A Different Approach for Pruning Micro-clusters in Data Stream Clustering

Argenis A. Aroche-Villarruel[1(✉)], José Fco. Martínez-Trinidad[1], Jesús Ariel Carrasco-Ochoa[1], and Airel Pérez-Suárez[2]

[1] Computer Science Department, Instituto Nacional de Astrofísica, Óptica y Electrónica, Luis Enrique Erro No. 1, Sta. María Tonantzintla, 72840 Puebla, Mexico
{argenis, fmartine, ariel}@inaoep.mx

[2] Advanced Technologies Application Center, 7ma A #21406 e/214 y 216, Rpto. Siboney, Playa, 12200 Havana, Cuba
asuarez@cenatav.co.cu

Abstract. DenStream is a data stream clustering algorithm which has been widely studied due to its ability to find clusters with arbitrary shapes and dealing with noisy objects. In this paper, we propose a different approach for pruning micro-clusters in DenStream. Our proposal unlike other previously reported pruning, introduces a different way for computing the micro-cluster radii and provides new options for the pruning stage of DenStream. From our experiments over public standard datasets we conclude that our approach improves the results obtained by DenStream.

Keywords: Clustering · Data streams · Data mining

1 Introduction

Nowadays many applications produce large volumes of information in short time intervals (data streams). A data stream is a potentially infinite sequence of objects $x_1, x_2, \ldots, x_k, \ldots$ with timestamps $T_1, T_2, \ldots, T_k, \ldots$, where each object x_i from the data stream is a vector of features containing m dimensions, denoted as $x_i = \left(x_i^1, x_i^2, \ldots, x_i^m\right)$. Processing data streams has become a current research topic in many fields of computer science, with different applications such as: intrusion detection in networks [1], observation of environments [2], medical systems [3, 4], stock exchange analysis [5], social network analysis [6], object tracking in video [7], among others.

A technique widely used for processing data streams is clustering. Due to the nature of a data stream, there are some constraints that a clustering algorithm has to consider in order to process a data stream. These constraints are: (1) the amount of data in the stream must be considered as infinite, (2) the time for processing a single object is limited, (3) memory is limited, (4) there are objects that may be noisy and (5) the data distribution may evolve. In addition, there are other problems, inherent to the clustering problem, that must be addressed: the number of clusters in the dataset is unknown, clusters can take arbitrary shapes and noise may affect the clustering process, among others.

STREAM [8], CluStream [9], Clus-Tree [10] and DenStream [1] are among the most relevant and cited clustering algorithms reported in the literature for processing data streams. Due to its ability to find clusters with arbitrary shapes and dealing with noisy objects, DenStream has been widely studied and some variants [2, 11–15] have been proposed. Despite the results achieved by these DenStream variants, they have some problems mainly in the pruning phase, because sometimes this phase takes long time to remove very old information and this affects the results for data streams which evolve rapidly and steadily. In addition, sometimes the pruning phase also removes information that could be useful in future cases, since there could be periods of inactivity in some kind of data.

Based on the above mentioned problems and considering the advantages of DenStream over other clustering algorithms, the aim of this paper is to propose solutions for some of the problems of DenStream, in order to improve the quality of its results. This paper is organized as follows: Sect. 2 describes the DenStream algorithm. Section 3 describes our proposal. Section 4 reports an experimental comparison of our proposal against DenStream. Finally, in Sect. 5 some conclusions and future research directions are presented.

2 DenStream

DenStream is a density based algorithm for clustering data streams [1], which considers a fading (Damped) window model. In this model, the weight of each object decreases exponentially through a fading function in dependence of the elapsed time t since it appears in the data stream. The exponential fading function is widely used in temporal applications where it is desirable to gradually discount the history of past behavior. The function used in DenStream is $f(t) = 2^{-\lambda t}, \lambda > 0$. The higher value for λ the lower importance of the data compared to more recent data.

DenStream uses three important concepts as the basis for its operation: core-micro-cluster (c-micro-cluster), potential core-micro-cluster (p-micro-cluster) and outlier micro-cluster (o-micro-cluster).

A c-micro-cluster at the time t for a set of nearby objects $x_{i1}, x_{i2}, \ldots, x_{in}$ with timestamps $T_{i1}, T_{i2}, \ldots, T_{in}$, is defined as a triplet $CMC(w, c, r)$, where

$w = \sum_{j=1}^{n} f\left(t - T_{ij}\right)$, is its weight

$c = \frac{1}{w}\sum_{j=1}^{n} f\left(t - T_{ij}\right)x_{ij}$, is the center

$r = \frac{1}{w}\sum_{j=1}^{n} f\left(t - T_{ij}\right)d\left(x_{ij}, c\right)$, is the radius

where $d\left(x_{ij}, c\right)$ is the Euclidean distance between the object x_{ij} and the center c. The weight of a micro-cluster determines the minimum number of objects that a micro-cluster should have, assuming that the relevance of these objects is modified according to the fading function. With this aim, DenStream requires that the weight of each micro-cluster be less or equal than a predefined threshold μ. The radius of a micro-cluster determines which objects will be added to the micro-cluster, depending on their distance to the center of the micro-cluster. For accomplishing this, DenStream requires that the radius of a micro-cluster be less or equal than a predefined threshold ϵ. Thus, if the addition of an object to a micro-cluster makes its radius greater than ϵ, the object is not included in this micro-cluster. Due to this constraint in the radius, the number of c-micro-clusters tends to be much larger than the real number of clusters. On the other hand, the minimum weight constraint produces a significantly smaller number of c-micro-clusters than the number of objects in the data stream (see Fig. 1).

During the evolution of a data stream, the clusters and the noise are in constant change thus the c-micro-clusters are gradually built as the data stream is processed. Because of this, following a similar idea as the one proposed in [16], in DenStream potential c-micro-clusters and outlier micro-clusters for incremental computation, are introduced.

A p-micro-cluster at time t. for a set of nearby objects $x_{i1}, x_{i2}, \ldots, x_{in}$ with time-stamps $T_{i1}, T_{i2}, \ldots, T_{in}$, is defined as a triplet $\left\{\overline{CF^1}, \overline{CF^2}, w\right\}$, where

$w = \sum_{j=1}^{n} f\left(t - T_{ij}\right), w \geq \beta\mu$, is the weight of the p-micro-cluster; being $\beta \in (0, 1]$ a parameter to determine the noise threshold relative to the c-micro-clusters.

$\overline{CF^1} = \sum_{j=1}^{n} f\left(t - T_{ij}\right) x_{ij}$, is the weighted linear sum of the objects in the p-micro-cluster.

$\overline{CF^2} = \sum_{j=1}^{n} f\left(t - T_{ij}\right) S\left(x_{ij}\right)$, is the weighted squared sum of the objects in the p-micro-cluster; where $S\left(x_{ij}\right) = \left(x_{ij1}^2, x_{ij2}^2, \ldots, x_{ijm}^2\right)$.

From $\overline{CF^1}$ and $\overline{CF^2}$, we can compute the center and radius of a p-micro-cluster, denoted as c *and* r, respectively, through the following expressions

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

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

An o-micro-cluster at the time t. for a set of nearby objects $x_{i1}, x_{i2}, \ldots, x_{in}$with timestamps $T_{i1}, T_{i2}, \ldots, T_{in}$ is defined as $\left\{\overline{CF^1}, \overline{CF^2}, w, T_o\right\}$, here $\overline{CF^1}, \overline{CF^2}$ are defined as in a p-micro-cluster. $T_o = T_{i1}$ is the creation time of the o-micro-cluster, which is used to determine its lifetime. However, in this case $w < \beta\mu$ since this parameter determines when an o-micro-cluster switches to a p-micro-cluster.

Both p-micro-clusters and o-micro-clusters can be maintained incrementally. For example, given a p-micro-cluster $c_p = \left(\overline{CF^1}, \overline{CF^2}, w\right)$. If no new objects have been added to c_p in a time interval δt, then the new representation for this micro-cluster will be $c_p = \left(2^{-\lambda\delta t} \cdot \overline{CF^1}, 2^{-\lambda\delta t} \cdot \overline{CF^2}, 2^{-\lambda\delta t} \cdot w\right)$. In the opposite case, if an object p is

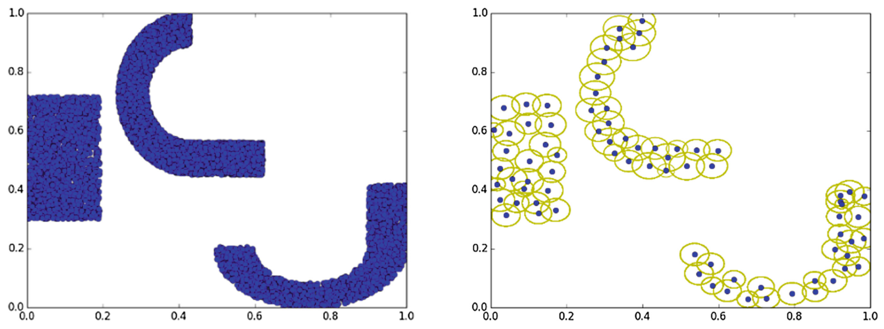

Fig. 1. Representation of a dataset using micro-clusters

added to c_p, then the micro-cluster will be represented as $c_p = \left(\overline{CF}^1 + p, \overline{CF}^2 + S(p), w + 1\right)$ where $S(p) = \left(p_1^2, p_2^2, \ldots, p_m^2\right)$. In a similar way the o-micro-clusters are updated.

The clustering algorithm DenStream is divided in two phases: an online phase for maintaining micro-clusters, and an offline phase to generate the final clustering; the offline phase is performed when the user requests it. DenStream has an initialization stage, where k objects are taken from the data stream and DBSCAN [17] is applied to detect k p-micro-clusters.

During the online phase, when an object x_i is retrieved from the data stream, the distance between x_i and each one of the centers of the p-micro-clusters is computed, the new radius r_j of the nearest p-micro-cluster is computed, if $r_j \leq \epsilon$ then x_i is added to that p-micro-cluster. Otherwise, the same process is done but with the o-micro-clusters, first we compute the distances and then the new radius r_j of the nearest o-micro-cluster is computed, if $r_j \leq \epsilon$ then x_i is added to the nearest o-micro-cluster, in this case it should be also determined if the nearest o-micro-cluster fulfills the constraints to become a p-micro-cluster. If x_i cannot be addedo any micro-cluster a new o-micro-cluster is generated from x_i.

In order to generate the final clustering, a variant of DBSCAN is applied. In this variant the radii of the p-micro-clusters and the distance between their centers allows determining if they are density reachable. Consider two p-micro-clusters with centers c_p and c_q and radii r_p and r_q. We say that these p-micro-clusters are density reachable if $d\left(c_p, c_q\right) \leq r_p + r_q$, where $d\left(c_p, c_q\right)$ is the Euclidean distance between c_p and c_q. Those p-micro-clusters that are density reachable are merged as a cluster.

Although the expression (2), for computing the radius of a micro-cluster, is reported in all variants of DenStream, in the practice the root argument can be negative, therefore this expression cannot be used. For this reason some of the algorithms that are extensions of DenStream [14, 15] are implemented within a Framework called MOA (Massive Online Analysis),[1] and they use a variant of the expression (2) to compute the radius in the micro-clusters. The expression used in MOA is:

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

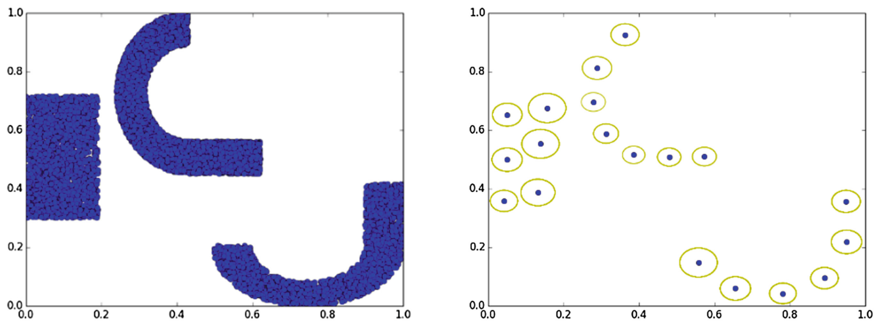

Fig. 2. p-micro-clusters obtained using the average of distances as radius.

$$r = \max_{1 \leq l \leq m} \left\{ \sqrt{\frac{\overline{CF^2}_l}{w} \left(\frac{\overline{CF^1}_l}{w} \right)^2} \right\} \tag{3}$$

In the expression (2) an approximation to the average of distances is computed, the expression (3) is similar to expression (2) but it only takes the maximum difference between the vector components instead of computing the Euclidean norm. Using the average of distances as the radius makes the radius smaller than it actually should be, which has a negative effect in the final clusters. This problem is illustrated in Fig. 2.

In the example of Fig. 2, the micro-clusters do not intersect their radii and therefore, when the DBSCAN variant is used to build the final clustering, those micro-clusters are not density reachable, and it results in many final clusters.

In MOA a constant is used to increase the size of micro-clusters radii. First the radius is computed then the product between the radius and that constant is computed. However, MOA does not explain how the value of this constant is obtained.

3 Our Proposal

In this section, we introduce some modifications of DenStream, which are depicted in Algorithm 1. In order to solve the problem of using the average of distances as radius in the micro-clusters, we propose to use as radius the distance from the center to the farthest object in the micro-cluster. In [1] this idea was previously proposed but the authors only report results over just one dataset. By using the distance from the farthest objects to the center, CF^2 can be replaced by the farthest object, the farthest object will be called x_f, also the timestamp T_f must be stored, since over time the relevance of the object also affects the radius. Considering this modification, we propose the following expression to compute the radius:

$$r = \sqrt{\left(\sum_{i=1}^{m} \left(c_i - x_{fi} \right)^2 \right) \times 2^{-\lambda \delta_t}} \tag{4}$$

where $\delta t = t - T_f$ is the difference between the current time and the time in which the farthest object x_f arrived; c_i is the i-th component of the micro-cluster center and, x_{fi} is the i-th component of the object x_f.

Using the proposed expression (4) allows to get more micro-clusters and they are less separated, which helps in the final phase to obtain a better clustering. Using the expression (4) adds to the clustering algorithm an extra computational cost, since each time that a new object is added into a micro-cluster, we have to verify if the new object is farther than the current x_f object.

In order to remove micro-clusters, DenStream considers the parameter λ and the elapsed time since the last change has occurred in each micro-cluster, such that those micro-clusters which have not been modified for a certain interval of time are removed.

Algorithm 1. Improved DenStream($DS, \epsilon, \beta, \mu, \lambda$)

```
01. $T_p = [\frac{1}{\lambda}\log(\frac{\beta\mu}{\beta\mu-1})$;  (value used for the pruning phase)
02. Get the object $x$ at current time $t$ from data stream $DS$;
03. Try to Merge $x$ into its nearest p-micro-cluster $c_p$;
04. if $x$ is not merged into $c_p$ then
05.   Try to Merge $x$ into its nearest o-micro-cluster $c_o$;
06.   if $x$ is not merged into $c_o$ then
07.     Create a new o-micro-cluster for $x$
08.   else
09.     if $w$ (the new weight of $c_o$) $> \beta\mu$ then
10.       Convert $c_o$ in a new p-micro-cluster;
11. if$(t \, mod \, T_p) = 0$ then (pruning phase)
12.   $\bar{x}_o = \frac{1}{n}\sum_{i=1}^{n} counter_i$;
13.   $\bar{x}_w = \frac{1}{n}\sum_{i=1}^{n} w_i$;
14.   for each o-micro-cluster $c_o$ do
15.     $\xi = \frac{2^{-\lambda(t-to+T_p)}-1}{2^{-\lambda T_p}-1}$;
16.     if $w_o(the\ weight\ of\ c_o) < \xi$ then
17.       Try to merge $c_o$ into its nearest p-micro-cluster $c_p$;
18.       if $c_o$ is not merged into $c_p$ then remove $c_o$
19.   for each p-micro-cluster $cp$ do
20.     if $w_p(the\ weight\ of\ c_p) < \bar{x}_w$ and $counter_p < \bar{x}_o$ then
21.       Convert $c_p$ in an o-micro-cluster;
22.       $counter_p = 0$;
23.   for each p-micro-cluster $cp$ do
24.     if $w_p(the\ weight\ of\ c_p) < \beta\mu$ then
25.       Convert $c_p$ in an o-micro-cluster;
26. if a clustering request arrives then
27.   Generate clusters;
```

Another problem with DenStream is that sometimes it takes too long time to remove those p-micro-clusters which do not receive objects frequently (depending on the value of λ). The above problem is solved by using a high value for λ but another problem arises, the radii of the micro-clusters decrease too fast and therefore, when a final clustering request is done, many clusters are generated. To solve this problem, we propose to consider those micro-clusters that have received too few objects in the latest updates as noise, i.e., they become o-micro-clusters. To determine when a p-micro-cluster have received too few objects, the average number of added objects in all p-micro-clusters is computed (step 12), also the average weight from all p-micro-clusters is computed (step 13); those p-micro-clusters whose weights and number of added objects are below of both averages become o-micro-clusters (steps 19–22).

Another change included in our proposal is that before removing an o-micro-cluster, we try to merge it with its nearest p-micro-cluster (steps 16–18); in this way we may retain certain information that could be useful in the future. In order to merge two micro-clusters the distance between them must be less than the radius of the p-micro-cluster, and the radius of the new micro-cluster must not be greater than ϵ. Finally, the last change in our proposal is that before removing p-micro-clusters they become o-micro-clusters (steps 23–25), it gives to them the opportunity to grow back, since as they were considered as relevants in previous windows, maybe they are in an inactivity period and they could receive new objects in the close future.

4 Experimental Analysis

For our experimental comparison, DenStream was in the same way as in the framework MOA. The assessment of the clustering results was done through *purity* measure which is defined in (5).

$$purity = \frac{1}{k}\sum\nolimits_{i=1}^{k} \frac{\left|c_i^d\right|}{\left|c_i\right|} \tag{5}$$

Where k is the number of clusters, $|c_i^d|$ denotes the number of objects with the most frequent label in the cluster i, and $|c_i|$ is the total number of objects in the cluster i. This evaluation is done on a predefined window, since objects are fading continuously.

Table 1 shows the characteristics of the datasets used in the experiments, KDD 99 Network Intrusion was used in [1] for evaluating DenStream, while Electricity, Forest Covertype and Poker-Hand datasets were suggested in MOA[2] to evaluate data stream clustering algorithms. In order to simulate a data stream in our experiments the objects are processed in the order that they are stored.

In our experiments all feature values were normalized in the range [0, 1] and in those datasets containing categorical features, we only use the numeric features. Different values were tested for the parameters of the algorithms and we used those in which both algorithms obtained the best results, which are: for Electricity $\epsilon = 0.12, \mu = 40$, for

[2] http://moa.cms.waikato.ac.nz/datasets/.

Table 1. Description of datasets used in our experiments

Name	#Objects	#Features	#Labels
Electricity	45,312	103	2
Network Intrusion (KDD 99)	494,020	41	23
Forest Covertype	581,012	54	7
Poker-Hand	829,201	11	10

KDD 99 $\epsilon = 0.15, \mu = 80$, for Forest Covertype $\epsilon = 0.16, \mu = 80$, for Poker-Hand $\epsilon = 0.17, \mu = 100$, in all datasets we use $\beta = 0.2$ *and* $\lambda = 0.25$. The window size was 10,000, and 1000 objects were used for the initialization phase. The results are shown in Fig. 3.

In the Fig. 3, we can see that our proposal performs better in most of the cases, except for Electricity, where DenStream performed better in 2 of the evaluated

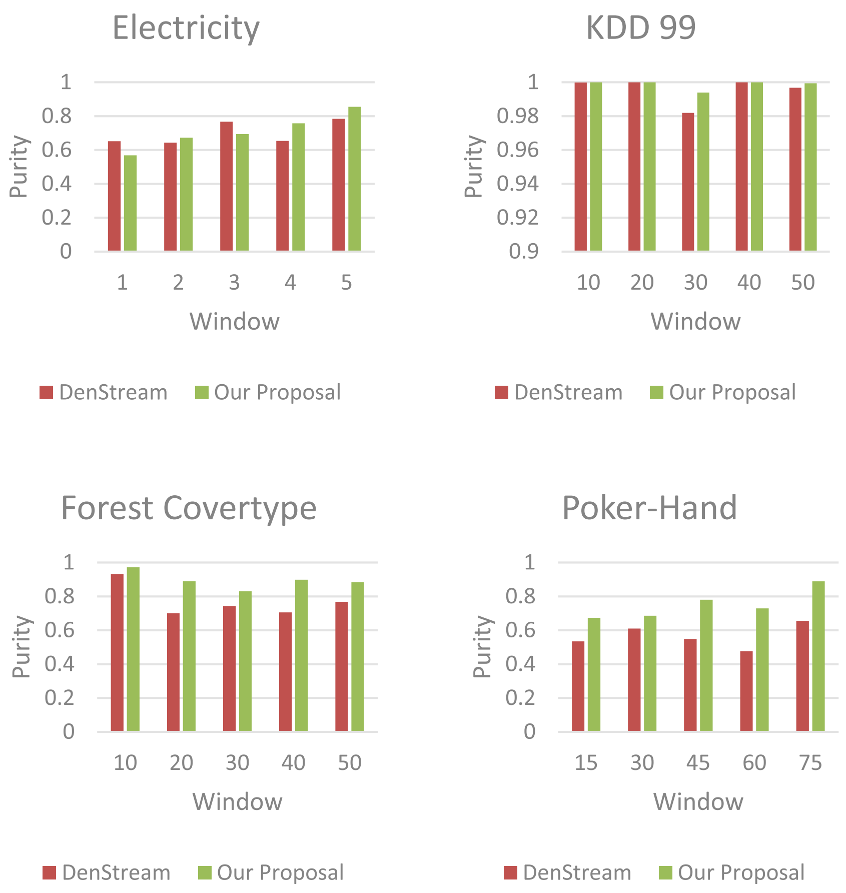

Fig. 3. Comparison of quality using purity

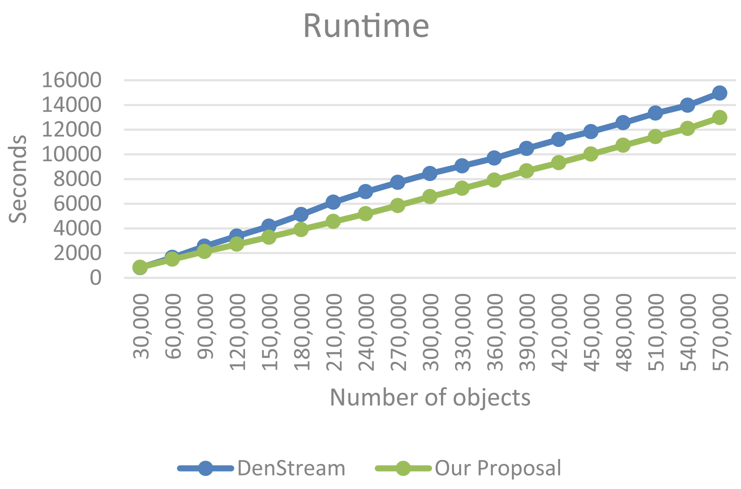

Fig. 4. Comparison of runtime in seconds

windows. For KDD 99 similar results were obtained with our proposal and DenStream, both algorithms perform well in all the windows. In Forest Covertype and Poker-Hand, our proposal preforms better in all the windows. From these experiments, we can conclude that the expression proposed for computing the radius and the changes proposed in the pruning phase of DenStream allows to improve the quality of the results provided by DenStream.

As it was previously mentioned in Sect. 3, getting the farthest object each time that a new object is added to a micro-cluster increases the computational cost of the clustering algorithm, however, in the practice there is not much difference in runtime, as it can be seen in Fig. 4, where we show a runtime comparison between our proposal and DenStream using the dataset Forest Covertype. It is important to comment that in our proposal since in the pruning phase we provide more options to change a p-micro-cluster into an o-micro-cluster, then more micro-clusters are removed and this reduces the amount of operations when we search the nearest p-micro-cluster.

5 Conclusions

In this paper a different approach to compute the radius of the micro-clusters in DenStream is proposed. Additionally, some pruning strategies are also included to remove useless micro-clusters. From our experiments we can conclude that the modifications proposed allows to improve the quality of the clustering results provided by DenStream, besides a similar runtime to that achieved by DenStream is maintained.

Since only the radius and center of micro-clusters are used to build the final clusters, and information like the weight and the lifetime of the micro-clusters are not used to build the final clustering, as future work, we propose to formulate a new strategy taking into account this additional information to build a better clustering in the final phase.

Acknowledgment. This work was partly supported by the National Council of Science and Technology of Mexico (CONACyT) through the project grant CB2008-106366; and the scholarship grant 362371.

References

1. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proceedings of the 6th SIAM International Conference on Data Mining, pp. 326–337. SIAM, Bethesda (2006)
2. Ruiz, C., Menasalvas, E., Spiliopoulou, M.: C-DenStream: using domain knowledge on a data stream. In: Gama, J., Costa, V.S., Jorge, A.M., Brazdil, P.B. (eds.) DS 2009. LNCS, vol. 5808, pp. 287–301. Springer, Heidelberg (2009)
3. Plant, C., Teipel, S.J., Oswald, A., Böhm, C., Meindl, T., Mourao-Miranda, J., Bokde, A.W., Hampel, H., Ewers, M.: Automated detection of brain atrophy patterns based on MRI for the prediction of Alzheimer's disease. NeuroImage **50**(1), 162–174 (2010)
4. Mete, M., Kockara, S., Aydin, K.: Fast density-based lesion detection in dermoscopy images. Comput. Med. Imaging Graph. **35**(2), 128–136 (2011)
5. Yang, D., Rundensteiner, E.A., Ward, M.O.: Summarization and matching of density-based clusters in streaming environments. Proc. VLDB Endow **5**(2), 121–132 (2011)
6. Lee, C.H.: Mining spatio-temporal information on microblogging streams using a density-based online clustering method. Expert Syst. Appl. **39**(10), 9623–9641 (2012)
7. Yu, Y., Wang, Q., Wang, X., Wang, H., He, J.: Online clustering for trajectory data stream of moving objects. Comput. Sci. Inf. Syst. **10**(3), 1293–1317 (2013)
8. Guha, S., Meyerson, A., Mishra, N., et al.: Clustering data streams: theory and practice. IEEE Trans. Knowl. Data Eng. **15**(3), 515–528 (2003)
9. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of VLDB (2003)
10. 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)
11. Lin, J., Lin, H.: A density-based clustering over evolving heterogeneous data stream. In: Proceeding of the 2nd International Colloquium on Computing, Communication, Control, and Management, pp. 275–277 (2009)
12. Ntoutsi, I., Zimek, A., Palpanas, T. et al.: Density-based projected clustering over high dimensional data streams. In: Proceedings of the 12th SIAM International Conference on Data Mining, pp. 987–998 (2012)
13. Forestiero, A., Pizzuti, C., Spezzano, G.: A single pass algorithm for clustering evolving data streams based on swarm intelligence. Data Min. Knowl. Disc. **26**(1), 1–26 (2013)
14. Hassani, M., Spaus, P., Gaber, M.M., Seidl, T.: Density-based projected clustering of data streams. In: Hüllermeier, E., Link, S., Fober, T., Seeger, B. (eds.) SUM 2012. LNCS, vol. 7520, pp. 311–324. Springer, Heidelberg (2012)
15. Amini, A., Saboohi, H., Wah, T.Y., Herawan, T.: DMM-Stream: a density mini-micro clustering algorithm for evolving data streams. In: Proceedings of the First International Conference on Advanced Data and Information Engineering (DaEng-2013), pp. 675–682. Springer (2014)

16. Zhang, T., Ramakrishnan, R., Livny, M.: BIRCH: an efficient data clustering method for very large databases. In: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data, pp. 103–114 (1996)
17. Ester, M., Kriegel, H., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: Proceeding of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)