Online Data Stream Clustering: Proposal

Nathalie A Barbosa, Louise Travé-Massuyès, Victor H Grisales

Online Data Stream Clustering: Proposal

Nathalie A. Barbosa [*,**] **Louise Travé-Massuyès** [**]
Victor H. Grisales [***]

[*] *Universidad Nacional de Colombia, Department of Electrical and Electronics Engineering, Bogotá, Colombia (Tel: (+571) 3165000 ext 11115; e-mail: nabarbosa@unal.edu.co)*
[**] *CNRS, LAAS, University of Toulouse,7 Avenue du Colonel Roche, F-31031 Toulouse, France (e-mail: {nabarbos,louise}@laas.fr*
[***] *Universidad Nacional de Colombia, Department of Mechanical and Mechatronics Engineering, Bogotá, Colombia (Tel: (+571) 3165000 ext 14103; e-mail: vhgrisalesp@unal.edu.co)*

Abstract
Pattern recognition methods, specially classification, has been growing in popularity because its ability to adapt in a changing environment. This paper proposes a classification of such techniques found on a literature review of dynamic classification techniques collected among a variety of fields, including fault detection, identification of moving objects, bank customer segmentation, among others. Based on how the dynamic behavior is incorporated in the classification process (data, classifier structure or both), three main categories are detected: Methods for classifying static objects using dynamic classifiers, methods classifying dynamic objects with static classifiers and finally methods classifying dynamic objects with dynamic classifiers.

Keywords: Pattern recognition, Data processing, Classification, Dynamic systems, Trajectories

1. INTRODUCTION

This report shows the initial solution proposed for the online dynamic clustering problem of time series. Our approach will solve the clustering problem in three stages, data preprocessing, online clustering and offline clustering. Each one of this periods is explain in more detail below.

The data preparation stage extracts information from data streams on episodes, that are a representation defined as a set of two elements: a trend context, given in this case by polynomial fitting coefficients and a time interval. Such episodes describe the stream behaviour. This kind of representation is useful in cases where not only the analysis of the instantaneous information but also the historic behaviour and evolution are needed.

The online-offline clustering stages are inspired in those shown for the first time in Zhang et al. [1997] and popularized by Ester et al. [1996], Aggarwal et al. [2003], Cao et al. [2006], Kranen et al. [2011] among others. The online stage creates micro clusters (μC) that are summarized representations of a data set made by using some statistical and temporal information. This group of data are close enough to make it possible to consider that they belong to the same final cluster. In our work we consider three different types of μC depending on μC density. This dense μC structure has proved outlier detection capabilities in evolving environments Cao et al. [2006]. The offline stage analyses the distribution of μCs and creates the final clusters by a density based approach, that is, dense micro cluster that are close enough(connected) are said to belong to the same cluster. Moreover, a cluster will be defined, similar to Chen and Tu [2007], as the group of connected μCs where every inside μC is dense and every outside cluster is either dense or semi dense. The notion of dense cluster will be explain in the next section.

Different approaches to data stream clustering have been proposed since the 90's. Among these, two main approaches are found, distance based methods and density based methods. On the side of density based methods the CluStream algorithm Aggarwal et al. [2003] is one of the best known. This algorithm is separated in two components: online and offline clustering. In the first component, data are collected, pre-processed and compressed in micro-clusters. In the second component, the micro-clusters are grouped into macro-clusters for different, specific time horizons, using stored pictures of the μCs (called "snapshots") and a modification of the k-means algorithm. These snapshots allow to make classifications in different time windows and also to analyse cluster time evolution. This pyramidal time framework as called for the technique authors, is the main advantage of this algorithm and will be used in this proposal. One disadvantage of the technique are the predefined constant number of micro clusters, that could lead the algorithm to create several clusters for the outliers, which may leave less space for true clusters. Another disadvantage, derived from the selection of k-means as clustering algorithm, is the inability to find non convex clusters.

Based on micro clusters as CluStream, ClusTree Kranen et al. [2011] proposes a hierarchical tree shaped index structure which provides efficient cluster location for object insertion at different levels of granularity achieving

μC-data assignation even at fast rates. Since the ideal is to locate as many data as possible in the leaf nodes of the tree, the algorithm proposes an "hitch-hiker" approach to help the data which were left in high-level nodes to go down to the leaf level nodes. If a point y comes before the insertion process of point x is finished, the algorithm interrupts the insertion process of x and stores it temporarily in a buffer from which it would be taken along in the future as an "hitch-hiker" when a new data arrives looking to be located in a μC on the same path. ClusTree improves the limitation of CluStream about the fix number of μCs, but establishes a maximum number of μCs and when the algorithm reaches this value, has the same limitations as CluStream. Furthermore, since each cluster can absorb only a limited amount of data, similar points can be located in different micro-clusters and possibly even in different final clusters. The proposal of a hierarchical ID structure saves a lot of time reducing the amount of calculation necessary for μC location, nevertheless the deep-first descent approach taken by the authors can lead to non optimal classification, since data could be located within a micro cluster that might not be the closest to it.

Between the algorithms that use density based final clustering DenStream Cao et al. [2006] makes a good proposal that handles outliers and cluster evolution with the use of outlier (low density) and potential (intermediate density) μCs. It saves time by searching with priority core (high density) or potential μCs to locate arriving data. The offline stage of this algorithm uses DBSCAN to cluster the μCs. The problem with this algorithm is the high number of user defined parameters, that makes it unsuitable for users with little knowledge of the process. That problem is also seen but to a lesser extent in the DStream Chen and Tu [2007] algorithm where context should be provided in order to create a grid fine enough for mapping. DStream also provides an explicit way to deal with outliers as low density grid areas that are not considered for initial mapping, making the grid location and the posterior cluster formation faster and accurate.

2. ON-LINE ALGORITHM

Starting from the μC concept shown in the previous section and considering the inclusion of episodes this approach will generate μCs on-line. On the basis of the μCs generated, a density based classification will be performed in an offline fashion.

2.1 Stream Pre-processing

Industrial processes are likely to have hundreds to thousands of signals coming from sensors all over the plant. This signals are usually in a normal operating region, characterized by zero value first and second derivatives (in practice due to the presence of noise $\dot{x} < \epsilon_1$ and $\ddot{x} < \epsilon_2$). It is expected that in the proximity of a process situation change (due to faulty behaviour or to change in the operating point), sensors would show departure from their normal value range ($\dot{x} > \epsilon_1$ and\or $\ddot{x} > \epsilon_2$) following a transitional dynamic that can be characterized by means of trends. A trend is a representation of the variation of a process variable within a temporal period. Process situation assessment can be achieved by extracting trends from the measured signals and evaluating its similarity with known behaviour.

The simplest approach to extract a trend is to fit a polynomial to the data series. According to the Weierstrass approximation theorem, a continuous function defined in a closed interval, can be approximated as closely as desired using polynomials of sufficiently high order as established by Dash et al. [2004]. Consequently, there are two ways to approximate a time function using polynomials: increase the polynomial order or reduce the interval size into one where the function could be described by a low order polynomial. Of these, the later is generally preferred, since polynomial-fitting complexity increases with the polynomial order. One example can be seen in Dash et al. [2004] where the authors propose an interval-halving algorithm for trend extraction which automatically identify qualitative trends using polynomial fitting. A graphical representation of the interval halving algorithm is shown in 1. The algorithm tries to fit a polynomial of order $O < n_max$ to a time series window. If the fitting error is bigger than some threshold established before, the window is split at the midpoint generating two intervals and polynomial fitting is performed on the new interval. The split process finish when the fitting error is below the threshold. Then the leftover data is analysed following the same method until all data in the initial window is fitted. An online implementation of the interval halving algorithm has been developed by Maurya et al. [2010]. There the online trend extraction is achieved using a window that moves as more data become available.

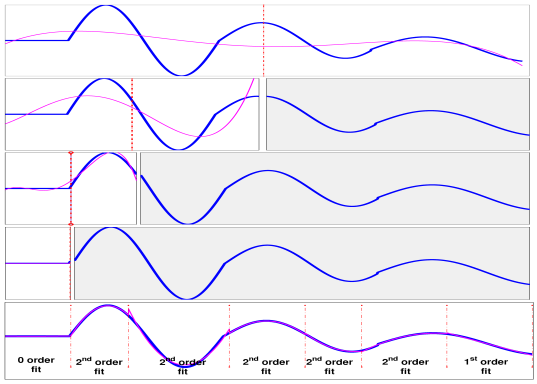

Figure 1. Interval halving polynomial approximation

Based on the same principle of window splitting we develop an algorithm that fits polynomials of order 2 as maximum to the data series. Window is built with the arriving data and as soon as the minimum length window is achieved, data is analysed and the current trend is temporarily characterized. The window will continue to grow until a leap is found in the signal or until the maximum window size is reached. Leaps are found using the difference between the current and the previous measure, Δx. When this difference is bigger than some threshold, the window is split in that point. The threshold is selected to be proportional to the Δx incremental average value as shown in equation (1). The parameter α is selected to reduce the false jump detection rate.

$$\Delta x > \alpha \bar{\Delta x}^* \tag{1}$$

Once a interval is established the fitting process is performed and the coefficients of the best fit (minimum fitting error) are saved if the fitting is acceptable, that is, if the fitting error is insignificant compared to the noise present in the signal (as dictated by F-test). In order to be able to evaluate this, the noise variance has to be determined. As reported by Bakshi [1999] wavelet theory and specifically wavelet multi-scale decomposition can be used to find an estimation of the noise standard deviation in the measured signal. Wavelet multi-scale decomposition is a technique widely used to filter signals with stochastic errors. In that method signals are decomposed on a selected family of basis functions (described by its coefficients), then the coefficients of small value are eliminated and the rectified signal reconstructed from the filtered coefficients. Decomposition coefficients exhibit some characteristics that help in the finding of noise variance. Coefficients corresponding to the true signal are larger in magnitude than those corresponding to noise, nevertheless, there are less in quantity. Given this, as stated by Bakshi, the robust median absolute deviation (MAD) gives a way to estimate the noise variance from wavelet decomposition coefficients. Noise standard deviation is calculated as seen in (2), where d_m denotes the wavelet coefficients at the selected scale.

$$\sigma_m = \frac{1}{0.6745} median\left(|d_m|\right) \tag{2}$$

Once a fit is found polynomial coefficients are append to current signal value forming an episode describes as:

$$e\left(x,t\right) = \left(x, c_2, c_1, c_0, t_0, \Delta t\right) \tag{3}$$

where x correspond to the feature vector current value. c_i with $i \in \{0,1,2\}$ are d-dimensional vectors containing the coefficients describing the polynomial, t_0 is the episode start time and Δt is the episode duration. From now on the polynomial coefficients will be considered as process features and treated as such. If the fitting process fails to find a polynomial approximation in which the fitting error is insignificant compared to the noise and the interval has been resized to its minimum size, the approximation with the minimum fitting error is used.

2.2 On-line Clustering

Considering a d-dimensional episode $e\left(x,t\right)$, an μC will be the representation of a group of episodes **close in all dimensions** and whose information is summarized in a characteristic feature vector (CF). This CF have the form $CF_k = (n_k, LS_k, Var_k, \Delta_k, QM_k, t_{lk}, t_{sk}, D_k, Class_k)$, where $n_k \in \Re$ is the number of elements in the cluster k, $LS_k \in \Re^d$ is the vector containing the linear sum of elements for each quantitative descriptor, $Var_k \in \Re$ is the variance representing the spread of the group of points assigned to cluster k, Δ_k is the episode duration, QM_k is the vector containing the qualitative descriptors, $t_{lk} \in \Re^1$ is the time when the last element was assigned to that μC, $t_{sk} \in \Re^1$ is the time when the μC was created, D_k is the cluster density and $Class_k$ is the cluster label if known. In order to maintain an up-to-date structure, micro clusters will be weighted with an exponential decay function dependant of current time t_i and last assignation time t_{lk}. This function $\beta^{-\lambda(t_i - t_{lk})}$ emulates ageing process over a damped window. If β is chosen as 2^{ψ} then the half life of data in the window will be $\frac{1}{\psi\lambda}$, $\lambda > 0$. When a new episode (e_i, Δ_i) is assigned to a cluster, features are actualized as follows:

$$n_k^{(t)} = n^{(t-1)}\beta^{-\lambda(t-t_{lk})} + 1 \tag{4}$$

$$LS_k^{(t)} = LS^{(t-1)}\beta^{-\lambda(t-t_{lk})} + e_i \tag{5}$$

$$Var_k^{(t)} = \frac{1}{n_k^{(t)}} SS_k^{(t)} - \left(\frac{LS_k^{(t)}}{n_k^{(t)}}\right)^2 \quad \text{with } SS_k^{(t)} = SS_k^{(t-1)}\beta^{-\lambda(t-t_{lk})} e_i^2 \tag{6}$$

$$\Delta_k^{(t)} = \Delta_k^{(t-1)}\beta^{-\lambda(t-t_{lk})} + \Delta_i \tag{7}$$

$$t_{lk} = t_i \tag{8}$$

$$Class_k = Class_i \tag{9}$$

In general, if no object is added to a cluster's CF during the time interval $(t, t+\Delta t)$, its CF in $t+\Delta t$ can be calculated from $CF^{(t)}$ using the decay function for the weighted parameters as follows:

$$CF^{(t+\Delta t)} = \beta^{(-\lambda\Delta t)} CF^{(t)} \tag{10}$$

To find clusters with arbitrary shape, density based clustering will be executed over the μCs. Three different types of μCs will be handled, dense μC ($D\mu C$), semi-dense μC ($S\mu C$) and low density or outlier μC ($O\mu C$). The difference between each type of cluster will be established based on a threshold. $S\mu C$ could be a product of an increment in the number of outliers or part of a cluster creation, so have to be updated more frequently than other clusters. To speed up the analysis we will use three lists. The first one maps the active micro clusters $D\mu C$ or $S\mu C$. The second one will contain the current $O\mu C$s. The third list maps those μCs that were once $D\mu C$ but, because of a lack of new elements, have lost density, and became non active μCs.

Regarding qualitative descriptors, each μC will accept elements that contain only the same set of descriptors, in other words, inside the qualitative vector QM_k only one modality of each descriptor can be found and all elements in that μC must present this set of modalities.

The size of the boxes are set as partially fix and is established according to the data context (see equation (11)). The algorithm can work over a known context, given by the user, or can built the context on the fly. The context gives quantitative descriptors minimum and maximum and qualitative descriptor modalities.

$$S_k^d = \frac{max - d - min_d}{\varphi} \quad \forall \text{ quantitative } d \tag{11}$$

Cluster density is calculated using the current number of points n_k and the current hypervolume of the bounding box as shown in (12).

$$D_k = \frac{n_k}{V} \tag{12}$$

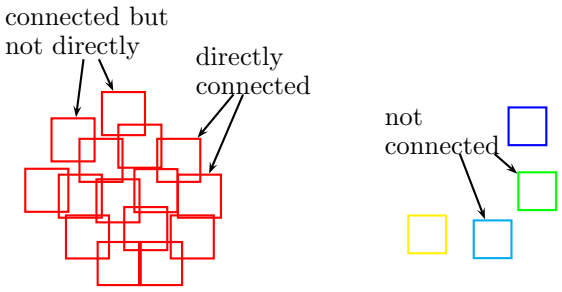

Figure 2. Representation of clustering based on μC

A μC_z is said to be dense at a time t if it satisfies the inequality (13), with V_k the volume of the cluster box k and α some proportional variable giving the threshold. For the case where all μC has the same size the condition can be simplified as seem in (14).

$$D_z \geq \alpha \frac{\sum_{k=1}^{K} n_k}{\sum_{k=1}^{K} V_k} \tag{13}$$

$$D_z \geq \alpha \frac{\sum_{k=1}^{K} n_k}{KV} \tag{14}$$

At time t, an μC_z is said to be semi-dense if its density fulfils the following inequality:

$$\alpha \frac{\sum_{k=1}^{K} n_k}{KV} \geq \frac{d_z}{\sum_{k=1}^{K} d_k} \geq \frac{\alpha}{2} \frac{\sum_{k=1}^{K} n_k}{KV} \tag{15}$$

An μC_z is considered $O\mu C$ if its density is lower than the lower limit in equation (15).

3. OFF-LINE DENSITY CLUSTERING

To find the final clusters or macro cluster (MC) the dense character of an μC and its neighbours is take into account. In this work the following definition are adapted from those given by Cao et al. [2006].

Definition 1. Let μC_x, μC_y be μCs, then μC_x and μC_y are said to be **directly connected** if their hyperboxes, overlap in all dimensions.

Definition 2. An μC μC_1 is said to be connected to μC_n if there is a chain of μCs $\{\mu C_1, \mu C_2, \cdots, \mu C_n\}$ such that μC_i is directly connected to μC_{i+1} for $i = 1, 2, \cdots, n-1$.

A set of connected μCs is said to be a group.

Finally, a μCs group id said to be a MC if every inside μC of the group is a $D\mu C$ and every outside μC is either a $D\mu C$ or a $S\mu C$, for illustration over this concept see figure 2.

4. ALGORITHM OPERATION

The algorithm can start working directly from data or taking into account clusters learn in an offline learning stage. If there is no previous knowledge, the algorithm receives the data coming from sensor measures and accumulated until the minimum window size of dyadic length is achieved. At that moment the denoising and fitting processes era applied over the window, and the first episode is extracted. The episode characterization process continues to detect episodes as more data become available. The already represented episodes pass to the clustering phase of the algorithm that will be carried on in parallel. When the first identified episode $e\,(x,t)$ arrives the algorithm verifies the existence of μCs (if an offline learning stage is used). If no μC exist, it creates a μC using the data of the current point. If there are already one or more μCs created, the algorithm finds out which $D\mu C$ or $S\mu C$ is the closest. The distance between a point x and a cluster c_k is calculated as the sum of the distances between cluster representation and point value for each descriptor. For the d_1 quantitative descriptors the distance is calculated using the absolute value (16) and for the d_2 qualitative descriptors, distance is defined as shown in (16), the total point-cluster distance is shown in equation (17).

$$dis^i = \begin{cases} abs\left(x^i - c_k^i\right) & \text{if } d_i \text{ is quantitative} \\ \begin{cases} 0 & \text{if } x^i = QM_k^i \\ 1 & otherwise \end{cases} & \text{if } d_i \text{ is qualitative} \end{cases} \tag{16}$$

$$dis_{(x,c_k)} = \sum_{i=1}^{d} dis^i \tag{17}$$

Once the closest cluster is found, the maximum distance condition is verified, that is, the point must fit inside the maximum possible bounding box MAX_{BB} centred in the μC. If there is a fit, the qualitative descriptors are checked and only if they are exactly the same, the point is absorbed by the cluster. Otherwise the algorithm tries to merge the point with the closest $O\mu C$. If there the distance condition and the qualitative descriptors condition stand, the point is merged and after that, the density of the $O\mu C$ is verified to decide if the cluster has grown into a $S\mu C$. When an $O\mu C$ becomes denser it is removed from the outlier list and with its statistics a new $S\mu C$ is created and inserted into the active μC list. In the case when no cluster is a radius match to the point a new $O\mu C$ is created with the point and the current time is used as $O\mu C$'s t_s. Each window period $S\mu C$ has to be updated and its density evaluated in order to establish if it has fall below the semi-dense threshold and then the $S\mu C$ is eliminated and with its statistics an $O\mu C$ is created. $O\mu C$s are also evaluated each window period to find out if its density is below a low-density threshold. If that is the case the $O\mu C$ is eliminated and no longer considered.

REFERENCES

Charu C Aggarwal, Jiawei Han, Jianyong Wang, and Philip S Yu. A framework for clustering evolving data streams. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 81–92. VLDB Endowment, 2003.

Bhavik R. Bakshi. Multiscale analysis and modeling using wavelets. *Journal of Chemometrics*, 13(3-4):415–434, 1999. ISSN 1099-128X. doi: 10.1002/(SICI)1099-128X(199905/08)13:3/4⟨415::AID-CEM544⟩3.0.CO;2-8. URL http://dx.doi.org/10.1002/(SICI)1099-128X(199905/08)13

Feng Cao, Martin Ester, Weining Qian, and Aoying Zhou. Density-based clustering over an evolving data stream with noise. In *SDM*, 2006.

Yixin Chen and Li Tu. Density-based clustering for real-time stream data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and data mining*, pages 133–142, 2007.

Sourabh Dash, Mano Ram Maurya, Venkat Venkatasubramanian, and Raghunathan Rengaswamy. A novel interval-halving framework for automated identification of process trends. *AIChE journal*, 50(1):149–162, 2004.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databdata with noise. In *KDD*, 1996.

Philipp Kranen, Ira Assent, Corinna Baldauf, and Thomas Seidl. The ClusTree: indexing micro-clusters for any stream mining. *Knoledge and information systems*, 29 (2):249–272, 2011.

Mano Ram Maurya, Praveen K Paritosh, Raghunathan Rengaswamy, and Venkat Venkatasubramanian. A framework for on-line trend extraction and fault diagnosis. *Engineering Applications of Artificial Intelligence*, 23(6):950–960, 2010.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997. ISSN 1384-5810. doi: 10.1023/A:1009783824328. URL http://dx.doi.org/10.1023/A%3A1009783824328.