Graded Possibilistic Clustering of Non-Stationary Data Streams

A. Abdullatif[1,2], F. Masulli[1,3], S. Rovetta[1], and A. Cabri[1]

[1] DIBRIS - Dept of Informatics, Bioingengering, Robotics and Systems Engineering
University of Genoa, Via Dodecaneso 35, 16146 Genoa, Italy
[2] VEDECOM Institute, Versailles, France
[3] Sbarro Institute for Cancer Research and Molecular Medicine
College of Science and Technology, Temple University, Philadelphia, PA, USA
{amr.abdullatif,francesco.masulli,stefano.rovetta}@unige.it

Abstract. Multidimensional data streams are a major paradigm in data science. This work focuses on possibilistic clustering algorithms as means to perform clustering of multidimensional streaming data. The proposed approach exploits fuzzy outlier analysis to provide good learning and tracking abilities in both concept shift and concept drift.

1 Introduction

Multidimensional data streams have arisen as a relevant topic in data science during the past decade [1]. They arise in an ever increasing range of fields, from the web, to wearable sensors, to intelligent transportation systems, to smart homes and cities.

Data streams may represent actual time series, or quasi-stationary phenomena that feature longer-term variability, e.g., changes in statistical distribution or a cyclical behavior. In these non-stationary conditions, any model is expected to be appropriate only in a neighborhood of the point in time where it has been learned. Its validity may decrease smoothly with time (*concept drift*), or there may be sudden changes, for instance when switching from one operating condition to a new one (*concept shift*).

This work focuses on possibilistic clustering [3] as a means to perform clustering of multidimensional streaming data. We specifically exploit the ability, provided by the Graded Possibilistic c-Means [2], to learn clustering models iteratively using both batch (sliding-window) and online (by-pattern) strategies that track and adapt to concept drift and shift in a natural way.

2 Clustering Non-Stationary Data Streams

Clustering streams requires tackling related, but different, problems: Handling unbounded, possibly large data; detecting model changes (drift and shift); adapting to model changes. These are often treated independently in the literature.

Model shift, either for rejection or tracking, has been extensively studied, also under the name of change detection or change point identification. The key to detect model differences is measuring the fit of observations. An observation that does not fit a given model is either called novelty [6] or outlier [7] depending on the focus of the inquiry. Individual outliers can be detected or rejected by evaluating their estimated probability or membership and comparing it to a threshold, using distance-based or density-based criteria [8], or simply not taken into account by using robust methods [9]. Alternatively it is possible to gather a new sample and apply sequential statistical tests for comparing empirical distributions [10]. Regarding fuzzy methods, techniques have been proposed mainly for robustness [11],although some approaches directly include measures of *outlierness* [13] that can indicate the inadequacy of a given current model.

Model drift is a much more difficult issue, usually simply tackled by continuous learning. This brings us to the third problem, learning strategies for dynamically changing models. Continuous learning can be done either by recomputing, if affordable, or by incremental updates. Most fuzzy clustering models are fit with a batch training procedure, owing to their derivation from (crisp) k Means. This batch optimization, iterative and prone to local minima, makes them unsuitable for data sets of very large/virtually unbounded cardinality. However, the same prototype-based representation lends itself very well to alternative training methods, e.g., online or "by pattern" [14],which are naturally suitable for incremental updating. Due to the non-stationarity, it is not possible to resort to the extensive literature about convergence of stochastic approximation or related methods [15]. However, a measure of model inadequacy can be used to modulate the required amount of update, avoiding unnecessary waste of computational resources especially when the data are large in size. In addition, it might be appropriate to dynamically change the number of centroids. In [16] this was addressed with a cross-validation approach. However we didn't take this problem into consideration.

We consider the case where the data are represented by numerical feature vectors. In general they will be multi-dimensional, even in the case of scalar-valued time series where time-lag encoding is used. We assume that the data are generated independently by some underlying probability distribution, but on the long run the distribution may change. As long as the underlying distribution doesn't "change much", it can be considered stationary. However, for sufficiently long observation periods this approximation is no longer valid. The extent of validity of a stationary approximation depends on the rate and intensity of change of the source distribution, which is clearly difficult to estimate. In this work we don't consider this problem, but provide methods for two different scenarios.

The goal of the analysis is to summarize these data by (fuzzy) clustering, with a process that should learn continuously from the input patterns as they arrive. The data are stored in a sliding window W of constant size $w \geq 1$ which is updated every s observations by deleting the oldest s patterns and adding the new ones, so that, at each time, the current W has an overlap of $w - s$ patterns with the previous one. We collect the s incoming observations in a probe set S

before incorporating them in the window; the probe set is used to adaptively tune the learning parameters.

We assume that, within W, the data can be considered independent and identically distributed i.i.d. As anticipated in the introduction, we will focus on the following two scenarios:

- The source change rate so slow that W is sufficient to infer the clustering structure;
- The rate is so high that W is not sufficient to perform a complete clustering, and an incremental procedure is required.

Accordingly, in this work we consider the cases $w > 1$, $s > 1$ (batch learning, batch density estimate) and $w = s = 1$ (online learning, online density estimate).

3 The Graded Possibilistic $c-$Means Model

In central clustering data objects are points or vectors in data space, and c clusters are represented by means of their "central" points or centroids $\mathbf{y}_j$. The Graded Possibilistic model is a *soft* central clustering method, implying that cluster membership can be partial. This is usually represented by means of cluster indicators (or membership functions) which are real-valued rather than integer.

In many cases methods are derived as the iterative optimization of a constrained objective function [4], usually the mean squared distortion:

$$D = \frac{1}{n} \sum_{l=1}^{n} \sum_{j=1}^{c} u_{lj} ||\mathrm{x}_l - \mathbf{y}_j||^2 \tag{1}$$

Centroids are obtained by imposing $\nabla D = 0$:

$$\mathbf{y}_j = \frac{\sum_{l=1}^{n} u_{lj} x_l}{\sum_{l=1}^{n} u_{lj}} \; . \tag{2}$$

Usually constraints are placed on the sum $\zeta_l = \sum_{j=1}^{c} u_{lj}$ of all memberships for any given observation x_l. The value ζ_l can be interpreted as the total membership mass of observation x_l. We now survey from this perspective two related soft clustering methods.

The Maximum Entropy (ME) approach [5] imposes $\zeta_l = 1$, so we are in the "probabilistic" case, where memberships are formally equivalent to probabilities.

The objective J_{ME} includes the distortion, plus an entropic penalty with weight β and the normality constraint $\sum_{j=1}^{c} u_{lj} = 1 \; \forall l$. The first-order necessary minimum condition $\nabla J_{\mathrm{ME}} = 0$ then yields

$$u_{lj} = \frac{e^{-||\mathrm{x}_l - \mathbf{y}_j||^2/\beta}}{\zeta_l} \; . \tag{3}$$

On the other end of the spectrum, the Possibilistic c-Means in its second formulation (PCM-II) [3] does not impose any constraint on ζ_l, so memberships are not formally equivalent to probabilities; they represent degrees of typicality.

The objective $J_{\mathrm{PCM-II}}$ includes the distortion plus a penalty term to discourage (but not exclude) extreme solutions. This term contains an individual parameter β_j for each cluster, and $\nabla J_{\mathrm{PCM-II}} = 0$ yields

$$u_{lj} = e^{-||\mathbf{x}_l - \mathbf{y}_j||^2/\beta_j} \ . \tag{4}$$

Both eq. (4) and eq. (3) can be generalized to a unique, common formulation as was done in [2] as follows:

$$u_{lj} = \frac{v_{lj}}{Z_l}, \tag{5}$$

where the *free membership* $v_{lj} = e^{-||\mathbf{x}_l - \mathbf{y}_j||^2/\beta_j}$ is normalized with some term Z_l, a function of $\mathbf{v}_l = [v_{l1}, v_{l2}, \dots, v_{cl}]$ but not necessarily equal to $\zeta_l = \sum_{j=1}^{c} v_{lj}$ $= |\mathbf{v}_l|_1$. This allows us to add a continuum of other, intermediate cases to the two extreme models just described, respectively characterized by $Z_l = \zeta_j$ (probabilistic) and $Z_l = 1$ (possibilistic). Here we use the following formulation:

$$Z_l = \zeta_l^{\alpha} = \left(\sum_{j=1}^{c} v_{lj} \right)^{\alpha} , \quad \alpha \in [0,1] \subset \mathbb{R} \tag{6}$$

The parameter α controls the "possibility level", from a totally probabilistic ($\alpha = 1$) to a totally possibilistic ($\alpha = 0$) model, with all intermediate cases for $0 < \alpha < 1$.

4 Outlierness measurement through graded possibilistic memberships

The nature of membership functions suggests a characterization of outliers similar to Davé's Noise Clustering model that was used in the context of robust clustering [12]. Given a trained clustering model, i.e., a set of centroids and a set of cluster widths β_j, we exploit the properties of the possibilistic memberships to evaluate the degree of *outlierness*. We define outlierness as the membership of an observation to the concept of "being an outlier" with respect to a given clustering model. Differently from other approaches based on analyzing pattern-centroid distances [17], the graded possibilistic model used in this work provides a direct measure of outlierness. We propose to measure the total mass of membership to clusters ζ_l, which, by definition of the graded possibilistic model, does not necessarily equal 1, and measure whether and how much it is less than 1. Quantitatively, we define an index Ω as follows:

$$\Omega(\mathbf{x}_l) = \max\{1 - \zeta_l, 0\} \ . \tag{7}$$

Outlierness can be modulated by an appropriate choice of α. Low values correspond to sharper outlier rejection, while higher values imply wider cluster regions and therefore lower rejection. For $\alpha = 1$ the model becomes probabilistic and loses any ability to identify or reject outliers.

We observe that $\zeta_l = \sum_j u_{lj} \in (0, c)$. However:

- values $\zeta_l > 1$ are typical of regions well covered by centroids;
- but $\zeta_l \gg 1$ is very unlikely for good clustering solutions without many overlapping clusters;
- finally, $\zeta_l \ll 1$ characterizes regions not covered by centroids, and any observation occurring there is an outlier.

The index Ω is defined as the complement to one of ζ_l, with negative values clipped out as not interesting.

The outlierness index is a pointwise measure, but it can be integrated to measure the frequency of outliers. For crisp decision-making, a point could be labeled as outlier when Ω exceeds some threshold. It is therefore easy to count the proportion (frequency) of outliers over a given set of probe points S.

However, we take advantage of the fact that Ω expresses a fuzzy concept, and rather than simply counting the frequency we can measure an *outlier density* $\rho \in [0, 1)$ defined in one of the following ways.

$$\rho_{\mathrm{M}} = \frac{1}{|S|} \sum_{l \in S} \Omega(\mathbf{x}_l) \tag{8}$$

The density ρ_{M} accounts for both frequency and intensity, or degree of anomaly, of outliers. A high number of borderline outliers is equivalent to a lower number of stronger outliers, provided their mean value is the same. To give more emphasis to the case where some observation have a higher outlierness, an alternative definition can be used:

$$\rho_{\mathrm{RMS}} = \frac{1}{|S|} \sqrt{\sum_{l \in S} \left(\Omega(\mathbf{x}_l)\right)^2} \tag{9}$$

The definition $\rho = \rho_{\mathrm{RMS}}$ will be adopted in the experiments.

5 Learning regimes

We distinguish between three different situations, corresponding to three possible *learning regimes.*

1. Concept drift. The source is stationary or changing smoothly and slowly ($\Omega = 0$, density is low). Action to be taken: The model should be incrementally updated to track possible variations. We can call this the *baseline learning regime.*
2. Outliers. One or few isolated observations are clearly not explained by the model, which means that they have outlierness ($\Omega > 0$, density is low). Action to be taken: Incremental learning should be paused to avoid skewing clusters with atypical observations (*no-learning regime*).
3. Concept shift. Several observations have outlierness ($\Omega > 0$, density is high). Action to be taken: The old clustering model should be replaced by a new one. This is the *re-learning regime.*

The learning depends on a parameter θ that balances between stability ($\theta \approx 0$, model stays the same) and plasticity ($\theta \approx 1$, model changes completely), so it is possible to modulate this parameter as a function of outlier density, so that the three learning regimes (baseline, no learning and re-learning) can be implemented.

Since learning regimes are yet another fuzzy concept, rather than splitting them into clear-cut regions, in our experiments we employed a smooth function that is controlled by parameters: The user should interactively select their values to obtain the desired profile. This procedure is similar to defining the membership function for a linguistic variable. The function we used is the following:

$$\theta = 1 + \theta_0 \exp\left(-\frac{\rho}{\tau_1}\right) - \exp\left(-\left(\frac{\rho}{\tau_2}\right)^{\gamma}\right) \qquad (10)$$

- θ_0 is the baseline value of θ, used when new data are well explained by the current model (baseline learning regime).
- τ_1 is a scale constant, analogue to a time constant in linear dynamical system eigenfunctions, that determines the range of values for which the baseline learning regime should hold.
- τ_2 is a scale constant that determines the range of values for which the re-learning regime should hold.
- γ is an exponential gain that controls how quick the relearning regime should go to saturation, i.e., to $\theta \approx 1$.

In the transition between baseline learning and re-learning, this function has a valley that brings the value of θ close to zero, implementing the no-learning regime.

6 Learning possibilistic stream clustering

In the algorithms proposed in this section, the degree of possibility α is assumed to be fixed. This quantity incorporates the a-priori knowledge about the amount of outlier sensitivity desired by the user.

In batch learning, at each time t we train the clustering model on a training set (window) W_t of size w and evaluate ρ on the next $s < w$ observations (set S_t). Then we compute W_{t+1} for the next step by removing the s oldest observations and adding S_t, so that the training set size remains constant. Finally, the amount of learning required is estimated by computing $\theta(\rho)$ according to (10).

From an optimization or learning perspective, we are estimating the true (expected) objective function on the basis of a set W of w observations, that we use to compute a sample average. The batch process is initialized by taking a first sample W_0 and performing a complete deterministic annealing optimization on it with an annealing schedule $B = \{\beta_1, \ldots, \beta_b\}$. In subsequent optimizations, the annealing schedule is shortened proportionally to the computed value of $\theta(\rho) \in [0, 1]$: When $\theta = 1$ the complete B is used ($|B| = b$ optimization steps); when $\theta = 0$ no training is performed (0 steps); when $0 \le \theta \le 1$ a corresponding fraction B_θ of the schedule B is used, starting from step number $\lceil b \cdot \theta \rceil$ up to β_b (that is, $\lfloor b \cdot (1 - \theta) \rfloor$ steps in total). The updating rule for a generic centroid (non-stationary data streams) is the following:

$$\mathbf{y}_j(t+1) = \mathbf{y}_j(t) + \eta \frac{\sum_{l=1}^{w} u_{lj} \left(\mathbf{x}_l - \mathbf{y}_j(t)\right)}{\sum_{l=1}^{w} u_{lj}} \tag{11}$$

$$= (1-\eta)\mathbf{y}_j(t) + \eta \frac{\sum_{l=1}^{w} u_{lj} \mathbf{x}_l}{\sum_{l=1}^{w} u_{lj}} \tag{12}$$

For *stationary data streams* the distribution of any sample $W(t)$ is constant w.r.t. t, and therefore its weighted mean $\frac{\sum_{l=1}^{w} u_{lj}\mathbf{x}_l}{\sum_{l=1}^{w} u_{lj}}$ is also constant. In this case

$$\mathbf{y}_j(t \to \infty) \to \frac{\sum_{l=1}^{w} u_{lj} \mathbf{x}_l}{\sum_{l=1}^{w} u_{lj}} \,. \tag{13}$$

With fixed η, Eq. (11) computes an exponentially discounted moving average.

In summary, for the batch case we use θ to modulate the number of annealing steps and, consequently, the scale parameter β. The optimization is longer, and starts with a higher coverage (higher β), in the re-learning regime; it is shorter and more localized in the baseline learning regime; it does not occur at all (zero steps) in the no-learning regime. Figure 1 outlines the batch learning algorithm.

```
init: Select $\alpha$, $B = \{\beta_1, \dots, \beta_b\}$, $\theta = \theta(\rho)$
      Read first $w$ observations into $W_0$
      Learn clustering model from $W_0$ using annealing schedule $B$

loop: (for each time $t : 1, \dots, \infty$)
      Read next $s$ observations into $S_t$
      Use (7) to compute $\Omega$ for all observations in $S_t$
      Use (9) to compute $\rho$
      Use (10) to compute $\theta$
      Discard $s$ oldest observations from $W_{t-1}$
      Update $W_t \leftarrow [\ W_{t-1} \quad S\ ]$
      Learn clustering model from $W_t$ using annealing schedule $B_\theta$
```

Fig. 1. Batch possibilistic stream clustering

Now for real time learning we provide an online learning method. This case can be modeled as a limit case of the batch method. At each time t we train the clustering model on a training set W_t of size 1, i.e., one observation, and evaluate ρ on the next $s = 1$ observation forming the "set" S_t. Then we compute W_{t+1} for the next step by replacing the single observation with that in S_t. The amount of learning required is estimated by computing $\theta(\rho)$ according to (10). In this case the updates are incremental and therefore for ρ as well we propose an incremental computation according to the following discounted average formula:

$$\rho_t = \lambda \varOmega_t + (1-\lambda)\rho_{t-1} \; . \tag{14}$$

In this case as well, the process is initialized by taking a first sample W_0 and performing a complete deterministic annealing optimization on it with an annealing schedule $B = \{\beta_1, \ldots, \beta_b\}$.

However, after the initial phase $w = 1$, and the estimate of the objective function cannot be obtained by approximating an expectation with an average. So we resort to a stochastic approximation procedure [18]. This results in the following iterative update equations:

$$\mathbf{y}_j(t+1) = \mathbf{y}_j(t) + \eta_t u_{lj}(\mathbf{x}_l - \mathbf{y}_j) \tag{15}$$

for each centroid, $j = 1, \ldots, c$, with learning step size η_t The update equation for the memberships is still given by Eqs. (5) and (6).

Differently from the batch case, after the initialization step a deterministic annealing schedule is not needed; rather, we have a stochastic annealing step size η_t. There are well-known conditions on η_t for convergence in the stationary case [18]:

$$\sum_{t=1}^{\infty} \eta_t = \infty \qquad \text{and} \qquad \sum_{t=1}^{\infty} \eta_t^2 < \infty \tag{16}$$

However, these conditions obviously do not hold in the nonstationary case, and this topic has not been thoroughly studied in the literature because the conclusions depend on the specific problem setting.

The strategy adopted in this work is to have the step size η be directly proportional to $\rho = \varOmega$, i.e., $\eta = \eta_0 \cdot \rho$ for a user-selected constant η_0. An averaging effect is obtained through the stochastic iterative updates. In this way, after initialization, the intensity of updates depends on the degree of outlierness of the current observation.

To avoid premature convergence, the possibility degree α is also made dependent on ρ, so as to increase centroid coverage when outliers are detected. The formula used is:

$$\alpha = \alpha_{\min} + \rho(1 - \alpha_{\min}) \tag{17}$$

Figure 2 sketches the online learning algorithm.

7 Experimental results

Synthetic data sets containing concept drift (we select the Gaussian and electricity data sets) were generated using the Matlab program ConceptDriftData.m[4] [19]. We also Integrated our model in a traffic flow management system [20]. The proposed work was used as a generative model to asses and improve the accuracy of a short term traffic flow forecasting model.

[4] Available under GPL at https://github.com/gditzler/ConceptDriftData.

```
init: Select $\alpha$, $B = \{\beta_1, \ldots, \beta_b\}$, $\theta = \theta(\rho)$, $\eta_0$
      Read first $w$ observations into $W_0$
      Learn clustering model from $W_0$ using annealing schedule $B$

loop: (for each time $t : 1, \ldots, \infty$)
      Read next observation
      Use (7) to compute $\Omega$ for current observation
      Use (10) to compute $\theta$ with $\rho = \Omega$
      Learn clustering model from current observation using Eq. (15) with learning step $\eta_t = eta_0 \cdot \theta$
```

Fig. 2. Online possibilistic stream clustering

The Gaussian data set already include concept drift, so outliers and concept shift were added by removing a number of observations in two parts of the data sequence. Discontinuities in the sequence are therefore introduced at 50% and 75% of the streams. In addition, the final 25% was shifted by adding an offset to all the data. Results are here shown for the Gaussians dataset. This includes four two-dimensional, evolving Gaussian with equal and known centers and spreads. After introducing discontinuities and shift, the data were remapped into $[0, 1] \times [0, 1]$. This procedure ensures that a ground truth is available at all times.

The data set contains 2500 observations. The parameters used for the experiments are listed in Table 1.

Table 1. Parameters used in the experiments

Parameter	**Symbol**	**Batch**	**Online**	**Note**
Training window size	w	200	—	
Probe window size	s	30	—	
Possibility degree	α	0.7	0.7	(1)
Num. annealing steps	b	20	—	
β schedule		linear	linear	(2)
Starting value for annealing	β_1	0.05	0.05	(2)
Ending value for annealing	β_b	0.002	0.002	(2)
Density estimation function	$\rho(\Omega)$	ρ_{RMS}	Ω	
Coefficient for discounted avg.	λ	—	0.01	
Parameters for computing θ	θ_0	0.3	0.3	
	τ_1	0.01	0.01	
	τ_2	0.5	0.5	
	γ	2	2	

(1) For online: minimum value, maximum is 1.
(2) For online: only in the batch initialization phase.

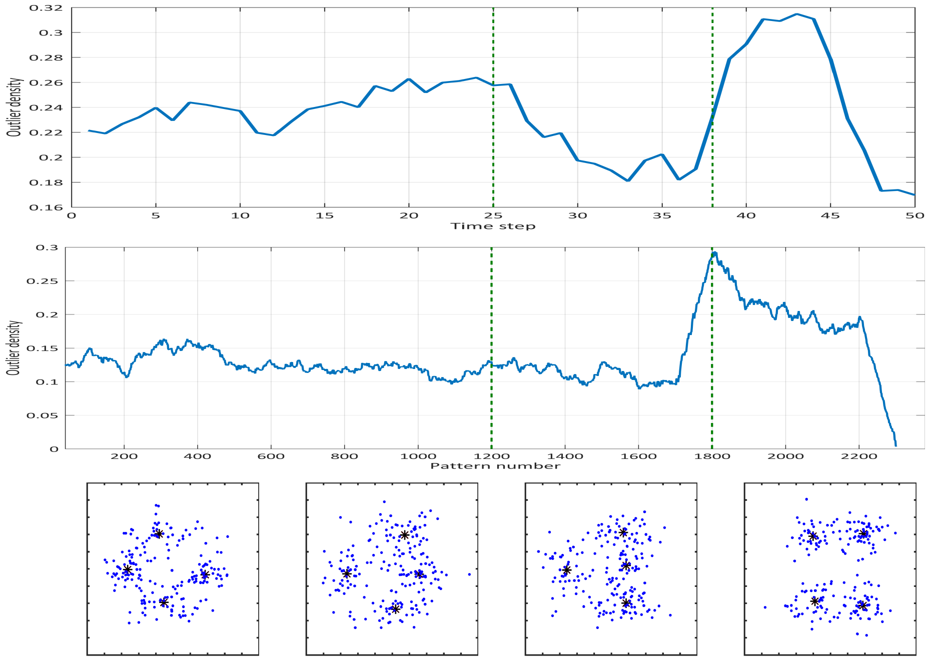

Fig. 3. Outlier density ρ with concept drift and shift. The true model is continuously evolving. The vertical lines mark points where the stream has been cut to create a discontinuity (concept shift). Samples of the training are shown below the graph.

Figure 3 shows the graphs of outlier density ρ in the batch (upper plot) and online (middle plot) cases. For reference, plots of the data taken at various stages are displayed in the bottom. The upper (batch) plot shows $\rho = \rho_{\mathrm{RMS}}$ computed on the probe set S at each iteration. The middle (online) plot shows $\rho = \Omega$. During training this is evaluated at each pattern. For clearer display, average over the past few iterations, rather than instantaneous value, is shown in the figure. In both plots, vertical lines indicate discontinuities (concept shift).

The graphs show the effectiveness of the proposed indexes in indicating the conditions occurring in the stream at each point in time. From the bottom plots it is evident that the first discontinuity occurs between similar configurations, so there is no actual concept shift; in fact, the graph in the batch case highlights that after the discontinuity clusters are closer to each other, so the concentration of data is higher, the clustering task is easier, and the outlier density decreases.

The second discontinuity produces instead a relevant change in centroid configuration, and this is evident in both graphs. However the online version is much quicker to respond to the variation, as indicated by the more steeply increasing plot around the second discontinuity.

A comparative study was performed with two other methods. The first, used as a baseline, is a simple non-tracking k-means, trained once and used without

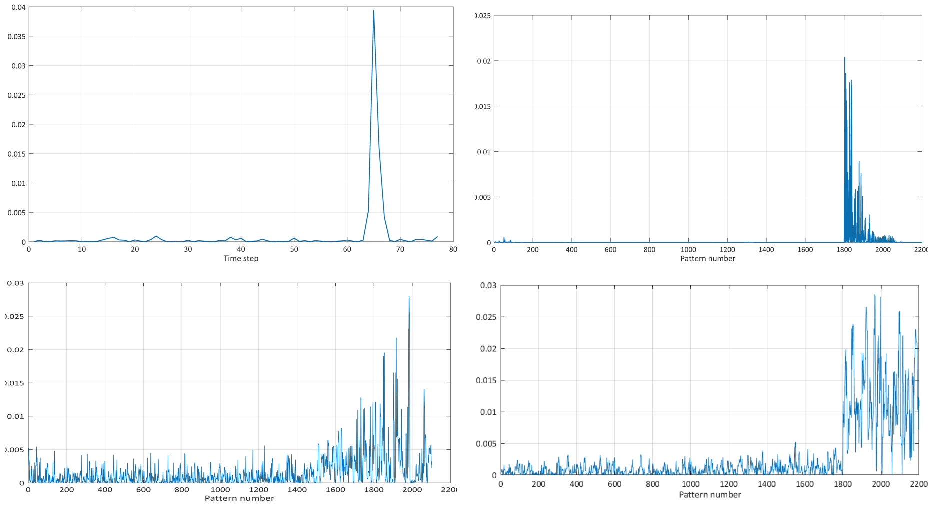

Fig. 4. Tracking error. Absolute difference between distortion w.r.t. true centroids and distortion w.r.t. learned centroids. Top left: batch. Top right: online. Bottom left: TRAC-STREAMS. Bottom right: k means, statically trained.

updates. The second comparison is with a method with similar goals, called TRAC-STREAMS [13], which performs a weighted PCM-like clustering while updating by pattern.

To perform a comparative analysis, for each method we measured tracking performance by computing the distortion (mean squared error) with respect to the learned centroids and to the true centroids, that we have available since the data set is synthetic. Figure 4 shows the tracking performance, obtained by plotting the absolute difference of distortions computed for the "true" and learned models. Apart from the abrupt discontinuity, quickly recovered in both the batch and online cases, the difference is extremely limited, and superior to the other two methods considered.

8 Conclusions

We have presented a method that exploits fuzzy outlier analysis to provide good learning and tracking abilities in both concept shift and concept drift. The method builds upon a possibilistic clustering model that naturally offers a fuzzy outlierness measure.

The proposed method is currently being deployed in several applications, ranging from urban traffic forecasting to ambient assisted living. Several aspects are being investigated. Future work will include improvements in automatic setting of model parameters, as well as in the optimization process.

References

1. C. C. Aggarwal, *Data streams: models and algorithms.* Springer Science & Business Media, 2007, vol. 31.
2. F. Masulli and S. Rovetta, "Soft transition from probabilistic to possibilistic fuzzy clustering," *IEEE Transactions on Fuzzy Systems*, vol. 14, no. 4, pp. 516–527, August 2006.
3. R. Krishnapuram and J. M. Keller, "The possibilistic C-Means algorithm: insights and recommendations," *IEEE Transactions on Fuzzy Systems*, vol. 4, no. 3, pp. 385–393, August 1996.
4. J. C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms.* Norwell, MA, USA: Kluwer Academic Publishers, 1981.
5. K. Rose, E. Gurewitz, and G. Fox, "A deterministic annealing approach to clustering," *Pattern Recognition Letters*, vol. 11, pp. 589–594, 1990.
6. V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.
7. D. M. Hawkins, *Identification of outliers.* Springer, 1980, vol. 11.
8. E. M. Knorr and R. T. Ng, "Finding intensional knowledge of distance-based outliers," in *VLDB*, vol. 99, 1999, pp. 211–222.
9. P. J. Huber, *Robust Statistics.* New York: John Wiley and Sons, 1981.
10. V. Balasubramanian, S.-S. Ho, and V. Vovk, *Conformal Prediction for Reliable Machine Learning: Theory, Adaptations and Applications.* Newnes, 2014.
11. A. Keller, "Fuzzy clustering with outliers," in *Fuzzy Information Processing Society, 2000. NAFIPS. 19th International Conference of the North American*, 2000, pp. 143–147.
12. R. N. Davé and R. Krishnapuram, "Robust clustering methods: a unified view," *IEEE Transactions on Fuzzy Systems*, vol. 5, no. 2, pp. 270–293, 1997.
13. O. Nasraoui and C. Rojas, "Robust clustering for tracking noisy evolving data streams." in *Proceedings of the 2006 SIAM International Conference on Data Mining.* SIAM, 2006, pp. 619–623.
14. T. Martinetz, S. Berkovich, and K. Schulten, "'Neural gas' network for vector quantization and its application to time-series prediction," *IEEE Transactions on Neural Networks*, vol. 4, no. 4, pp. 558–569, 1993.
15. H. Kushner and G. Yin, *Stochastic Approximation and Recursive Algorithms and Applications*, ser. Stochastic Modelling and Applied Probability. Springer New York, 2003.
16. S. Ridella, S. Rovetta, and R. Zunino, "Plastic algorithm for adaptive vector quantization," *Neural Computing and Applications*, vol. 7, no. 1, pp. 37–51, 1998.
17. K.-A. Yoon, O.-S. Kwon, and D.-H. Bae, "An approach to outlier detection of software measurement data using the k-means clustering method," in *Empirical Software Engineering and Measurement, 2007. ESEM 2007. First International Symposium on.* IEEE, 2007, pp. 443–445.
18. H. Robbins and S. Monro, "A stochastic approximation method," *Ann. Math. Statist.*, vol. 22, no. 3, pp. 400–407, 09 1951.
19. G. Ditzler and R. Polikar, "Incremental learning of concept drift from streaming imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 10, pp. 2283–2301, 2013.
20. A. Abdullatif, F. Masulli, S. Rovetta, "Layered ensemble model for short-term traffic flow forecasting with outlier detection," *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI) (IEEE RTSI 2016)*, pp. 1–6, 2016.