Density estimation on data streams : an application to Change Detection

Alexis Bondu*, Benoît Grossin*,
Marie-Luce Picard*

*EDF R&D ICAME/SOAD, 1 avenue du Général de Gaulle, 92140 Clamart.
firstname.name@edf.fr

Abstract. In recent years, the amount of data to process has increased in many application areas such as network monitoring, web click and sensor data analysis. Data stream mining answers to the challenge of massive data processing, this paradigm allows for treating pieces of data on the fly and overcoming data storage. The detection of changes in a data stream distribution is an important issue. This article proposes a new schema of change detection : i) the summarization of the input data stream by a set of micro-clusters; ii) the estimate of the data stream distribution exploiting micro-clusters; iii) the estimate of the divergence between the current estimated distribution and a reference distribution; iv) diagnostic step through the contribution of each predictive variable to the overall divergence between both distributions. Our schema of change detection is applied and evaluated on artificial data streams.

1 Introduction

In recent years, the amount of data to process has increased in many application areas such as network flows, web click and sensor data analysis. Data stream mining indicates algorithms which process tuples [1] on the fly : when they are emitted, without storing them. The processing of tuples should be as fast as possible which allows for managing high rate data streams. An important issue in processing data streams is detecting changes in underlying distribution that is generated by tuples. The designing of change detection schemes which are general, scalable and statistically relevant is a great challenge.

A change in the underlying distribution can be interpreted into different ways : i) the observed phenomenon is naturally drifting due to a change in some hidden *context* (Widmer and Kubat, 1996) which is not explicitly given by predictive features; ii) an abnormal change is taking place in the observed system. Distinguish the two cases is a very difficult issue which requires expertise on the application. In this article we assume an expert, who well knows the observed data stream, may rule on the interpretation of detected changes.

An overview of the main change detection approaches is given by A. Dries (Dries and Rückert, 2009) : *Change detection in the distribution of tuples can be considered as a statistical hypothesis test which involves two samples of multidimensional tuples. Such problems are studied in the statistical literature. The Wald-Wolfowitz and Smirnov tests was generalized*

[1] The term “tuple” refers to a piece of data which is emitted from the input stream.

to multidimensional data sets in (Friedman and Rafsky, 2006). Later, approaches based on nearest-neighbor analyses (Hall, 2002) or distance between density estimates (Anderson et al., 1994) have been developed. Most recently, statistics based on maximum mean discrepancy for universal kernels have become popular (Gretton et al., 2006). A range of statistical work on abrupt change detection have been done (Basseville and Nikiforov, 1993; Desobry and Davy, 2003) .

In this article, a new schema of change detection is proposed (see Figure 1). This schema is composed by four successive steps. In Section 2 the input data stream is summarized by a micro-clustering algorithm. This first step is necessary because of the high rate of the input data stream, in practice all tuples can not be processed in real time. The "Denstream" algorithm (Feng Cao et al., 2006) has the ability to summarize dense areas of the input space and to forget the old tuples through a time-based weighting. We propose a simple way to tune this algorithm in terms of durations. In Section 3 a new variant of Parzen window is proposed and it is used to estimate the underlying distribution of the data stream. This density estimator exploits the summary of the input data stream instead of tuples. This step is periodically repeated with a lower rate than the emission of tuples from the data stream. Section 4 shows how the distance between the current estimated distribution and a reference distribution can be evaluated by the Kullback-Leibler divergence. This measure allows for sending an alarm to the expert when both distributions are significantly different. The last step of our schema consists in a diagnostic which is given to the expert to help him understand the causes of the detected anomaly. The contribution of each variable to the overall distance between both distributions is evaluated owing to a new proposed criterion.

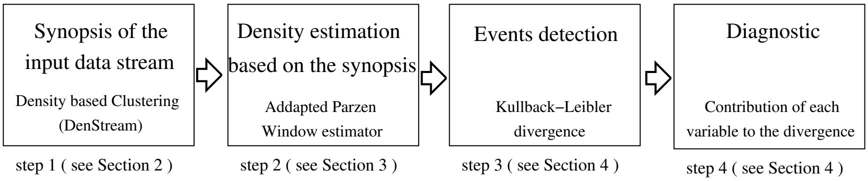
Synopsis of the input data stream
Density based Clustering (DenStream)
step 1 (see Section 2)
Density estimation based on the synopsis
Addapted Parzen Window estimator
step 2 (see Section 3)
Events detection
Kullback−Leibler divergence
step 3 (see Section 4)
Diagnostic
Contribution of each variable to the divergence
step 4 (see Section 4)

FIG. 1 – *global schema for change detection in the input data stream distribution.*

Finally, our approach is applied and evaluated on two artificial data streams in Section 5. Possible industrial applications of our schema and future works are discussed in Section 6.

2 Summarization of the input data stream

In the data stream paradigm, emitted tuples can not be exhaustively stored and processed due to the high rate of the input stream. This section presents the summarization of the input data stream which is a preliminary step in the change detection processing. Our approach exploits the "Denstream" algorithm (Feng Cao et al., 2006) to summarize the data stream : a time based weighting is applied on a set of micro-clusters.

2.1 Weighted data stream

Tuples are progressively emitted from the data stream and are weighted with regard to their *age*. More precisely, tuples (denoted by x_i) are defined in $\mathbb{R}^k$ and are characterized by the vector $\{x_i^1, x_i^2 ... x_i^k\}$. Each tuple x_i is emitted at the instant $(t_{current} - \alpha_i)$ with α_i denoting the age of x_i. At current time $t_{current}$ each tuple is weighted by $w_i = 2^{-\lambda.\alpha_i}$, where λ is a fading parameter belonging to the interval $]0, \infty]$. The higher the value of λ, the lower the importance of the historical data compared to more recent data. In this article, N denotes the total number of emitted tuples at $t_{current}$.

Let W_N be the overall weight of the data stream at the instant $t_{current}$ when N tuples were emitted. We have $W_N = \sum_{i=1}^{N} 2^{-\lambda.\alpha_i}$ and $W_{N+1} = \sum_{i=1}^{N} 2^{-\lambda(\alpha_i + \Delta_t)} + 1$, with Δ_t corresponding to the elapsed time between the emission of the two last tuples. Under the hypothesis that the rate of the data stream is constant, the overall weight is recursively defined as $W_{N+1} = 1 + 2^{-\lambda.\Delta_t} \times W_N$. This *"arithmetic geometric"* serie converges[2] to $\lim_{N \to +\infty} W_N = \frac{1}{1 - 2^{-\lambda \Delta_t}}$.

2.2 Micro-clusters

A set of micro-clusters aims at summarizing the input data stream keeping information on the density distribution. This synopsis is maintained in memory at any time. The *"jth"* micro-cluster $mc_j(c_j, r_j, w_j)$ is defined by : i) a weight w_j that corresponds to the sum of the weights of tuples belonging to the cluster (denoted by $x_{1j}, x_{2j} ... x_{n_j j}$) with $w_j = \sum_{i=1}^{n_j} 2^{-\lambda \alpha_{ij}}$; ii) the center c_j that is a vector corresponding to the weighted barycenter of examples with $c_j = \frac{1}{w_j} \sum_{i=1}^{n_j} w_{ij} x_{ij}$; iii) the radius r_j that is a vector corresponding to the weighted standard deviation with $r_j = \frac{1}{w_j} \sqrt{\sum_{i=1}^{n_j} w_{ij} . d(x_{ij}, c_j)^2}$ with the Euclidean distance denoted by $d()$.

The *"age"* of tuples increases when a new tuple is emitted such as $\alpha_i \leftarrow \alpha_i + \Delta_t$, the elapsed time between two tuple Δ_t is considered as constant. The new tuple is affected to the closer micro-cluster. The set of micro-clusters is maintained owing to an iterative process. The weights of micro-clusters are maintained through the following successive steps :

1. the aging of all micro-clusters such as $w_j^{(1)} \leftarrow w_j . 2^{-\lambda \Delta_t} \quad \forall j \in [1, C]$;
2. the increase of the weight of the micro-cluster j^* where the emitted tuple is affected such as $w_{j^*}^{(2)} \leftarrow w_{j^*}^{(1)} + 1$.

Two clusters features are required to maintain the center and the radius of micro-clusters (Zhang et al., 1996). Let CF_j^1 [*respectively* CF_j^2] be a k-dimentional vector storing, for each variable, the weighted sum of coordinates [*respectively* the sum of squared coordinates] of examples belonging to the *"jth"* micro-cluster : $CF_j^1 = \sum_{i=1}^{n_j} w_{ij} x_{ij}$ [*respectively* $CF_j^2 = \sum_{i=1}^{n_j} w_{ij} x_{ij}^2$]. c_j and r_j are maintained as follows : $c_j = \frac{CF^1}{w_j}$ and $r_j = \sqrt{\frac{|CF_j^2|}{w_j} - \left(\frac{|CF_j^1|}{w_j}\right)^2}$

2.3 The Denstream approach

The Denstream approach handles two kinds of micro-cluster corresponding to different functions. The set of *"potential-micro-clusters"*, denoted by mc_p, summarizes significant in-

[2] In this case, the condition $|2^{-\lambda \Delta_t}| \leq 1$ is always satisfied.

formation from the data stream. Micro-clusters exceeding a minimal weight are considered as representing significant information. The set of *"outlier-micro-clusters"*, denoted by mc_o, consists in a buffer keeping insignificant information from the data stream. The intuition is the following : a slight micro-cluster (under the minimal weight) can grow if the density distribution of the data stream is changing. The objective is to keep insignificant information to early detect new dense areas in data stream. Two constraints are applied on micro-clusters : i) micro-clusters of which the weight decreases below a minimum weight (denoted by μ) are deleted; ii) a new tuple is merged into its nearest micro-cluster if its updated radius r_j^* is less than a maximum standard deviation (denoted by ϵ). These constraints ensure the micro-clusters represent dense areas of the space $\mathbb{R}^k$ where tuples have appeared recently. A pruning strategy is implemented by the “Denstream” algorithm. This strategy aims at regulating the memory space necessary to store the two sets of micro-clusters mc_p and mc_o.

How we tune parameters in terms of durations : We assume our change detection approach is exploited by an expert who well knows phenomena embedded into the data stream. The “Denstream” algorithm involves several parameters (λ, μ and ϵ) that may be difficult to adjust by the expert. In this paragraph, consideration is made about how to adjust parameters in a understandable way. The expert knows the length of validity of emitted tuples and he is able to set up a half-live period [3] (denoted by $\Delta_t^{HalfLive}$). The fading parameter can be determined in a second time such as $\lambda = -\log_2(\frac{1}{2})/\Delta_t^{HalfLive}$. We demonstrate that the parameter μ which represents the minimum weight of clusters is bounded as follows :

$$\frac{2^{-\lambda\Delta_t^{ClusMin}}}{1-2^{-\lambda\Delta_t}} > \mu \geq \frac{1}{1-2^{-\lambda\Delta_t^{ClusMax}}}$$

Let $\Delta_t^{ClusMax}$ be the span of time beyond the arrival of a new tuple into a p-micro-cluster that is not suffisant to keep the *“potential”* status. For any p-micro-cluster, we have $w_j.2^{-\lambda\Delta_t^{ClusMax}} + 1 < \mu$ and $w_j < \mu$. At the end we obtain $\mu > 1/(1-2^{-\lambda\Delta_t^{ClusMax}})$. Let $\Delta_t^{ClusMin}$ be the minimum span of time that an un-updated p-micro-cluster must be maintained in the synopsis. For any p-micro-cluster, we have $w_j.2^{-\lambda\Delta_t^{ClusMin}} > \mu$. The weight of a p-micro-cluster is inferior or equal to the overall weight of the data stream, thus we have $W.2^{-\lambda\Delta_t^{ClusMin}} > \mu$. At the end we obtain $(2^{-\lambda\Delta_t^{ClusMin}})/(1-2^{-\lambda\Delta_t}) > \mu$. In this article, we adopt the same choice than in (Feng Cao et al., 2006) where the authors define the pruning time period T as the minimum of $\Delta_t^{ClusMax}$. We consider that T and $\Delta_t^{HalfLive}$ are given by the expert. In these conditions μ can be expressed as follows :

$$\mu = \frac{1}{1-2^{-\frac{T}{\Delta_t^{HalfLive}}}}$$

A new micro-cluster is created when the maximum standard deviation ϵ is reached in the nearest micro-cluster of an emitted tuple. Intuitively, the value of ϵ influences the number of potential micro-clusters which are maintained in memory. The tuning of ϵ is an issue because the overall standard deviation of the input data stream is not known in the general case. In this article, we assume the overall standard deviation to be known by the expert and ϵ is adjusted as a proportion of the overall standard deviation.

[3] The value of w_i is periodically divided by 2.

Notations :

- mc_p the set of potential-micro-cluster;
- mc_o the set of outlier-micro-cluster;
- μ the minimum weight of a potential-micro-cluster;
- ϵ the maximum standard deviation of a potential-micro-cluster;
- T the pruning time period.

```
Repeat
    Get the next point $x_{i+1}$ from the data stream.

    /*merging procedure*/
    Try to merge $x_{i+1}$ to its nearest p-micro-cluster, denoted by $mc_p^\diamond(c_p^\diamond, r_p^\diamond, w_p^\diamond)$.
    Let $r_p^{*\diamond}$ be the new radius of $mc_p^\diamond$.
    If $r_p^{*\diamond} \leq \epsilon$ then
        Merge $x_{i+1}$ into $mc_p^\diamond$, and update $c_p^\diamond, r_p^\diamond, w_p^\diamond$.
    else
        Try to merge $x_{i+1}$ to its nearest o-micro-cluster, denoted by $mc_o^\diamond(c_o^\diamond, r_o^\diamond, w_o^\diamond)$. Let $r_o^{*\diamond}$
        be the new radius of $mc_o^\diamond$.
        If $r_o^{*\diamond} \leq \epsilon$ then
            Merge $x_{i+1}$ into $mc_o^\diamond$, and update $c_o^\diamond, r_o^\diamond, w_o^\diamond$.
            If $w_o^\diamond > \mu$ then
                Remove $mc_o^\diamond$ from outlier-buffer and create a new p-microcluster by $mc_o^\diamond$.
            end If
        else
            Create a new o-micro-cluster by $x_{i+1}$ and insert it into the outlier-buffer.
        end If
    end If

    /*pruning procedure*/
    If The pruning periode $T$ is elapsed then
        For each p-micro-cluster $mc_p(c_p, r_p, w_p)$ do
            If $w_p < \mu$ then
                Delete $mc_p$
            end If
        end For
        For each o-micro-cluster $mc_o(c_o, r_o, w_o)$ do
            If $w_o < \frac{2^{-\lambda(t_o+T)}-1}{2^{-\lambda T}-1}$ then
                Delete $mc_o$
            end If
        end For
    end If
until the data stream exists
```

Algorithm 1: Data stream synopsis by Denstream approach

3 Density estimation exploiting the synopsis

This section shows how the synopsis of the input data stream is exploited to estimate the density of data. We modify the Parzen window density estimator (Parzen, 1962) to exploit micro-clusters instead of tuples. Subsection 3.1 presents the classical Parzen window in the case of gaussian kernel, in subsection 3.2 this density estimator is adapted to micro-clusters.

3.1 Parzen Windows

Among the large range of models able to estimate data density from a set of tuples, Parzen window provided with a gaussian kernel (Shawe-Taylor and Cristianini, 2004) has the advantage of requiring few parameters. Equation 1 corresponds to the "output" of this predictive model which is an estimate of the probability to observe the tuple $x \in \mathbb{R}^k$. $K(x - x_i)$ is a kernel function evaluating the proximity between the tuples x and x_i, this function is summed over all emitted tuples.

$$\hat{P}(x) = \frac{1}{N} \sum_{i=1}^{N} K(x - x_i) \tag{1}$$

In practice, the kernel function must be specified. Equation 2 corresponds to the "output" of a Parzen window provided with a gaussian kernel [4]. In this case, the Parzen window involves a single parameter that is σ : the standard deviation of the gaussian kernel.

$$K(x - x_i) = \frac{1}{\left(\sigma\sqrt{2\pi}\right)^k} \exp^{-\frac{d(x,x_i)^2}{2.\sigma^2}} \tag{2}$$

Figure 2 illustrates the estimate of $P(x)$ by a Parzen windows estimator. Gaussian kernels are positioned on each tuple, next they are summed and normalized. In this case, each tuple contributes to the estimate of $P(x)$.

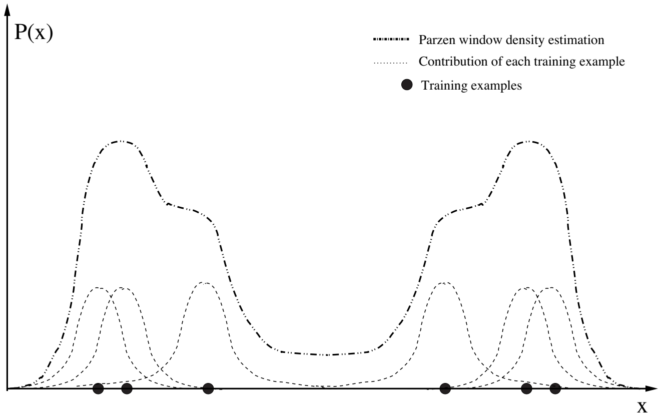

FIG. 2 – *Estimation of the data stream distribution owing to a Parzen windows.*

[4] We consider the standard deviation of the gaussian kernel is constant over all dimension of the input space $\mathbb{R}^k$.

3.2 Our modified Parzen windows

In this subsection, the Parzen window density estimator is adapted to exploit the set of potential micro-clusters instead of tuples. The distribution of $P(x)$ is approximated by Equation 3 :

$$\hat{P}^*(x) = \frac{1}{C.W} \sum_{j=1}^{C} \frac{\omega_j}{\sqrt{2\pi\left(\delta^2 + r_j^2\right)}^k} \exp^{-\frac{d(x,c_j)^2}{2\left(\delta^2 + r_j^2\right)}} \tag{3}$$

- W denotes the total weight of the data stream;
- C denotes the number of potential micro-clusters summarizing the data stream;
- ω_j denotes the weight of the *jth* micro cluster;
- c_j denotes the barycenter of weighted points belonging to the *jth* micro cluster;
- r_j denotes the standard deviation of weighted points belonging to the *jth* micro cluster;
- δ denotes a flatness parameter which plays the same role than σ in Equation 2.

Each observed tuple is supposed to be the most probable of an unobserved set of tuples which is normally distributed with a standard deviation equal to δ. Under this assumption, the law of total variance gives the variance of the *"jth"* potential micro-cluster as the sum of the within-variance δ^2 and the between-variance r_j^2. In Equation 3 gaussian kernels are positioned on the center of each potential micro-cluster. Then gaussian kernels are summed and normalized regarding the number of potential micro-clusters and the overall weight of the data stream.

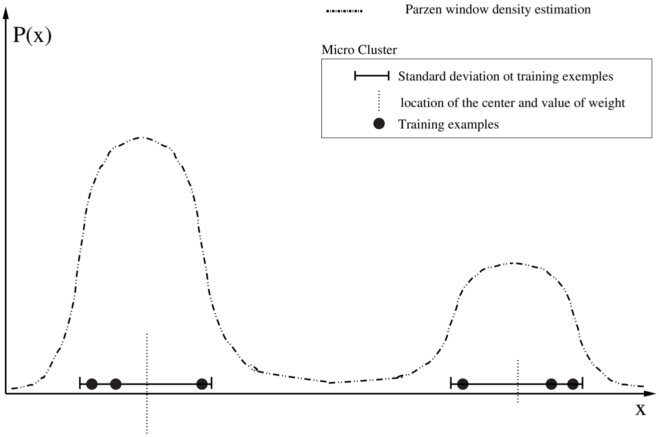

FIG. 3 – *Influence of the weight of micro-cluster on the density estimate*

Figure 3 illustrates the estimate of $P(x)$ by our modified Parzen windows. On this figure, the set of tuples is split into two potential micro-clusters which radius are symbolized by a horizontal full line and weights are symbolized by a vertical dashed line. On the one hand, the estimate of $P(x)$ is less accurate than on Figure 2 because of the loss of each tuple location. On the other hand, this estimate takes into account the weight of each cluster. The estimate of the distribution of $P(x)$ changes over time due to the aging of micro-clusters. If no changes occur into the underlying distribution, the tuples whose weight decreases are replaced by new emitted ones : in this case, the estimate of $P(x)$ will not change. Otherwise, non-replaced tuples into a micro-cluster engender a decrease in its weight : then a change of the estimate of $P(x)$ is observed.

4 Change detection and diagnostic

We assume that an anomaly that occurs in the input data stream results in a change in distribution of $P(x)$. A reference distribution is set up after a learning period without anomalies to detect . The expert examines the input data stream to ensure no anomaly has occurred during this period. Then the current estimate of the distribution of $P(x)$ is compared to the reference distribution owing to the Kullback-Leibler divergence (Hershey and Olsen, 2007) shown by Equation 4. The Kullback-Leibler divergence has interesting statistical properties, in particular finding parameters of a statistical model maximizing the likelihood is analogous to finding parameters minimizing the divergence (Eguchi and Copas, 2006). The Kullback-Leibler divergence generalizes standard statistical tests as the *t-test* and the χ^2 : i) the *t-test* is equivalent to the Kullback-Leigler divergence between two normal distributions; ii) the χ^2 function is the first term in the Taylor expansion of the Kullback-Leigler divergence. In our change detection schema, an alarm is sent to the expert when the divergence between distributions P_{ref} and $\hat{P}^*$ reaches a fixed threshold.

$$KL < P_{ref}(x) \| \hat{P}^*(x) >= - \int_{\mathbb{R}^k} P_{ref}(x) \log \frac{P_{ref}(x)}{\hat{P}^*(x)} dx \tag{4}$$

A diagnostic is required by the expert in order to give a proper response to the alarm. The diagnostic phase aims at evaluating the contribution of each variable to the divergence between P_{ref} and $\hat{P}^*$. Thus, the expert is informed which predictive features are involved in the detected change. The contribution of variables is evaluated by Equation 5. Let KL^i_{minus} be the Kullback-Leigler divergence evaluated in a $(k-1)$ dimensional subspace after exclusion of the *"ith"* variable. When the contribution of the"lth" variable is evaluated, KL^l_{minus} is compared to the sum of KL_{minus} over all variables, then the contribution is normalized.

$$Contrib(l) = \frac{\left(\sum_{i=1}^{k} KL^i_{minus}\right) - KL^l_{minus}}{\sum_{i=1}^{k} KL^i_{minus}} \tag{5}$$

The contribution of each variable aims at assisting the expert to rule on the interpretation of the detected change. In practice, the expert may be allowed to update the reference distribution with the current distribution if the detected change is not an abnormality. This update constitutes one possible way to take into account natural drift of the observed phenomenon.

5 Experiments

In this section, our schema of change detection is applied on two artificial data streams. The objective is to evaluate the ability of our schema to detect two different types of changes : i) a change in the mean of a normal distribution; i) a change in the standard deviation of a normal distribution.

5.1 Experimental protocol

The two considered artificial data streams share the same temporal structure. Each second, one tuple is drawn from an underlying distribution which changes over time. Figure 4 shows how the underlying distribution evolves. The 2000 first tuples are emitted from the *"initial distribution"* which represents the usual operation. At this moment the reference distribution is set up : our schema of change detection starts. Between 4000 and 6000 seconds, the underlying distribution progressively moves from the *"initial"* to the *"modified"* distribution. Then 2000 tuples are emitted from the *"modified"* distribution. Between 8000 and 10000 seconds, the underlying distribution progressively returns to its *"initial"* state. At last, 2000 tuples are emitted from the *"initial"* distribution.

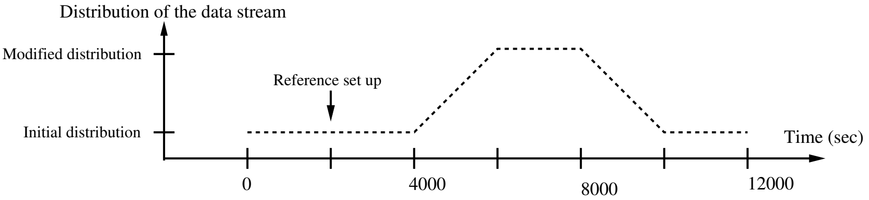

FIG. 4 – *Temporal structure of both artificial data streams.*

In our experiments tuples are defined in $\mathbb{R}^2$. The *"initial"* and *"modified"* distributions are defined on Table 1 for both artificial data streams. These normal distributions are denoted by $\mathcal{N}(m, v)$, where m is a two-dimensional vector corresponding to the mean and v is the covariance matrix.

	initial distribution	**modified distribution**
Data stream 1 : change in mean	$\mathcal{N}\left(\begin{matrix}0 & 0\end{matrix}, \begin{matrix}1 & 0\\0 & 1\end{matrix}\right)$	$\mathcal{N}\left(\begin{matrix}4 & 8\end{matrix}, \begin{matrix}1 & 0\\0 & 1\end{matrix}\right)$
Data stream 2 : change in standard deviation	$\mathcal{N}\left(\begin{matrix}0 & 0\end{matrix}, \begin{matrix}1 & 0\\0 & 1\end{matrix}\right)$	$\mathcal{N}\left(\begin{matrix}0 & 0\end{matrix}, \begin{matrix}4 & 0\\0 & 9\end{matrix}\right)$

TAB. 1 – *Definition of "initial" and "modified" distributions for both artificial data streams.*

Our schema of change detection involves several parameters which must be fixed before the experiments. The "Denstream" algorithm which summarize the input data stream (see Section 2.3) is parametrized by $\epsilon = 0.1$, $\Delta_t^{HalfLive} = 300s$ and $T = 1000s$. The flatness parameter of our density estimator (see Section 3.2) is fixed by $\delta = 1$.

5.2 Results

Figure 5 presents the results of our experiments, the left chart [*respectively* the right chart] shows the detection of a change in the mean [*respectively* in the standard deviation] of the underlying distribution (described on Table 1). On both chart, the horizontal axis corresponds to the time and starts when the *"reference"* distribution is set up (at $t = 2000$). The vertical axis corresponds to the divergence between the *"reference"* and the *"current"* distributions. The contribution of each variable to the divergence is also symbolized by colors.

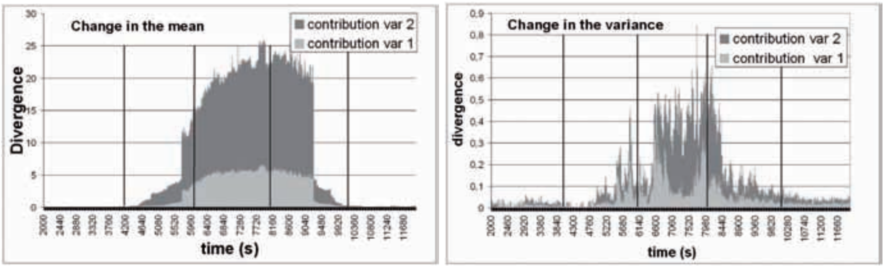

FIG. 5 – *Change detection in the distribution of both artificial data streams.*

The first artificial data stream involves a change in the mean of a normal distribution (left chart on Figure 5). In this case the change which occur when $t \in [4000, 6000]$ is early detected, indeed the divergence increases significantly once $t = 4500$. Between 6000 and 8000 seconds, the divergence increases to its maximum ($KL = 25$) and the contributions well estimate the move of the underlying distribution on both dimensions. The return to the initial underlying distribution ($t \in [8000, 10000]$) is detected relatively late. The divergence keeps high values until $t = 9000$ and drops sharply after. This behavior can be explained be the span of time necessary to delete useless potential micro-clusters in the summary of the input data stream.

The second artificial data stream involves a change in the standard deviation of a normal distribution (right chart on Figure 5). In this case, change detection is less distinct than previously : i) the divergence strongly varies over time and does not stabilize; ii) the divergence reaches a small maximum value ($KL = 0.6$). However, the first change in the underlying distribution is detected : the divergence increases from $t = 4800$ to $t = 6000$. Between 6000 and 8000 seconds, the divergence reaches its maximum value that is consistent with the structure of the input data stream. During this period of time, the contribution of the second variable tends to be more important than the first variable. At last, the return to the initial underlying distribution is detected in time.

These experiments show the interest of our approach for the detection of progressive drift in the underlying distribution. Others conclusive tests have been done on abrupt changes. In this case, a very short latency is observed because of the span of time necessary to create new potential micro-clusters. We notice the tuning of the fading parameter λ is sensitive and raises the dilemma between reducing the latency of detections and ensuring the statistical significance of the distribution estimate. Adjusting the parameter λ could be less sensitive in practice, if the pace of changes is known in advance by the expert.

6 Conclusion and perspectives

This article proposes a new schema of change detection in the underlying distribution of a data stream. Our approach is composed by four successive steps. First, the input data stream is summarized by a set of micro-clusters owing to the “Denstream” algorithm (Feng Cao et al., 2006), thus data streams with high rate can be processed. The “Denstream” algorithm has the ability to summarize dense areas of the input space and to forget the old tuples through a time-based weighting. We propose a simple way to tune the parameters of this algorithm in terms of durations. The second step consists in an estimation of the underlying distribution exploiting the summary of the data stream : a new variant of the Parzen window estimator (Parzen, 1962) is proposed. Then, the drift of the current estimated distribution is evaluated in comparison to a reference distribution : the Kullback-Leibler divergence is exploited (Hershey and Olsen, 2007). At the end, a diagnostic is given by a new criterion which estimates the contribution of each variable to the overall distance between both distributions. In practice, this last step could be helpful to understand the causes of a detected anomaly and respond to it in a proper way.

Since our schema of change detection involves a density estimator, the probability of each emitted tuple could be estimated by the current Parzen window. This information should be exploited to early detect abrupt changes in the underlying distribution, under the assumption that a change causes the emission of an improbable sequence of tuples. In this case the main difficulty is to manage the temporal dependency of emitted tuples, future works will study this point. An other aspect on which we are working on is the theoretical quantification of the information that is lost using micro-clusters instead of tuples, when the distribution of the data stream is estimated.

The “Denstream” algorithm handles the variance of each micro-cluster as a single scalar value, this represents a loss of substantial information. For instance, the covariance matrix of emitted tuples could be maintained online for each micro-cluster. In futur works, we will study the online maintaining of the covariance matrix and higher statistical moments, and we will use these new pieces of information to estimate more precisely the distribution of tuples.

Our schema of change detection was favorably evaluated on two artificial data streams. In future works, others experiments will evaluate the influence of increasing the dimension of the input space on the ability of our schema to detect changes. At last, our schema will be applied on real data streams. In particular, we aim at improving the preventive maintenance in power plants thanks to the detection of unusual events. More generally, our schema of change detection could be exploited in many applications areas. For instance, the NASA began a large research program in Integrated Vehicle Health Management of which goal is to automatically detect, diagnose, predict, and mitigate adverse events during the flight of an aircraft (Srivastava, 2009). The early detection of anomalies on sensor data streams represents a real interest for the scientific community.

References

Anderson, N. H., P. Hall, and D. M. Titterington (1994). Two-sample test statistics for measuring discrepancies between two multivariate probability density functions using kernel-based density estimates. *Journal of Multivariate Analysis 50*(1), 41–54.

Basseville, M. and I. V. Nikiforov (1993). *Detection of Abrupt Changes: Theory and Application*. Prentice-Hall.

Desobry, F. and M. Davy (2003). Support Vector-Based Online Detection of Abrupt Changes. In *Proc. IEEE ICASSP, Hong Kong*, pp. 872–875.

Dries, A. and U. Rückert (2009). Adaptive Concept Drift Detection. In *SIAM Conference on Data Mining*, pp. 233–244.

Eguchi, S. and J. Copas (2006). Interpreting Kullback-Leibler divergence with the Neyman-Pearson lemma. *Journal of Multivariate Analysis 97*(9), 2034–2040.

Feng Cao, F., M. Ester, W. Qian, and A. Zhou (2006). Density-based clustering over an evolving data stream with noise. In *SIAM Conference on Data Mining*, pp. 328–339.

Friedman, J. and L. Rafsky (2006). Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistic 7*(4), 697–717.

Gretton, A., K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and S. A. J. (2006). A Kernel Method for the Two-Sample-Problem. In *NIPS*, pp. 513–520. MIT Press.

Hall, P. (2002). Permutation tests for equality of distributions in high-dimensional settings. *Biometrika 89*(2), 359–374.

Hershey, J. R. and P. A. Olsen (2007). Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In *ICASSP : IEEE International Conference on Acoustics, Speech and Signal Processing.*, Volume 4, pp. 317–320.

Parzen, E. (1962). On estimation of a probability density function and mode. *Annals of Mathematical Statistics 33*, 1065–1076.

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Srivastava, A. (2009). Data mining at NASA : from theory to applications. In *Proc. KDD, Paris*, pp. 7–8.

Widmer, G. and M. Kubat (1996). Learning in the Presence of Concept Drift and Hidden Contexts. *Machine Learning 23'*(1), 69–101.

Zhang, T., R. Ramakrishan, and M. Livny (1996). BIRCH : An Efficient Data Clustering Method for very Large Databases. *Sigmod Rec. 25*(2), 103–114.

Résumé

Ces dernières années, la quantité de données à traiter à considérablement augmentée dans de nombreuses applications. La fouille de flux de données répond au défi des données massives par des traitements à la volée qui requièrent une capacité de stockage raisonnable. La détection de changements dans la densité de probabilité d'un flux est une question importante. Cet article propose un nouveau schéma de détection de changement qui se compose de quatre étapes successives : i) le résumé du flux par un ensemble de micro-clusters; ii) l'estimation la densité de probabilité du flux grâce aux micro-clusters; iii) l'estimation de la divergence entre la densité estimée à l'instant courant et une densité de référence; iv) un diagnostic estimant la contribution de chaque variable descriptive à la divergence globale qui sépare les deux densités. Notre schéma de détection de changement est finalement appliqué et évalué sur deux flux de données artificiels.