quad tree structure that allows for data clusters to be created at different resolution levels. The MR-Stream algorithm allocates all new data into the appropriate cells at every time stamp interval during the online phase and also updates the summarized data.

OPCluStream [11] is another density-based algorithm for clustering data streams. This algorithm utilizes a tree topology for organizing points and directional pointers to link all related points together. This algorithm is able to detect arbitrary shaped clusters.

Although, the above-mentioned methods are able to cluster data streams efficiently, however they have low clustering quality when they deal with data space with different densities.

Here, some different density-based techniques to cluster multi-density data are described. ExDBSCAN [12] is a multi-density clustering algorithm that is based on greedy technique. This algorithm gets just one parameter, MinPts, and the value of radius increases gradually to take the real neighbourhood. This algorithm has high execution time that makes it unsuitable for data streams.

Another density-based clustering algorithm for streaming data is the DSCLU algorithm [13]. DSCLU uses microclusters to detect suitable clusters, focusing on localizing dominant microclusters on the basis of their neighbours' weights. It is able to detect clusters in multi-density environments but it works with same radius to form microclusters.

In [14] the authors proposed a new extension of DBSCAN algorithm for detecting of multi-density of data. This algorithm considers the different value for radius of clusters based on *k*-nearest neighbours curve. It has some problems to manage the variety of density within the same cluster and has high computational time.

3.0 DENSTREAM

DenStream [8] is a density-based data stream clustering algorithm that consists of two phases: online and offline. In the former phase the summary of data is created as the microclusters and in the latter phase the algorithm uses the synopsis of data that has been stored from first phase to create the final clusters which named macroclusters. Figure 2 shows a general framework of DenStream. This algorithm uses the fading window model for clustering data stream. In this model weight would be assigned for every data stream record on the basis of the fading function $f(t) = 2^{-\lambda t}$, where $\lambda > 0$,and more weights would be assigned to recent data as compared to old data. The overall weight of data stream is a fixed value W:

$$W = \frac{v}{1-2^{-\lambda}} \quad (1)$$

where v is the number of points arrived in one time unit (stream speed). The importance of historical data is decreased by assuming the higher values of λ.

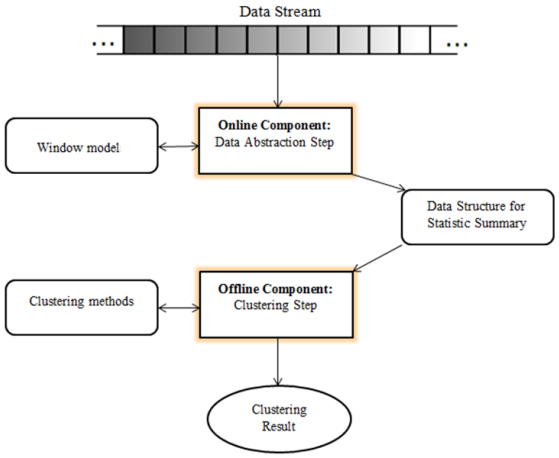


Figure 2 DenStream algorithm framework

Cao *et al*. expanded the notion of core point proposed in [15] and used the concept of microcluster for storing an approximate representation of data objects. A core point is a point in whose ε neighbourhood the overall weight of the data points is at least an integer μ.

They introduced three definitions of microclusers: core-microcluster, potential c-microcluster, and outlier microcluster.

Definition 1. core-microcluster: A core-microcluster or c-microcluster at time t for a group of near points $p_{i1}, \ldots, p_{in}$ with time stamps $T_{i1}, \ldots, T_{in}$ is determined as $CMC\ (w, c, r)$,where w is the weight, c is the center, and r is the radius of the c-microcluster. The weight is defined as:

$$w = \sum_{j=1}^{n} f(t - T_{ij}) \tag{2}$$

, the center is defined as:

$$c = \frac{\sum_{j=1}^{n} f(t - T_{ij}) p_{ij}}{w} \tag{3}$$

, and the radius is defined as:

$$r = \frac{\sum_{j=1}^{n} f(t - T_{ij})\, dist(p_{ij}, c)}{w} \tag{4}$$

, where $dist(p_{ij},c)$ is the Euclidean distance between the point p_{ij} and the center c.

It is noteworthy that a microcluster in order to be considered as a c-microcluster, its weight has to be above or equal to a predefined value of μ $(w \geq \mu)$ and also its radius has to be below or equal to ε $(r \leq \varepsilon)$.

Definition 2. potential c-microcluster: A potential c-microcluster or p-microcluster, at time t for a group of near points $p_{i1}, \ldots, p_{in}$ with time stamps $T_{i1}, \ldots, T_{in}$ is determined as $\{CF^1, CF^2, w\}$, where the weight of a p-microcluster w, as mentioned above, has to be above or equal to $\beta\mu$ $(w \geq \beta\mu)$ and β, $0 < \beta \leq 1$, is a parameter for determining the outlier threshold relative to c-microclusters. CF^1 is the weighted linear sum of the points and is defined as:

$$CF^1 = \sum_{j=1}^{n} f(t - T_{ij}) p_{ij} \tag{5}$$

and CF^2 is the weighed square sum of the points and is defined as:

$$CF^2 = \sum_{j=1}^{n} f(t - T_{ij}) p_{ij}^2 \tag{6}$$

the center of a p-microcluster is defined as:

$$c = \frac{CF^1}{w} \tag{7}$$

and the radius of a p-microcluster is defined as:

$$r = \sqrt{\frac{CF^2}{w} - \left(\frac{CF^1}{w}\right)^2} \tag{8}$$

Note that the radius of a p-microcluster has to be below or equal to ε $(r \leq \varepsilon)$.

Definition 3. outlier microcluster: An outlier microcluster or o-microcluster, at time t for a group of near points $p_{i1}, \ldots, p_{in}$ with time stamps $T_{i1}, \ldots, T_{in}$ is determined as $\{CF^1, CF^2, w, t_0\}$. The definitions of weight (w), CF^1, CF^2, center (c), and radius (r) of an o-microcluster are the same as a p-microcluster . $t_0 = T_{i1}$ shows the generation of the o-microcluster. It is noteworthy that the weight of an o-microcluster has to be below $\beta\mu$ $(w < \beta\mu)$. Algorithm 1 illustrates the pseudocode of DenStream algorithm.

Algorithm 1 DenStream algorithm

1: $T_p = \left\lceil \frac{1}{\lambda} \log\left(\frac{\beta\mu}{\beta\mu - 1}\right) \right\rceil$;
2: Get the next point p at current time t from data stream DS;
3: Merging (p);
4: **if** (t mod T_p)=0 **then**
5: **for** each p-microcluster c_p **do**
6: **if** w_p (the weight of c_p) < $\beta\mu$ **then**
7: Delete c_p;
8: **end if**
9: **end for**
10: **for** each o-microcluster c_o **do**
11: $\xi = \frac{2^{-\lambda(t - t_0 + T_p)} - 1}{2^{-\lambda T_p} - 1}$;

12: **if** w_o (the weight of c_o)< ξ **then**
13: Delete c_o;
14: **end if**
15: **end for**
16: **end if**
17: **if** a clustering request arrives **then**
18: Generating clusters;
19: **end if**

4.0 PROPOSED METHOD

Since DenStream algorithm is not applicable for datasets with multi-density clusters, in this section, we propose a new density-based algorithm for solving this problem. The DenStream algorithm applies DBSCAN algorithm in its offline phase. Since this algorithm uses a global set of parameters, hence it is not able to detect clusters with different densities.

The proposed method works based on only one parameter (MinPts). When a clustering request arrives, this algorithm gets all the p-microclusters as virtual points with centers of c_p and weights w. For identifying the dense regions of data, all points must be classified to [n/MinPts] subsets called preliminary clusters (pre-clusters). First, the proposed algorithm considers all points as a single cluster and then calculates its center. Then the farthest point from the center is considered as a noise and deleted from the dataset. After removing this point, the center point has to be recalculated. This step has to be done iteratively until only the MinPts or less than it are left. These points form the first pre-cluster. The process of generation pre-clusters continues until all points are grouped into pre-cluster = [n/MinPts]. To create dense regions and final clusters, these pre-clusters have to be merged based on pairwise distance between the center points of pre-clusters. The pre-clusters that are close enough are merged and they form the final clusters. The process of this algorithm is shown in algorithm 2.

Algorithm 2 Proposed algorithm

Input: p-microclusters
Output: Final clusters
1: Create an empty list
2: D ← all centers of p-microclusters
3: **for** all points ∈ D **do**
4: Candidate ←{x ∈ D}
5: **while** |Candidate| > MinPts **do**
6: Center ←mean of Candidate
7: y = max distance (x, Center)
8: Candidate \ {y}
9: **end while**
10: Add Candidate to list
11: **end for**
12: **return** list
13: **for** all pre-clusters in the list **do**
14: Calculate the distances of pre-clusters' centers
15: Merge close pre-clusters
16: **end for**

5.0 EXPERIMENTAL EVALUATION

We implemented the proposed method in Microsoft visual C#. The network intrusion detection (KDD CUP 99) dataset was used as a real dataset to show the performance and the quality of proposed method compared to DenStream algorithm. In experiments, analogous to [8] all 34 continuous attributes out of the total 42 available attributes were utilized. The various subsets of network intrusion detection dataset were selected to calculate the purity. The parameters used in the experiment were chosen same as DenStream algorithm.

The clustering quality is calculated by the average purity of clusters that is described as below:

$$purity = \frac{\sum_{i=1}^{K} \frac{|C_i^d|}{|C_i|}}{K} \times 100\% \quad (9)$$

where K is the number of clusters, $|C_i^d|$ is the number of points with the dominant class label in cluster i and $|C_i|$ shows the number of points in cluster i.

The comparison between proposed method and DenStream on the KDD CUP 99 dataset is illustrated in Figure 3. The time units selected in the experiments are same as those adopted by DenStream because in these time points some attacks occur. In the Figure 3, the horizon and the stream speed (the number of points per time unit) have been set to 1 and 1000, respectively.

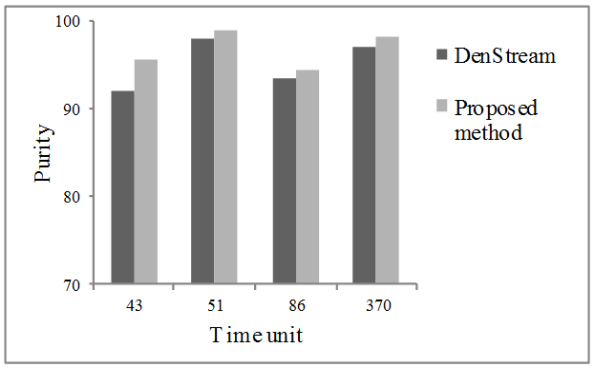


Figure 3 Clustering quality for KDD CUP 99 dataset with horizon=1 and stream speed=1000

The experimental results showed that in the most of the cases the proposed algorithm outperformed DenStream algorithm in terms of clustering quality in data streams.

6.0 CONCLUSION

The major research field in data stream mining is to develop efficient methods to mine the data stream. However, the mining task is complicated because of the specific characteristics of the data stream. The data stream clustering approach is one of the data mining techniques that can extract knowledge from such data.

Density-based algorithms are the outstanding category in clustering data streams that are able to recognize the clusters with arbitrary shapes and can handle noises. Also they do not require prior knowledge of the number of clusters. However, one of the challenging problems of these algorithms is in detecting clusters with different densities where it can lead to reduce the clustering quality. In this paper, for solving this problem, a new density-based clustering algorithm for data stream with various densities was proposed. The experimental results showed that the proposed algorithm can improve the clustering quality in data streams with variant densities.

Acknowledgment

This work is supported by Ministry of Higher Education (MOHE) Malaysia under the projects with the codes of "FRGS/1/2013/ICT02/UKM/01/1" and "ERGS/1/2012/STG07/UKM/01/1".

References

[1] J. A. Silva, E. R. Faria, R. C. Barros, E. R. Hruschka, A. C. d. Carvalho, and J. Gama. 2013. Data Stream Clustering: A Survey. *ACM Computing Surveys (CSUR)*. 46: 13.

[2] S. Ding, F. Wu, J. Qian, H. Jia, and F. Jin. 2013. Research on Data Stream Clustering Algorithms. *Artificial Intelligence Review*. 1-8.

[3] A. Madraky, Z. A. 2014. Othman, and A. R. Hamdan, "Analytic Methods for Spatio-Temporal Data in a Nature-Inspired Data Model. *International Review on Computers and Software (IRECOS)*. 9: 547-556.

[4] M. R. Ackermann, M. Märtens, C. Raupach, K. Swierkot, C. Lammersen, and C. Sohler. 2012. StreamKM++: A Clustering Algorithm for Data Streams. *Journal of Experimental Algorithmics (JEA)*. 17: 2.4.

[5] J. Han, M. Kamber, and J. Pei. 2006. *Data Mining: Concepts and Techniques*. Morgan kaufmann.

[6] H.-L. Nguyen, Y.-K. Woon, and W.-K. Ng. 2014. A Survey on Data Stream Clustering and Classification. *Knowledge and Information Systems*. 1-35.

[7] W.-K. Loh and Y.-H. Park. 2014. A Survey on Density-Based Clustering Algorithms. In *Ubiquitous Information Technologies and Applications*. ed: Springer. 775-780.

[8] F. Cao, M. Ester, W. Qian, and A. Zhou. 2006. Density-based Clustering Over an Evolving Data Stream with Noise. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. 328-339.

[9] L. Tu and Y. Chen. 2009. Stream Data Clustering Based on Grid Density and Attraction. *ACM Transactions on Knowledge Discovery from Data (TKDD)*. 3: 12.

[10] L. Wan, W. K. Ng, X. H. Dang, P. S. Yu, and K. Zhang. 2009. Density-based Clustering of Data Streams at Multiple Resolutions. *ACM Transactions on Knowledge Discovery from Data (TKDD)*. 3: 14.

[11] H. Wang, Y. Yu, Q. Wang, and Y. Wan. 2012. A density-based clustering structure mining algorithm for data streams. In *Proceedings of the 1st International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications*. 69-76.

[12] A. Ghanbarpour and B. Minaei. 2014. EXDBSCAN: An Extension of DBSCAN to Detect Clusters in Multi-Density Datasets. In *Intelligent Systems (ICIS), 2014 Iranian Conference on*. 1-5.

[13] A. Namadchian and G. Esfandani. 2012. DSCLU: a new Data Stream CLUstring algorithm for multi density environments. In *Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on*. 83-88.

[14] S. Louhichi, M. Gzara, and H. Ben Abdallah. 2014. A Density Based Algorithm for Discovering Clusters with Varied Density. in *Computer Applications and Information Systems (WCCAIS), 2014 World Congress on*. 1-6.

[15] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.