Expert Systems With Applications 117 (2019) 90–102

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

On learning guarantees to unsupervised concept drift detection on data streams

Rodrigo F. de Mello[a,*], Yule Vaz[a], Carlos H. Grossi[b], Albert Bifet[c]

[a] University of São Paulo, Institute of Mathematics and Computer Science, Department of Computer Science, Av. Trabalhador São-carlense, São Carlos, SP 400, Brazil

[b] University of São Paulo, Institute of Mathematics and Computer Science, Department of Mathematics, Av. Trabalhador São-carlense, São Carlos, SP 400, Brazil

[c] LTCI, Télécom ParisTech – Data, Intelligence and Graphs Team, Office: C201-2, 46 rue Barrault, Paris Cedex 13 75634, France

ARTICLE INFO

Article history:
Received 21 June 2018
Revised 31 August 2018
Accepted 31 August 2018
Available online 21 September 2018

Keywords:
Data streams
Concept drift
Algorithmic stability
McDiarmid's inequality

ABSTRACT

Motivated by the Statistical Learning Theory (SLT), which provides a theoretical framework to ensure when supervised learning algorithms generalize input data, this manuscript relies on the Algorithmic Stability framework to prove learning bounds for the unsupervised concept drift detection on data streams. Based on such proof, we also designed the Plover algorithm to detect drifts using different measure functions, such as Statistical Moments and the Power Spectrum. In this way, the criterion for issuing data changes can also be adapted to better address the target task. From synthetic and real-world scenarios, we observed that each data stream may require a different measure function to identify concept drifts, according to the underlying characteristics of the corresponding application domain. In addition, we discussed about the differences of our approach against others from literature, and showed illustrative results confirming the usefulness of our proposal.

1. Introduction

The Statistical Learning Theory (SLT) provides the theoretical foundation and learning guarantees for supervised machine learning algorithms (Vapnik, 1998; 1995). SLT considers the Bias-Variance dilemma to study and analyze algorithm biases in terms of sample sizes. According to the SLT, one can study the complexity of such bias, also referred to as the space of admissible functions which classifiers are selected from, in order to define an adequate supervised algorithm to address some classification or regression task. This theoretical foundation ensures supervised learning in different scenarios and application domains (de Mello & Ponti, 2018; Vapnik, 1995).

To counterpose, the area of unsupervised learning does not count on a similar theoretical framework to provide learning guarantees, consequently results typically rely on internal or external indices (Rendón et al., 2011). Internals are directly computed on the resulting partition, usually measuring how compact individual clusters are, and how far such groups are from one other. The more compact they are, the best is the resulting measure. The far groups are from each other, the best is the second measure. Both measure may be individually considered or combined into a single index. The external indices assume supervision labels are available to proceed with the final analysis, what is not completely fair in terms of unsupervised learning, given it is assumed no label is present and only the structural data organization in some input space is analyzed. Given such scenario, we conclude unsupervised learning lacks in terms of theoretical foundation to provide learning guarantees, therefore, depending on the scenario, results may be obtained simply by chance.

Besides such limitation, unsupervised learning has been specially considered to address data-stream concept drift detection, given the complexity involved in supervising observations, specially when data is collected at high frequencies (Faria, Gama, & Carvalho, 2013; Hayat & Hashemi, 2010; Masud et al., 2010; Sethi & Kantardzic, 2017; Sethi, Kantardzic, & Hu, 2016; Spinosa, de Leon F. de Carvalho, & Gama, 2007). Data streams are open-ended sequences of uni or multidimensional observations collected from some phenomenon, such as the temperature of a given world region, flood sensing, motor and cognitive development, etc. (Agarwal, 1995; Metzger, 1997; Rios, Parrott, Lange, & de Mello, 2015). Observations are here assumed to be generated from some process, such as a stochastic or a deterministic dynamical system (Kantz & Schreiber, 2003). Those generating processes

* Corresponding author.
E-mail addresses: mello@icmc.usp.br (R.F. de Mello), yule.vaz@usp.br (Y. Vaz), grossi@icmc.usp.br (C.H. Grossi), albert.bifet@telecom-paristech.fr (A. Bifet).

https://doi.org/10.1016/j.eswa.2018.08.054
0957-4174/

or their parameters may change along time due to some other phenomena interacting or acting on them, such as a medicine impacting someone's blood pressure, or the movement of air masses affecting local temperatures. Data behavior changes are here referred to as concept drifts (da Costa, Duarte, Vallim, & de Mello, 2017) and they allow to point out decisive moments in which some system or phenomenon should be studied in order to comprehend anomalous behaviors. For instance, a stream may start containing data observations produced by the following sinusoidal function $\sin(2\pi x)$ and, then, change to some other process such as: i) a Normal distribution with a given mean and standard deviation, causing an abrupt data behavior change; or ii) to another sinusoidal function $\sin(2.1\pi x)$ with a small frequency change (from 2 to 2.1). Both scenarios are indeed considered throughout our study.

Besides the current unsupervised algorithms found in literature report fair concept-drift-detection results (Faria et al., 2013; Hayat & Hashemi, 2010; Masud et al., 2010; Sethi & Kantardzic, 2017; Sethi et al., 2016; Spinosa et al., 2007), they do not provide theoretical learning guarantees for concept drift detection. This drawback has motivated this manuscript, which considers the Algorithmic Stability (Devroye, Györfi, & Lugosi, 1996) to quantify relevant divergences along the data stream collection, based on the assumption that drifts occur whenever the current learning model [1] stops satisfying the uniform stability property, as theoretically ensured by Concentration Inequalities (more details in Section 3). As discussed throughout this manuscript, our framework relies on a given measure function, which is selected according to the target problem. For example, if one wishes to detect drifts after noticing changes in mean values of data streams, the first statistical moment should be used (mean value). Otherwise, if another criterion is required, then the corresponding measure function must be set to quantify, for instance, other statistical moments, changes in amplitudes, frequencies, etc. The reader may also claim that there are other similar approach to ours, such as the one introduced in Sethi and Kantardzic (2017) which basically relies on verifying how data distributions change along time. It is relevant to mention such study because many others follow a similar strategy, however, besides its relevance, it does not model nor analyze the learning behavior as in here. In addition to that, as we employ the concept of Algorithmic Stability, our proposed framework holds for any measure function and not only on specific learning models and paradigms as also discussed in that later reference.

In this context, we highlight that the main contribution of this manuscript is a theoretical framework to ensure the results of unsupervised learning algorithms are not simply by chance when detecting concept drifts on data streams. In addition, we propose a straight-forward algorithm, named Plover, to detect model changes as new data observations are received, so whenever the input data behavior significantly changes (i.e. the generating process has been modified) instability is observed, allowing us to point out a concept drift has happened. The main intent is not to counterpose Plover to other algorithms from the literature, but to illustrate our theoretical framework and to share a fully-compatible algorithm so other researchers can analyze their own algorithms using our framework.

Besides that, as Plover illustrates our framework, we considered it to show experimental results to the reader. In order to provide a wide as possible analysis, statistical moments are used as measure functions by Plover, supporting the detection of drifts in terms of changes in means, variances, skewness and kurtosis. In addition, we decided to employ the Power Spectrum analysis (Yang, Feng, Pan, & Yang, 2009) as measure function in attempt to detect drifts in terms of data amplitudes and frequencies. The user may extend our result to consider other measure functions as well.

Experiments were performed using six different scenarios, four of them based on synthetic data streams and two on real-world data. Synthetic streams are relevant to confirm that our approach is capable of marking down drifts, as well as to test how different measure functions impact such detections. Using real-world data, we confirmed the detection of relevant drifts in Beethoven's Für Elise and also in the S&P 500 Index (stock market data).

This paper is organized as follows: Related work is discussed in Section 2; Section 3 provides the theoretical background required to formulate the Algorithmic Stability; The theoretical demonstration of our stable concept drift detection approach is detailed in Section 4; The Plover algorithm is introduced in Section 5; Section 6 presents the experimental results using synthetic and real-world data; Additional experiments to motivate the reader are provided in Section 7; A discussion and a comparison to yet other algorithms are showed in Section 8; Concluding remarks and future work are detailed in Section 10; Appendix A contains a detailed proof of McDiarmid's Inequality, which is central to our formulation; Finally, the references are listed.

2. Related work

There are several explicit (term here used to refer to supervised learning) concept drift algorithms found in the literature that consider a map $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{X}$ is the input space, $\mathcal{Y}$ corresponds to classes, and f is a classifier. Among those algorithms are: FLORA (Widmer & Kubat, 1996), Support Vector Machines (Klinkenberg & Joachims, 2000), OLIN (Last, 2002), CVFDT (Hulten, Spencer, & Domingos, 2001), UFFT (Gama, Medas, & Rocha, 2004), DDM (Gama, Medas, Castillo, & Rodrigues, 2004), EDDM (Baena-García et al., 2006), DDM-OCI (Wang et al., 2013b), ADWIN Bagging (Bifet, Holmes, Pfahringer, Kirkby, & Gavaldà, 2009), ASHT Bagging (Bifet et al., 2009), Random Forest based classifier (Abdulsalam, Skillicorn, & Martin, 2011), VHT (Prasad & Agarwal, 2017), Fuzzy Passive-aggressive classification (Wang, Ji, & Jin, 2013a), SimC (Mena-Torres & Aguilar-Ruiz, 2014), Online Stream classifier with incremental semi-supervised learning (Loo & Marsono, 2015), Distance-based Ensemble online classifier with kernel clustering (Jedrzejowicz & Jedrzejowicz, 2015), One-class classifier with incremental learning and forgetting (Krawczyk & Woźniak, 2015), and Classifying recurring concept using fuzzy similarity function (Angel, Bartolo, & Ernestina, 2016).

All explicit drift detection algorithms rely on classification/regression, therefore presenting issues/challenges associated to supervised learning algorithms, specially in the context of concept drift detection:

- They build up some mapping function $f : \mathcal{X} \rightarrow \mathcal{Y}$ and, therefore, require class labels;
- They face issues when a model is no longer valid for three main reasons:
  - Concept drifts cannot be predicted as they represent an unknown change over an underlying data probability distribution;
  - New labels should be considered along time, which would require further supervision. So, many application domains cannot take advantage of those algorithms, given that streams may be produced at high frequencies and specialists could never supervise them out at the same rate as they are collected;
  - A next model built up based on new data may produce an overfitted classifier, therefore generalization would never be ensured according to the Empirical Risk Minimization Prin-

[1] The learning model is here seen as the result of some unsupervised algorithm, such as a clustering partition.

ciple (ERMP) from the Statistical Learning Theory (Vapnik, 1998; 1995);
- Those algorithms rely on the Statistical Learning Theory (Vapnik, 1998; 1995), so they assume input data to be always independent and identically distributed (i.i.d.). This is not the case in most real-world applications, e.g., the temperature of given world region, the stock market, human heartbeats, etc.;
- Since i.i.d. must be ensured, non-stationary data must be dealt with in a different manner (say, embedded into phase spaces, as performed with dynamical systems (de Carvalho Pagliosa & de Mello, 2017)), which is not provided by those solutions.

On the other hand, implicit (term here used to refer to unsupervised learning) drift detection counts on methods such as OLINDDA (Spinosa et al., 2007), MINAS (Faria et al., 2013), DETECTNOD (Hayat & Hashemi, 2010), ECSMiner (Masud et al., 2010) and GC3 (Sethi et al., 2016), which, by employing clustering algorithms, assume that a drift is detected whenever new groups are found on new data. Those algorithms rely only on the data distribution estimation, disregarding any change in the behavior of learning models. In this sense, the algorithm may detect drifts even if the learning model is still adequate to represent data. MD3 (Sethi & Kantardzic, 2017) tackles this problem by verifying the density of points in uncertainty regions produced by a margin-bounded supervised learning model, focusing on its performance and not explicitly on data distributions. In this sense, whenever this density overpasses a threshold, the algorithm indicates a possible concept drift. In addition, it is worth to mention that (Harel, Crammer, El-Yaniv, & Mannor, 2014) compares empirical risks computed on ordered data S_{ord}, extracted directly from the stream, and on random permutations of S_{ord} in order to detect drifts. Sethi and Kantardzic (2017) and Harel et al. (2014) focus on the analysis of supervised predictions to perform drift detection, disregarding unsupervised learning models and their structural descriptions of input spaces.

In this context, we claim that supervised learning makes little sense in the context of concept drift detection for data streams, specially when data is produced at high frequencies. However, there is still a missing link in between the formalization provided by the Statistical Learning Theory (Vapnik, 1998; 1995) and Concentration Inequalities (Devroye et al., 1996). In that sense, we see the Symmetrization Lemma (de Mello & Ponti, 2018), considered by SLT, brings important connections with Concentration Inequalities, mainly with respect to McDiarmid's Inequality as it considers perturbed data acquired through the same probability distribution of the original data. It is worthwhile mentioning that this inequality has made possible the development of Algorithmic Stability, which is the main concept we considered to design our approach and provide learning guarantees for concept drift detection on data streams.

3. Concentration inequalities

Supervised Machine Learning relies on the Empirical Risk Minimization Principle (ERMP) to ensure learning generalization. ERMP is the central point of the Statistical Learning Theory (SLT) (Vapnik, 1998; 1995) which provides the theoretical foundation to analyze algorithms based on their biases and sample sizes. In this context, SLT can only be considered when data examples are labeled, consequently it is not suitable to model unsupervised learning. This lack of foundation for unsupervised learning finds support on the concept of Algorithmic Stability (Bousquet & Elisseeff, 2002), which presents conditions for the probabilistic convergence between some arbitrary function and its expected value, given that its domain is formed by independent random variables. By stipulating some divergence measure, one can verify stability for unsupervised learning algorithms. This concept of Algorithmic Stability relies on Concentration Inequalities to prove asymptotic bounds as briefly introduced below.

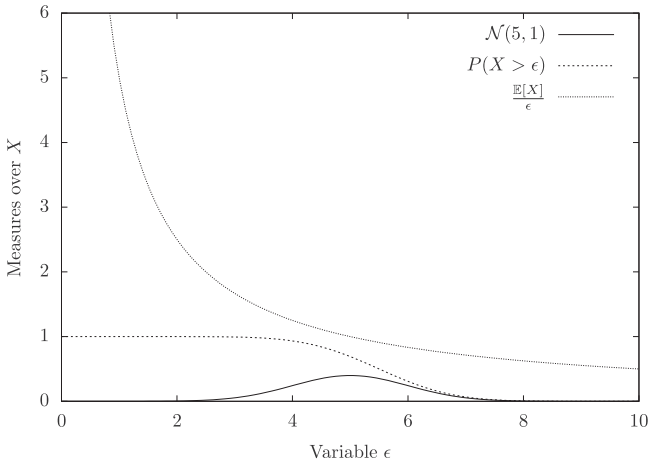

Fig. 1. Illustrating the upper bound ensured by the Markov Inequality, given $P(X > \epsilon) \leq \mathbb{E}[X]/\epsilon$.

In order to introduce Concentration Inequalities, we decided to start with the Markov Inequality, given it is used as a source to proceed with the next formulations. This inequality defines a bound for the complementary cumulative distribution $P(X > \epsilon)$, provided a non-negative random variable X sampled from some probability distribution function (PDF) $P(X)$ whose expected value $\mathbb{E}[X]$ is bounded as follows:

$$P(X > \epsilon) = \int_{\epsilon}^{\infty} dp(x).$$

Given $X \geq 0$, we have:

$$\int_{\epsilon}^{\infty} dp(x) \leq \int_{\epsilon}^{\infty} \frac{x}{\epsilon} dp(x) \leq \int_{0}^{\infty} \frac{x}{\epsilon} dp(x),$$

provided $\frac{X}{\epsilon} \geq 1$, for any $X \in [\epsilon, \infty]$, and $\int_0^{\epsilon} \frac{x}{\epsilon} dp(x) \geq 0$. From that, the expected value of X is $\mathbb{E}[X] = \int_0^{\infty} x dp(x)$, as consequence we obtain:

$$P(X > \epsilon) \leq \frac{\mathbb{E}[X]}{\epsilon} \tag{1}$$

which is known as the Markov Inequality.

As illustrated in Fig. 1, this inequality has a fractional behavior with respect to ϵ. It is worthwhile mentioning that for $\epsilon < \mathbb{E}[X]$, the Markov Inequality produces a result greater than one, consequently, as $P(X > \epsilon) \leq 1$, $\epsilon < \mathbb{E}[X]$ implies a tautology, which does not provide meaningful information about the distribution. Thus, ϵ must assume values greater than $\mathbb{E}[X]$ in order to produce relevant conclusions.

In attempt to extend the Markov Inequality, Chernoff (Devroye et al., 1996; von Luxburg & Schölkopf, 2011) employed the moment generating function to guarantee tighten convergences for $P(X > \epsilon)$. Provided $P(X > \epsilon) = P(sX > s\epsilon = P(e^{sX} > e^{s\epsilon})$ for $s > 0$, the Markov Inequality was used to define:

$$P(X > \epsilon) \leq \inf_{s>0} e^{-s\epsilon} \mathbb{E}\left[e^{sX}\right] \leq e^{-s\epsilon} \mathbb{E}\left[e^{sX}\right], \tag{2}$$

in which $\mathbb{E}\left[e^{sX}\right]$ is referred to as a moment generating function. Fig. 2 helps the reader to understand the motivation for using the moment generating function, given it allows to find the tightest bound for $e^{-s\epsilon}\mathbb{E}\left[e^{sX}\right]$ from its infimum.

Fig. 3 illustrates $f(X) = e^{sX}$ which, according to Jensen's Inequality, is convex for any $a < \infty$ and $b < \infty$ in the domain e^{sX}.

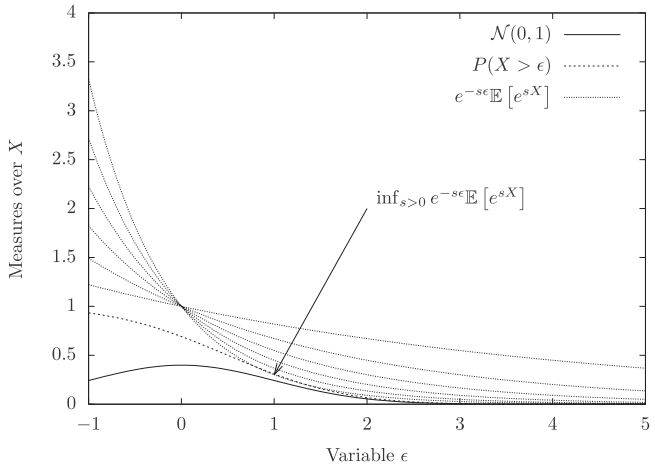

Fig. 2. A series of results produced by the moment generating function used in the Chernoff inequality. The infimum function is also illustrated to confirm its usefulness to formulate a tight convergence for $P(X>\epsilon)$, finally motivating the usage of such function as upper bound.

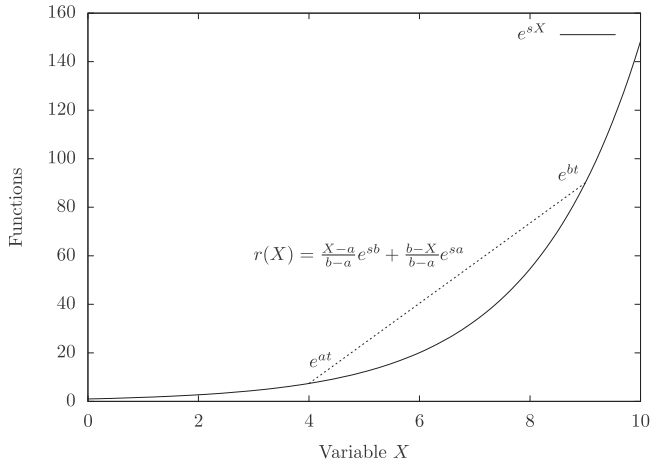

Fig. 3. Affine function $r(X)$ defined using the Jensen's inequality over an exponential function. This helps to illustrate the convexity occurring in some interval of variable X (in this case it is [4,9]), which is used as a necessary step to formulate the Chernoff bound.

Due to the convexity, a line segment or affine function $r(X)$ can be traced above $f(X)$, from e^{sa} to e^{sb}, as follows:

$$r(X) = \frac{X-a}{b-a}e^{sb} + \frac{b-X}{b-a}e^{sa}$$

so the domain of $e^{sX} \leq r(X)$ lies in the interval $X=[a,b]$. From that, we assume the random variable X is bounded in $[a, b]$ as well as $\mathbb{E}[X]=0$, therefore the expected value of e^{sX} is:

$$\mathbb{E}\left[e^{sX}\right] \leq \mathbb{E}[r(X)] = \frac{-a}{b-a}e^{sb} + \frac{b}{b-a}e^{sa}. \tag{3}$$

Let $p=\frac{b}{b-a}$ and $u=(b-a)s$, so that we have $\mathbb{E}[r(X)] = e^{sa}(p+(1-p)e^{u})$ and given some function $\phi(u)=\ln(\mathbb{E}[r(X)])$:

$$\begin{aligned}\phi(u) &= \ln(e^{sa}(p+(1-p)e^{u})) \\ &= sa + \ln(p+(1-p)e^{u}) \\ &= (p-1)u + \ln(p+(1-p)e^{u}).\end{aligned}$$

Now, it is a good strategy to apply the Second Order Taylor's Expansion over ϕ, so that $\phi(0)=0$, $\phi'(0)=0$, allowing us to obtain:

$$\begin{aligned}\phi''(\xi) &= \frac{(1-p)e^{\xi}}{p+(1-p)e^{\xi}} - \frac{((1-p)e^{\xi})^2}{(p+(1-p)e^{\xi})^2} \\ &= \frac{w}{p+w} - \frac{w^2}{(p+w)^2} \\ &= \frac{w}{p+w}\left(1-\frac{w}{p+w}\right).\end{aligned} \tag{4}$$

Thus, by maximizing $\phi''(\xi)$, we have $\phi(\xi) \leq \max_u \frac{1}{2}\phi''(\xi)u^2$. As $\frac{dw}{d\xi} = (1-p)e^{\xi} = w$:

$$\frac{d\phi''(\xi)}{dw} = \frac{2w}{w^2} - 2\frac{2w}{w^2}.$$

Considering $\frac{d\phi''(\xi)}{dw}=0$, we have that $w=\frac{1}{2}$ and, therefore, by using Eq. (4), $\max_{\xi}\phi''(\xi)=\frac{1}{4}$, so $\phi(u) \leq \frac{u^2}{8} \leq \frac{s^2(b-a)^2}{8}$. In this context, it follows from Eq. (3) that $\ln(\mathbb{E}\left[e^{sX}\right]) \leq \phi(u) \leq \frac{s^2(b-a)^2}{8}$ which implies:

$$\mathbb{E}\left[e^{sX}\right] \leq e^{\frac{s^2(b-a)^2}{8}}. \tag{5}$$

Calculating the infimum of $e^{s\epsilon}e^{\frac{s^2(b-a)^2}{8}}$ with respect to s, we find $s=\frac{4\epsilon}{(b-a)^2}$ and, therefore:

$$P(X>\epsilon) \leq e^{\frac{-2\epsilon^2}{(b-a)^2}},$$

which gives the Chernoff bound.

This Chernoff bound was extended by Hoeffding (Devroye et al., 1996) in order to calculate the infimum of $h(s)=e^{-s\epsilon}\mathbb{E}\left[e^{sA_n}\right]$. In such context, Hoeffding considered that X corresponds to a sum of independent random variables $A_n=\sum_{i=1}^{n}X_i$ bounded in interval $[a_i, b_i]$.

In this scenario, provided $(X_1, X_2, \ldots, X_n)$ form a sequence of independent random variables, the following holds $h(s)=e^{-s\epsilon}\prod_{i=1}^{n}\mathbb{E}\left[e^{sX_i}\right]$ and, thus, Inequality (5) implies $h(s) \leq e^{-s\epsilon}e^{s^2\sum\frac{(b_i-a_i)^2}{8}} = \alpha(s)$. Finally, calculating $\inf_{s>0}\alpha(s)$, we have the Hoeffding Inequality:

$$P(A_n>\epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}(b_i-a_i)^2}}. \tag{6}$$

This last inequality (Inequation (6)) requires a sequence of independent random variables, consequently if X_n depends on past observations, then one cannot employ it. [2] The independence requirement led Azuma (Devroye et al., 1996) to adopt Martingales in order to tackle such a dependency problem. Martingales are stochastic processes in which random variables rely on previous data and have a limited and stationary expected value, as shown in Fig. 4.

Formally, a Martingale process is defined as follows:

Definition 3.1 (Martingale)**.** Let $(X_1, X_2, \ldots, X_n)$ be a sequence of random variables. A martingale is a stochastic process satisfying:

$$\mathbb{E}[X] < \infty, \mathbb{E}[X_n|X_1, X_2, \ldots, X_{n-1}] = X_{n-1}. \tag{7}$$

For instance, take a random walk process to represent a scenario in which a point is moved to the left or right over the integer line. Displacements are represented by independent random variables $X_n \in \{-1, 1\}$ with expected value $\mathbb{E}[X]=0$. Although variables are independent, the point position still depends on the previous movements.

Let such random walk process be represented variable Y, whose nth value corresponds to the point position at step n, so that:

$$Y_n = Y_{n-1} + X_{n-1},$$

consequently,

[2] Here we have a very important requirement to proceed with Concentration Inequalities. For a detailed study on such matter, we suggest Devroye et al. (1996).

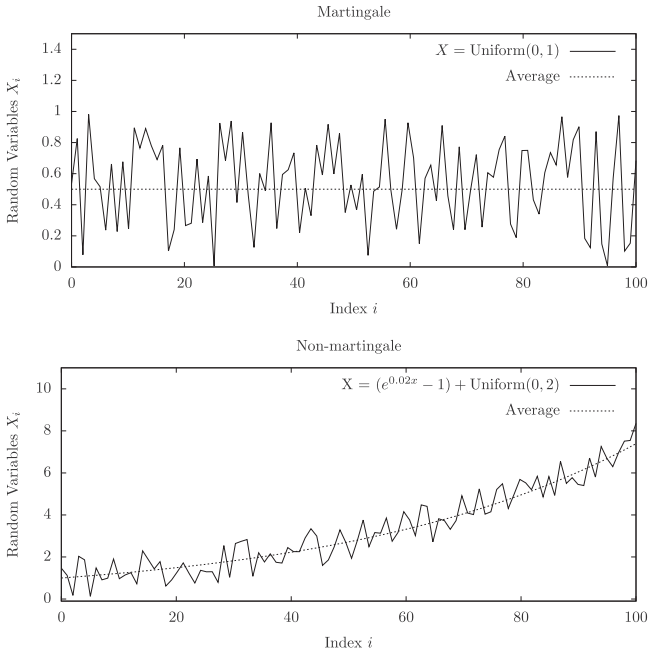

Fig. 4. At the top, an illustration of a Martingale process using independent random variables sampled from a uniform distribution. At the bottom, an example with random variables whose average value exponentially increases. The latter is out of the scope of Martingale processes.

$$\begin{aligned}\mathbb{E}[Y_n|X_1, X_2, \ldots, X_{n-1}] &= \mathbb{E}[Y_{n-1} + X_{n-1}|X_1, X_2, \ldots, X_{n-1}] \\ &= \mathbb{E}[Y_{n-1}|X_1, X_2, \ldots, X_{n-1}] \\ &+\mathbb{E}[X_{n-1}|X_1, X_2, \ldots, X_{n-1}] \\ &= Y_{n-1} + 0\end{aligned}$$

proving that such random walk relies on a martingale process.

Let $(X_1, X_2, \ldots, X_n)$ be a sequence of random variables produced by a martingale process. From that we can extend the previous formulation to show a martingale of differences $V_n = X_n - X_{n-1}$, in which $\mathbb{E}[V_n|X_1, X_2, \ldots, X_{n-1}] = 0$, is also a martingale process, so that one may take advantage of such step to ensure the independence requirement as proceed with his/her objectives:

$$\begin{aligned}\mathbb{E}[V_n|X_1, \ldots, X_{n-1}] &= \mathbb{E}[X_n - X_{n-1}|X_1, \ldots, X_{n-1}] \\ &= \mathbb{E}[X_n|X_1, \ldots, X_{n-1}] - \mathbb{E}[X_{n-1}|X_1, \ldots, X_{n-1}] \\ &= X_{n-1} - X_{n-1} = 0,\end{aligned}$$

consequently, a martingale of differences is also a martingale process. From such step, Azuma (1967) proved that, given V_i is bounded by a constant c_i, the sum $S_n = \sum_{i=1}^{n} V_i$ of a martingale of differences ensures the Hoeffding Inequality. Thus, consider the following lemma:

Lemma 3.1. *Let V and Z be random variables whose expected value $\mathbb{E}[V|Z] = 0$ with probability one. Assume that, for any function f and constant $c \geq 0$,*

$$f(Z) \leq V \leq f(Z) + c. \tag{8}$$

Then

$$\mathbb{E}\left[e^{sV}|Z\right] \leq e^{s^2c^2/8} \tag{9}$$

for every $s \geq 0$.

From Jensen's Inequality (Inequality 3), $\mathbb{E}\left[e^{sV}|Z\right]$ is defined as follows:

$$\mathbb{E}\left[e^{sV}|Z\right] \leq \mathbb{E}\left[\tfrac{V-a}{b-a}e^{bt}|Z\right] + \mathbb{E}\left[\tfrac{b-V}{b-a}e^{at}|Z\right] \tag{10}$$

$$= \tfrac{\mathbb{E}[V|Z]-a}{b-a}e^{bt} + \tfrac{b-\mathbb{E}[V|Z]}{b-a}e^{at}. \tag{11}$$

Note that $\mathbb{E}[V|Z] = 0$ and the problem is therefore analogous to the Hoeffding Inequality, thus proving Lemma 3.1. Extending this result for sums of a martingale of differences and having $Z_i < V_i < Z_i + c_i$, using the Chernoff bound one finds:

$$P(S_n > \epsilon) \leq \inf_{s>0} e^{-s\epsilon}\mathbb{E}\left[e^{sS_n}\right] = \inf_{s>0} e^{-s\epsilon}\mathbb{E}\left[e^{s(S_{n-1}+V_i)}\right] \tag{12}$$

$$= \inf_{s>0} e^{-s\epsilon}\mathbb{E}\left[e^{sS_{n-1}}\mathbb{E}\left[e^{sV_n}|X_1, X_2, \ldots, X_{n-1}\right]\right] \tag{13}$$

$$\leq \inf_{s>0} e^{-s\epsilon}\mathbb{E}\left[e^{sS_{n-1}}\right]e^{\frac{s^2c_n^2}{8}} \text{ (By Lemma 3.1)} \tag{14}$$

$$\vdots \tag{15}$$

$$\leq \inf_{s>0} e^{-s\epsilon}e^{\frac{s^2\sum_{i=1}^{n-1}c_i^2}{8}}, \tag{16}$$

which is then considered in conjunction with the Hoeffding Inequality to finally obtain:

$$P(U_n > \epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n-1}c_i^2}}. \tag{17}$$

As a next formulation step, McDiarmid (1989) took advantage of such theoretical foundation to study how perturbations on random variables would propagate to any arbitrary function. McDiarmid's Inequality ensures that, given independent random variables $(X_1, X_2, \ldots, X_n)$, if perturbations are limited over a function g, then the divergence between $g(X_1, X_2, \ldots, X_n)$ and its expected value follows Hoeffding's Inequality. Formally:

$$\text{If} \quad \sup_{x_1, x_2, \ldots, x_n, x_i'} \left|g(x_1, x_2, \ldots, x_i, \ldots, x_n) - g(x_1, x_2, \ldots, x_i', \ldots, x_n)\right| \leq c_i \tag{18}$$

$$\text{then} \quad P(g(X_1, X_2, \ldots, X_n) - \mathbb{E}[g(X_1, X_2, \ldots, X_n)] > \epsilon) \leq e^{\frac{-2\epsilon^2}{\sum_{i=1}^{n}c_i^2}}, \tag{19}$$

where $c_i = b_i - a_i$ and $X_i \in [a_i, b_i]$.

All the formulation detailed in this section provides the theoretical background we considered to employ the concept of stability in the context of concept drift detection, as discussed next.

4. Employing stability to detect concept drifts

In this section, the uniform stability, which provides generalization (Bousquet & Elisseeff, 2002), is defined. Let S be a training set containing m instances in form $\{z_1, z_2, \ldots, z_m\} \in Z^m$, which may or may not be associated to class labels. Consider they are sampled from some unknown probability distribution P, and assume perturbations given by permutations as follows:

$$S^i = z_i, \ldots, z_{i-1}, z_i', z_{i+1}, \ldots, z_m, \tag{20}$$

where z_i' is independent of S but sampled from the same probability distribution $P(Z)$. Uniform Stability is defined as follows:

Definition 4.1 (Uniform Stability)**.** An algorithm A is β-uniformly stable with respect to some loss function $\ell(\cdot)$ iff

$$\forall S \in Z^m, \ \forall i \in 1, \ldots, m, \ \|\ell(A_S, \cdot) - \ell(A_{S^i}, \cdot)\|_\infty \leq \beta, \tag{21}$$

where A_S is a model produced by the application of an algorithm A over the training set S. Notice that if Inequality (21) holds, then the condition of McDiarmid's Inequality (Inequation (A.7)) is respected and, therefore:

$$P(\ell(A_S, \cdot) - \mathbb{E}[\ell(A_S, \cdot)] > \epsilon) \leq e^{\frac{-2\epsilon^2}{m\beta^2}}. \tag{22}$$

Let us apply Uniform Stability to find a framework to detect changes in data streams. First, we take McDiarmid's Inequality from Inequality (22) and assume that:

$$\delta = e^{\frac{-2\epsilon^2}{m\beta^2}},$$

consequently,

$$P(\ell(A_S,\cdot) - \mathbb{E}[\ell(A_S,\cdot)] > \epsilon) \leq \delta. \tag{23}$$

Solving for ϵ, we find:

$$\begin{aligned} \delta &= e^{\frac{-2\epsilon^2}{m\beta^2}} \\ \log\delta &= \frac{-2\epsilon^2}{m\beta^2} \\ \epsilon &= \pm\sqrt{-\tfrac{1}{2}\log(\delta)m\beta^2}. \end{aligned}$$

It follows from Inequality (19) that:

$$\epsilon = \pm\sqrt{-\frac{1}{2}\left(\log\delta\sum_{i=1}^{m}c_i^2\right)},$$

since $\beta = \max(c_1,\ldots,c_n)$, $c_i = b_i - a_i$, and $|\ell(A_S,\cdot) - \ell(A_{S^i},\cdot)| \in [a_i, b_i]$.

Similarly to the way Vapnik (1998, 1995) formulated a step of the Statistical Learning Theory, we have:

$$\begin{aligned} \ell(A_S,\cdot) - \mathbb{E}[\ell(A_S,\cdot)] &> \epsilon, \\ \ell(A_S,\cdot) - \mathbb{E}[\ell(A_S,\cdot)] &> \pm\sqrt{-\tfrac{1}{2}\left(\textstyle\sum_{i=1}^{n}c_i^2\log\delta\right)}. \end{aligned}$$

Taking the absolute value (as only the divergence is relevant, not its sign), we obtain at:

$$|\ell(A_S,\cdot) - \mathbb{E}[\ell(A_S,\cdot)]| > \sqrt{-\frac{1}{2}\left(\log\delta\sum_{i=1}^{n}c_i^2\right)}.$$

The above inequality means that divergence happens. However, we are interested in the complementary case, i.e., all successful cases:

$$|\ell(A_S,\cdot) - \mathbb{E}[\ell(A_S,\cdot)]| \leq \sqrt{-\frac{1}{2}\left(\log\delta\sum_{i=1}^{n}c_i^2\right)},$$

from which we find:

$$\mathbb{E}[\ell(A_S,\cdot)] \leq \ell(A_S,\cdot) + \sqrt{-\frac{1}{2}\left(\log\delta\sum_{i=1}^{n}c_i^2\right)}, \tag{24}$$

which is the criterion we propose in this manuscript to assess the Uniform Stability of unsupervised learning algorithms in the context of Data Streams. Such stability can be used to ensure a model is enough to represent the stream and, therefore, point out when relevant changes occur, either to update modeling or to detect concept drifts as discussed in the next section.

Next section also introduces the Plover unsupervised learning algorithm, designed to illustrate how the Uniform Stability criterion, defined in Inequation (24), can be held. Here it is relevant to mention that some valid function $\ell(\cdot)$ must be also defined to proceed with the concept drift detection. As discussed by Vapnik (1998, 1995), this function must be independent of the data we have access to; what could exactly cause inconsistency to the Empirical Risk Minimization Principle in the context of supervised learning, since the learning algorithm could converge to a memory-based classifier (von Luxburg & Schölkopf, 2011). In our scenario, we consider data streams are unlabeled, consequently our learning task is associated to unsupervised learning and, therefore, function $\ell(\cdot)$ can be any one capable of somehow capturing the nature of the data under consideration (measuring the similarities between the actual and past data). However, there is an additional requirement to proceed with our formulation which is based on the Law of Large Numbers (von Luxburg & Schölkopf, 2011): input data has to be independent and identically distributed (i.i.d.).

This i.d.d. requirement brings several challenges, specially on how to ensure such property in the context of data streams given observations typically depend on each other along time. Fortunately, all those questions were already answered in de Carvalho Pagliosa and de Mello (2017) by reconstructing data-stream observations into phase spaces (Kantz & Schreiber, 2003), so that we can proceed with our theoretical framework.

5. The plover: a uniformly-stable concept drift detection algorithm

As discussed in the previous section, there are two necessary assumptions to carry on while employing the theoretical framework proposed in this article: i) the function $\ell(\cdot)$ has to be selected independently of the input data; and ii) input data has to be independent and identically distributed (as already addressed in de Carvalho Pagliosa & de Mello, 2017). From that, we suggest the use of any invariant measure one intends to assess. For example, $\ell(\cdot)$ could be the average or the standard deviation function.

The Plover concept drift detection algorithm is introduced in Algorithm 1. It employs the above-proposed criterion:

Algorithm 1 The Plover algorithm.

Require: data stream D; function $\ell(.)$; window length w_l; probability δ; threshold T;
  Initialize queue Q as empty
  while there is data **do**
    Collect a next data observation z_i
    Store z_i in a queue Q\{If the queue contains enough data, then process it.\}
    if length(Q) == w_l **then**
      $l_i = \ell(Plover, Q)$ {Computing the measure function on the current data window.}
      $c_i = \frac{|\max l_{1:i} - \min l_{1:i}|}{i}$ {Computing the divergence between the current measure and the previous one to find c_i.}
      $div = \sqrt{-\frac{1}{2}\left(\log\delta\sum_{j=1}^{i}c_j^2\right)}$ {Computing the divergence term.}
      if $div > T$ **then**
        Warn other systems or individuals about the drift detection
      end if
    end if
  end while

$$\mathbb{E}[\ell(A_S,\cdot)] \leq \ell(A_S,\cdot) + \sqrt{-\frac{1}{2}\left(\log\delta\sum_{i=1}^{n}c_i^2\right)},$$

where A_S is some pattern recognition algorithm applied over some dataset S; $\mathbb{E}[\ell(A_S,\cdot)]$ is the expected value of some measure $\ell(\cdot)$ applied over A_S; c_i is a constant which bounds the divergence between $\ell(A_S,\ \cdot)$ and $\ell(A_{S^i},\cdot)$, as represented on McDiarmid's Inequality (Inequality (A.7)); n is the number of measurements collected along the input data; δ is the probability we wish to ensure that $\ell(A_S,\ \cdot)$ is a good estimator for $\mathbb{E}[\ell(A_S,\cdot)]$, as defined in Inequality (23); $\ell(A_S,\ \cdot)$ is the empirical measure computed on a given input data sample; and, finally, the last term computes possible variations in random variables.

In order to apply the Plover, meaningful parameters must be provided. In fact, they are not difficult to set. For example, the probability δ may be equal to 0.05, meaning we expect $\ell(A_S, \cdot)$ to be a good estimator for $\mathbb{E}[\ell(A_S, \cdot)]$ in 95% of cases, as typically addressed in statistical hypothesis tests. [3] Threshold T may be a static value or some index relative to the past divergences (*div* in Algorithm 1). Now we stay with the two most important parameters: the function $\ell(\cdot)$ and the window length w_l. In this paper we employ the Statistical moments as function $\ell(\cdot)$, allowing to evaluate the following aspects:

1. Mean: it allows to understand if the average data behavior is maintained along the collection;
2. Variance: permits to assess the expectation of the squared deviation from the mean. It measures how far a set of random variables is spread out from its average value. It also helps to understand the historical data divergences;
3. Skewness: this is a measure of asymmetry of the data probability distribution about the mean. It can be positive or negative, or undefined. This allows to point out data tendencies along collection;
4. Kurtosis: this allows to describe the shape of the data probability distribution. In this situation we are interested in pointing out infrequent extreme deviations (or outliers) when obtaining greater values, as opposed to frequent modestly sized deviations.

Other data stream aspects could be assessed using the same algorithm proposed in this paper by simply changing the function $\ell(\cdot)$. There is still one parameter missing: the window length. Researchers typically face some issues while setting such parameter, but we look at it in a different way. For example, if the data stream collected is produced by the heart beats of some patient and measured by an electrocardiogram (ECG) whose sampling frequency is 125 Hz, we could take advantage of the Nyquist–Shannon sampling theorem (Shannon, 1949) to set w_l. According to such theorem, if we have the double number of observations, then we can analyze the signal properly and take fair conclusions. In that situation, windows should contain at least twice the number of observations, i.e., 250 elements.

However, there are situations in which we do not know which is the underlying sampling frequency of the data collection. In those cases, we should employ a Power Spectrum (PS) to assess the Fourier complex coefficients and observe magnitude differences, something quite common in the context of signal processing analysis (Yang et al., 2009). If PS produces similar magnitudes along frequencies normalized in the range [0,1], then we can reduce the number of observations being assessed, which is the same as reducing the sampling frequency and, therefore, the window length. An adequate window length is found when the magnitudes computed for some data window having p elements is approximately the same as by using $2p$ elements. In such situation, we could set $w_l = p$ and proceed with the algorithm running.

Based on such concept, we designed Algorithm 2 to assess the window length based on the Power Spectrum (PS) analysis. PS computes the Fourier Transform on a data window to produce a matrix associating frequencies and amplitudes to the data. In this algorithm, we compute PS for two different data windows, the first with length w and a next with $w+1$ observations, so we can compute how divergent the amplitudes and frequencies are as we increase the window length.

We employ the Dynamic Time Warping (DTW) (Berndt & Clifford, 1994), which provides a best matching between time series, to compare the PS acquired by two different data windows. In this case those series are resultant of magnitudes along frequencies, i.e., the absolute value of Fourier complex coefficients over the frequency axis. After running this algorithm, we can plot matrix Q to obtain a chart with the window length versus the divergences. As long as the divergences significantly reduce and have only some small perturbation, we can estimate an adequate window length w_l to run the Plover algorithm next. During the experiments we provide some examples of output for this window estimation algorithm.

Algorithm 2 Estimating the Plover window length .

Require: data stream D; starting window length w_s; final window length w_f;
  return the association of divergences produced along frequencies and amplitudes as the window length is increased;
  Initialize matrix Q to contain outputs
  for w from w_s to w_f **do**
    Initialize vector V with zeros
    for i from 1 to $|D| - w$ **do**
      $c_1 = PS(D[i : (i + w)])$ {Computes the power spectrum}
      $c_2 = PS(D[i : (i + w + 1)])$ {Observe this adds up one extra observation to provide the power spectrum comparison}
      $div = dtw(c_1, c_2)$ {Computes the Dynamic Time Warping measure between the magnitudes of both power spectrum coefficients to assess frequency and amplitude changes}
      $V[i] = div$
    end for
    $Q[w,] = [w, V]$
  end for

After setting all those parameters, we here explain the processing results of Plover to illustrate its usage. Using Algorithm 1, observe Plover receives data observations as they are produced by a given phenomenon, then it stores such data into a buffer or queue Q. When the queue has at least w_l elements, the processing starts by computing function $\ell(.)$ over Q, resulting in some real value l_i. For instance, consider the queue contains observations produced using $\sin(2 \times pix)$, given $x \in \mathbb{R}$, and that $\ell(.)$ computes simply the mean value of observations in Q, which tends to approach zero for a fair queue length. In this scenario, l_i would receive the average value for the current queue and, next, it would find the largest and the smallest values considering all past values of l_i for $i \in \mathbb{Z}_+$. Then, the minimal and the maximal values are used to compute c_i, i.e. the range of data variation since the beginning of our assessment, which is then applied into our formulation to verify if the current data mean value sufficiently diverges (*div*) from historical measurements. If divergence *div* is greater than some threshold T, then a drift is detected. If the next observations slightly change their mean value, c_i will contain some value greater than zero that will affect the Algorithmic Stability, so that $div > 0$. Eventually, $div > T$ and the drift is issued.

From this detailed explication, one notice the measure function $\ell(.)$ can be any other, such as the variance, skewness, kurtosis or more complex ones. For example, if one wishes to assess the same data stream but using the Fourier Transform, such a researcher could use the Fourier complex coefficients to obtain a matrix with frequencies and their corresponding magnitudes (or amplitudes). Consecutive magnitude-frequency matrices l_{i-1} and l_i (observe l_i would be the current matrix in here) should be compared to obtain some index c_i (in this scenario, it could compute the Dynamic Time Warping distance simply using the resulting amplitudes, once they are already organized along frequencies) which would be used along the next steps to ensure the concept drift detection.

[3] Any other probability bound may be employed instead.

Require: data stream D; function $\ell(.)$; window length w_l; probability δ; threshold T;
Initialize queue Q as empty
while there is data **do**
Collect a next data observation z_i
Store z_i in a queue Q\{If the queue contains enough data, then process it.\}
if length(Q) == w_l **then**
$l_i = \ell(Plover, Q)$ {Computing the measure function on the current data window.}
$c_i = \frac{|\max l_{1:i} - \min l_{1:i}|}{i}$ {Computing the divergence between the current measure and the previous one to find c_i.}
$div = \sqrt{-\frac{1}{2}\left(\log \delta \sum_{j=1}^{i} c_j^2\right)}$ {Computing the divergence term.}
if $div > T$ **then**
Warn other systems or individuals about the drift detection
end if
end if
end while

6. Results

6.1. Setup

Experiments were conducted on synthetic and real-world datasets. The synthetic datasets were designed to explore all the statistical moments and the power spectrum taken as measure functions. The first, second and third experiments receive data streams produced according to Normal probability distributions. A sinusoidal data stream is the input for the fourth experiment, in order to confirm that the power spectrum can analyze variations in frequencies along time.

Next, two other experiments take real-world data to assess drift change. The first considers Beethoven's Für Elise song as input and points out when frequency changes occur (due to a harmonic change). The last attempts to point out when the S&P 500 Index reflected the great American crisis in 2008. The next section detail all experiments and discuss results.

6.2. Synthetic scenarios

The first experiment takes the data stream illustrated in Fig. 5(a) as input and it was designed to confirm whether the mean measure function would be enough to detect a relevant drift at the time index equal to 10,000. This data stream was generated using the Normal probability distribution with mean equal to 0 and standard deviation of 0.75 for the first 10,000 observations; the next 10,000 were produced with the mean and deviation of 1 and 0.1, respectively.

Our interest here is to confirm that the Plover algorithm is enough to detect concept drifts, but the measure function has indeed to be set according to the application domain and goals. We employed Algorithm 2 to estimate the best sliding window length for this dataset. As shown in Fig. 5(b), $w_l = 400$ provides a good enough sampling rate for this stream.

Fig. 5(c) confirms the mean function is adequate, as expected, to indicate when the concept drift has happened. From this initial setting, we proceed with other scenarios and assess the Plover algorithm under different measure functions.

The second experiment receives a data stream produced also with the Normal distribution, having the first 10,000 observations with mean and standard deviation equal to 0 and 0.75, respectively. Then, the next 10,000 observations have the same mean value but the deviation is 0.1.

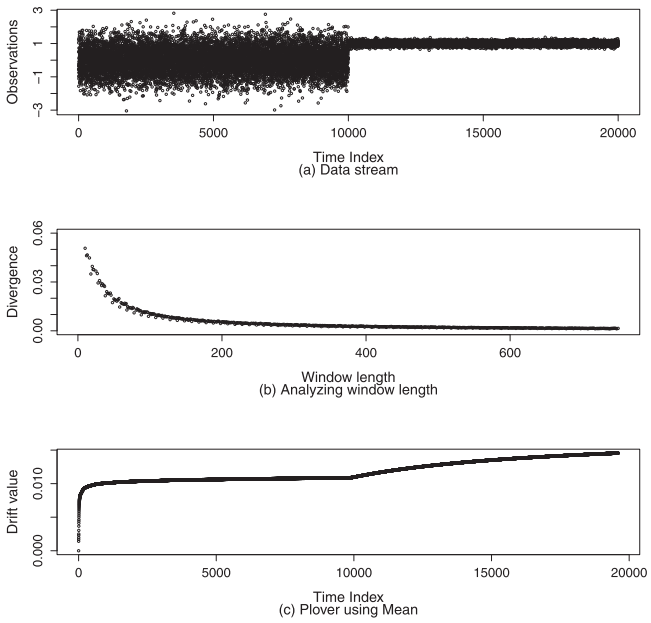

Fig. 5. Applying the Plover algorithm on the first synthetic dataset.

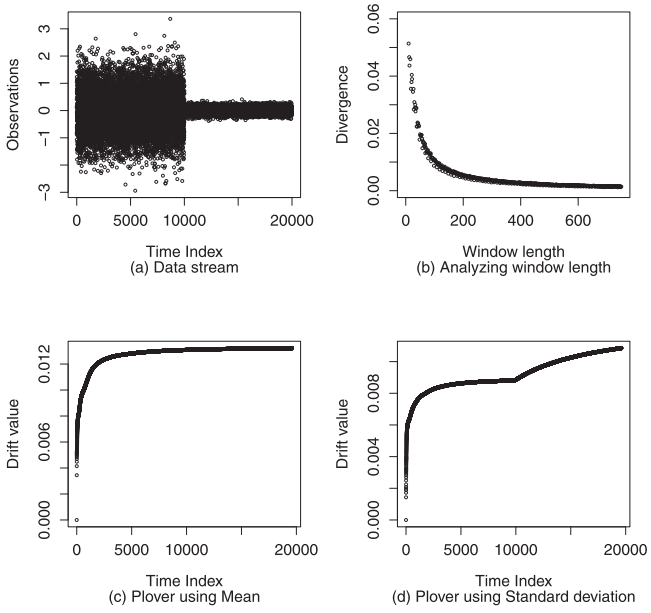

Fig. 6. Applying the Plover algorithm on the second synthetic dataset.

In this situation the overall mean value along time is around the same, but the standard deviation changes, thus another measure function should be capable of detecting the concept drift. Fig. 6 shows the data stream, the assessment of the window length (from that we set $w_l = 400$), and finally the Plover results using the mean (c) and the standard deviation measure functions (d). As observed, the standard deviation function is capable of pointing out the drift along data collection.

The third experiment considers the same dataset as the first, but assesses the Skewness and Kurtosis functions instead. Here we again set $w_l = 400$. Fig. 7(a and b) illustrate the drift detection using the Plover algorithm with the Skewness and Kurtosis, respectively. Observe both improve the results of the first experiment, given that the mean decentralization produces a greater impact in both measures. In this sense, the first samples acquired from the new data probability distribution could be considered as outliers,

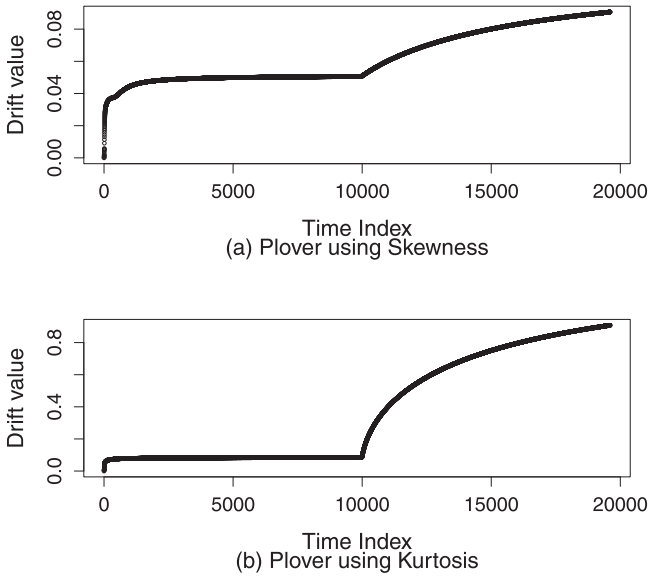

Fig. 7. Applying the Plover algorithm on the third synthetic dataset.

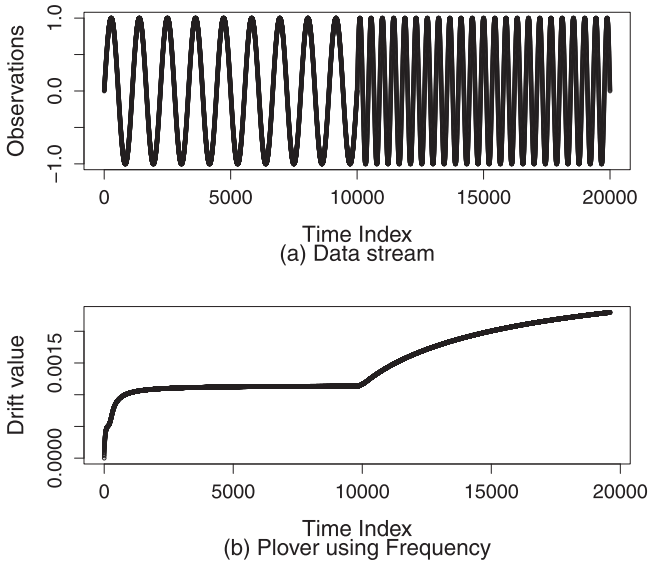

Fig. 8. Applying the Plover algorithm on the fourth synthetic dataset.

as they are not the usual data. Then, as new instances are presented, an asymmetric behavior is propagated on the overall distribution.

The fourth and last synthetic experiment takes a data stream composed of 20,000 observations. The first half of them follows function $\sin 2\pi x$, while the second $\sin 5\pi x$, having x as a linear variable. Therefore, the sine frequency changes at time 10,000, something that is well observed while using the Power Spectrum (PS) as a measure function for Plover (Fig. 8). Here we set $w_l = 400$.

We did not provide the analysis for this dataset using the other measure functions, simply because they do not bring any relevant evidence for this task. From that we conclude that every particular data stream should take advantage of one measure function to point out changes along collection. However, the theoretical foundation provided in this paper is still the same that motivated the design of the Plover algorithm.

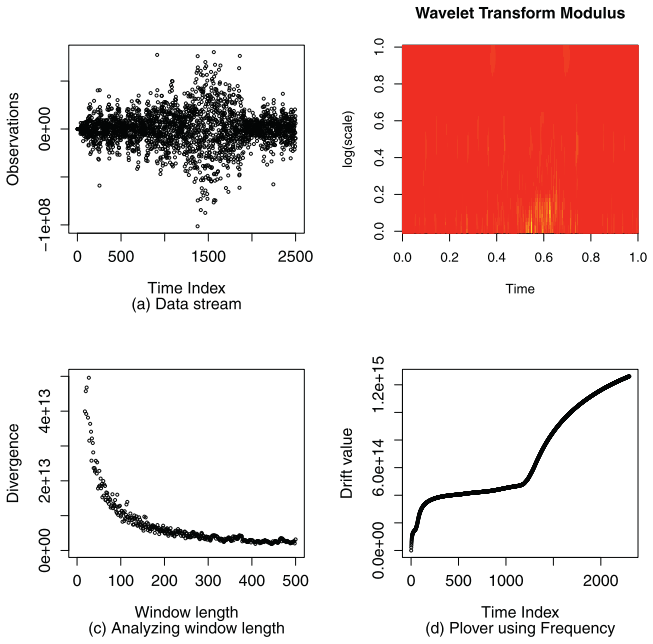

Fig. 9. Applying the Plover algorithm on Beethoven's Für Elise.

6.3. Real-world scenarios

This first experiment considered the first 29 s of the song Für Elise by Beethoven, which has a relevant frequency change along time. Figs. 9(a–d) show the data stream, a spectrogram to make evident the frequency change, followed by the window length analysis which allowed to set $w_l = 200$, and, finally, the drift values produced by the Plover algorithm with the Power Spectrum (PS) as a measure function.

As observed, the Plover was again effective to point out the change. However, it is worth to mention that the other measure functions did not provide any relevant result. This makes sense since this problem requires some function capable of summarizing frequencies and amplitudes along time.

The last experiment was performed using the Standard & Poor's 500, an American stock market index based on the market capitalizations of the 500 largest companies listed on NYSE and NASDAQ. We took the daily adjusted closing price from November 5th, 2003 to January 1st, 2018 as the input stream.

Figs. 10(a–d) start by illustrating the data stream, in which the index already reflects the American crisis from June 6th, 2007 and the economy changes its movement from February 4th, 2009. Next, the window length for the Plover algorithm is assessed and from that we set $w_l = 200$. As next charts we provide the drift results with the Plover algorithm while using the Standard deviation and the Power Spectrum as measure functions.

The other measure functions were omitted due to the fact that they do not provide any drift information. We again confirm that our theoretical approach provides the necessary foundation to detect concept drift in data streams. However, it is again very important to emphasize that the measure function to be used must reflect the target task requirements.

7. Additional experiments using unknown drifts

The main objective of this section is to bring some motivating results in terms of a broader range of experimentation on data streams where concept drifts are unknown in advance. In order to proceed, we introduce the selected streams: i) Historical

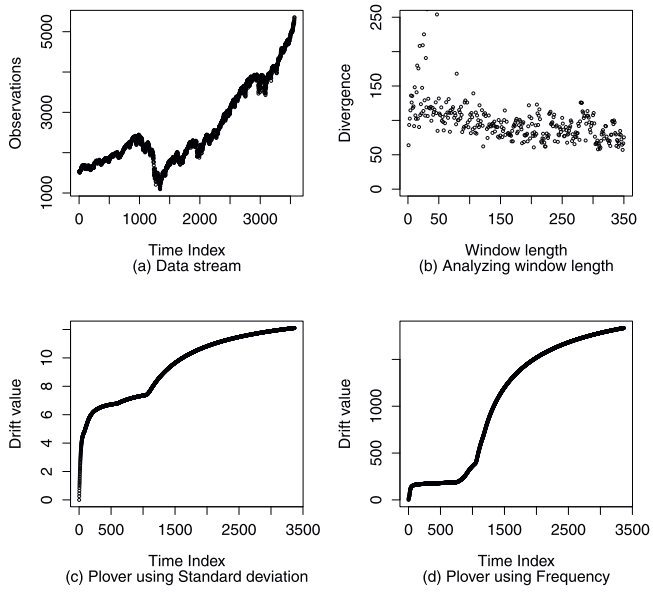

Fig. 10. Applying the Plover algorithm on S&P 500 Index.

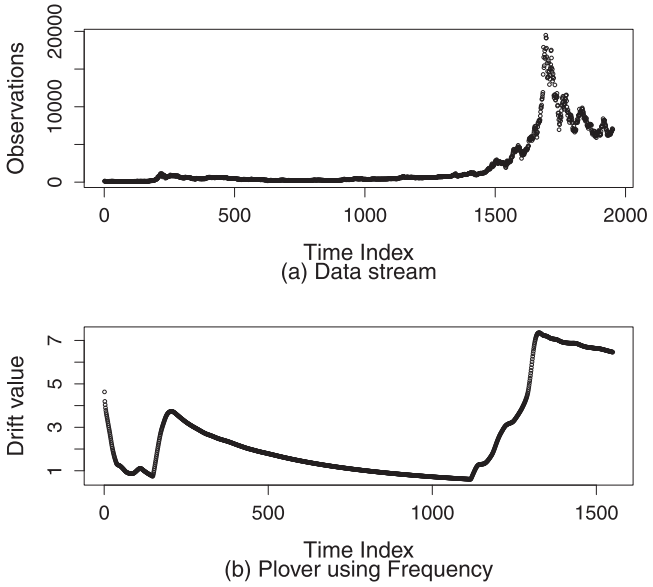

Fig. 11. Analyzing Bitcoin prices using the Plover algorithm.

prices of Bitcoin [4] – we decided to consider the close prices of bitcoin since its beginning in attempt to mark down the most relevant events along time. Besides no supervision is provided, we already had a fair idea that changes would occur, specially in recent prices; ii) Electricity [5] – this stream was collected from the Australian New South Wales Electricity Market, knowing prices are not static and claimed to be eventually affected by the overall demand and supply. Observations were collected at every five minutes (Bifet, Holmes, Kirkby, & Pfahringer, 2010); and iii) Data of flight leaving the Los Angeles International Airport (LAX) [6] – This stream was inspired in the regression dataset from Elena Ikonomovska. [7] The purpose of this data is to predict whether a given flight will be delayed, given the information of the scheduled departure (Bifet et al., 2010).

We decided to employ the Power Spectrum analysis (Yang et al., 2009) as measure function in attempt to detect drifts in terms of data amplitudes and frequencies, as performed in previous experiments. The main reason is that it supports the analysis of frequencies and amplitudes along time, providing a fair point of view about data streams with unknown drifts (motivated by Fourier and Wavelets Analyses).

Fig. 11 confirms two relevant changes, the first is associated with day 250 and the second with day 1,400. The second was indeed expected, but the first might look insignificant while analyzing the prices in the top-most chart. That first drift happened in December 2013, what is particularly interesting given many relevant news come from that period. [8] The second detection happens in January 2017, when again news discuss the Bitcoin effect and compare it against December 2013. [9] We claim our detections reflect relevant points of interest, which are confirmed by news and analysts. However, no proper supervision was provided in here.

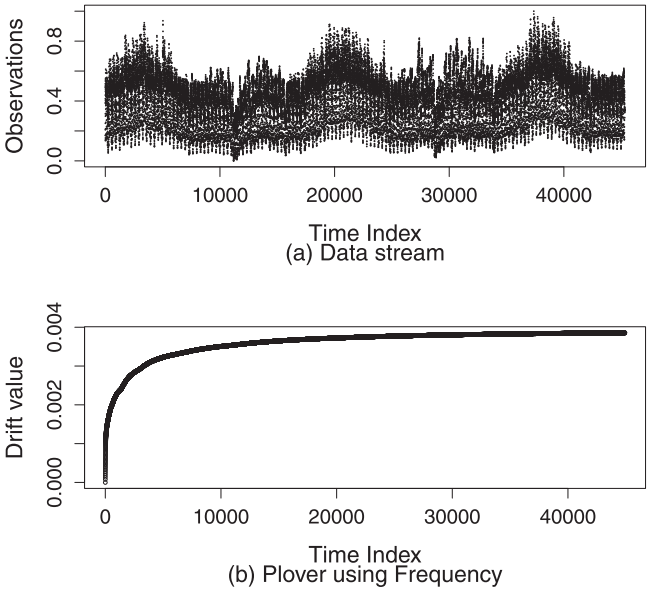

Fig. 12. Analyzing Electricity demands using the Plover algorithm.

Fig. 12 brings a different perspective. Besides changes in electricity prices, the overall demand did not significantly change according to the frequency and amplitude analysis. In fact, our drift curve (bottom-most chart) points out a relevant stability in the demand along time, what might also be visually observed through the sinusoidal behavior seen at the top-most chart.

Fig. 13 shows our analysis for flights leaving the LAX airport. A fraction of observation is shown at the top-most chart – once values are binary, indicating whether the delay has happened (1) or not (0), all data would clutter the chart. The same stability observed in the Electricity demand is seen in such results, confirming the LAX Airport has kept its relative delay occurrences stable along time. Depending on the point of view that may look good or bad. Good in the sense it did not worsen, but bad in the sense it did not improve either.

All those results motivate further studies and improvement in the area of concept drift detection and we, as authors, hope they

[4] Available at https://coinmarketcap.com/currencies/bitcoin/historical-data/?start=20130428&end=20180831.

[5] Available at https://sourceforge.net/projects/moa-datastream/files/Datasets/Classification/elecNormNew.arff.zip/download.

[6] Available at https://sourceforge.net/projects/moa-datastream/files/Datasets/Classification/airlines.arff.zip/download.

[7] More information at https://kt.ijs.si/elena_ikonomovska/data.html.

[8] See https://www.forbes.com/sites/kitconews/2013/12/10/2013-year-of-the-bitcoin#73c9f1c3303c.

[9] See https://www.coindesk.com/bitcoin-not-2013.

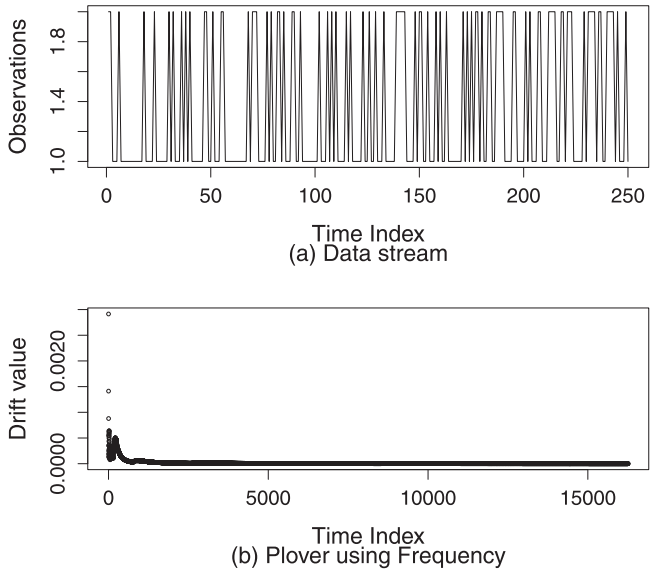

Fig. 13. Analyzing Airport delays using the Plover algorithm.

motivate the reader to understand application scenarios and the usefulness of such area of study.

8. Discusion

After presenting our approach, one may wonder how to compare it against other concept drift detection algorithms based on data changes, such as CUSUM (Page, 1954), PHT (Page, 1954), EWMA (Ross, Adams, Tasoulis, & Hand, 2012), SONDE (Albertini & de Mello, 2007), GWR (Marsland, Shapiro, & Nehmzow, 2002), and ADWIN (Bifet & Gavaldà, 2007).

The Cumulative Sum (CUSUM) and the Page Hinkley-Test (PHT) are memory-less data tendency estimators. The first models a cumulative residue to point out changes along time, while the second computes the divergence of those residues in a linear manner.

The Exponential Weighted Moving Average proposed in Ross et al. (2012) is similar to both CUSUM and PHT, in the sense that it models residues along time in a linear fashion. However, it also takes in consideration both average and standard deviation estimators in order to provide some confidence level to detect drift.

The Self-Organizing Novelty Detection (SONDE) and Grow When Required (GWR) neural networks were designed to incrementally build up data clusters to point out new events occurring on input space. SONDE extends GWR by including an Entropy measure to also detect changes along time. Both algorithms build up new centroids as new observations are received. They basically follow data tendencies along collection and take advantage of Entropy as a measurement for detecting behavior modifications. They both lack in terms of formal aspects to ensure learning and, in principle, they could also take advantage from our framework to be improved in that respect.

The Adaptive Sliding Window algorithm (ADWIN) is probably the closest to our formulation in the sense that it estimates an equation that is very close to McDiarmid's Inequality (Inequality (19)), but yet missing the other formulation steps, justifications, and the discussion about other measure functions as performed throughout this work. In fact, ADWIN was one of our main motivations for this study after several discussions about how to ensure theoretical learning guarantees for unsupervised concept drift detection on data streams.

9. Acknowledgements

This paper is based upon projects sponsored by CoordenaÆo de AperfeiÆoamento de Pessoal de NqvelSuperior – Coordination for the Improvement of Higher Level Personnel (grants PROEX #5881819/D and #302077/2017-0), FundaÆo Amazÿnia Paraense de Amparo Pesquisa – Sao Paulo Research Foundation (grants #2017/16548-6), and Conselho Nacional de Desenvolvimento Cientqfico e Tecnolgico – The Brazilian National Council for Scientific and Technological Development (grant #302077/2017-0). Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of CAPES, FAPESP nor CNPq.

10. Concluding remarks

This paper is specially motivated by the lack of theoretical foundation to ensure unsupervised concept drift detection on data streams. Statistical Learning Theory (SLT) was also taken as a basis to formulate our proposal. In fact, SLT has a strong connection to the whole theory we employed to ensure Algorithmic Stability. Such a link comes from the association of the Symmetrization Lemma (von Luxburg & Schölkopf, 2011) with the perturbations assumed on McDiarmid's Inequality (McDiarmid, 1989). From that, we formulated an upper bound to ensure that a model significantly changes along time (Inequality (24)), independently of the measure function.

We also formulated the Plover algorithm to unsupervisedly and incrementally detect concept drifts in data streams. Besides that, we designed an algorithm to estimate the window length parameter for Plover, supporting the selection of the minimal window length to compute its following steps. Results confirm that our proposal is capable of detecting drifts, but an adequate measure function must be selected, according to the application domain.

As next steps, we intend to integrate Plover with MOA, the Massive Online Analysis framework, in order to provide this algorithm to other researchers. We also intend to build up a taxonomy mapping the best as possible measure function to different application domains and analyze how the amount of past samples could impact learning.

Appendix A. Demonstrating McDiarmid's inequality

Suppose a function g applied to a sequence of n random variables $X_{1:m} = (X_i)_{i=1,2,\ldots,m}$ and its conditional expected value:

$$\mu(g, X_{1:i}) = \mathbb{E}[g(X_1, X_2, \ldots, X_m)|X_{1:i}] = \int_{X \in X_{(i+1):m}} g(X_1, X_2, \ldots, X_m) P_j(X_{1:m}|X_{1:i}) dX, \quad (A.1)$$

in which $X_{1:i}$ is the sub-sequence with the first i variables of $X_{1:m}$, and $X_{(i+1):m}$ the remaining $m-i$ terms. If $X_{1:m}$ are independent, then:

$$P(X_{1:m}|X_{1:i}) = \frac{\prod_{j=1}^{m} P_j(X_j)}{\prod_{j=1}^{i} P_j(X_j)} = \prod_{j=i+1}^{m} P_j(X_j). \quad (A.2)$$

Consequently:

$$\mu(g, X_{1:i}) = \int_{X \in X_{(i+1):m}} g(X_1, X_2, \ldots, X_m) \prod_{j=i+1}^{m} P_j(X_j) dX. \quad (A.3)$$

Let $\mu(g, X_{1:i})$ be a function defined over the domain $dom(\mu(g, X_{1:i})) = X_{1:i}$, the integration area for the expected value is defined by the random variable X_i:

$$\mu(\mu(g, X_{1:i}), X_{1:(i-1)}) = \mathbb{E}\Big[\mathbb{E}[g(X_1, X_2, \ldots, X_m)|X_{1:i}]|X_{1:(m-1)}\Big].$$

In addition:

$$P(X_{1:i}|X_{1:(i-1)}) = \frac{\prod_{j=1}^{i} P_j(X_j)}{\prod_{j=1}^{i-1} P_j(X_j)} = P_i(X_i), \tag{A.4}$$

thus, similarly to Eq. (A.3):

$$\begin{aligned}
&\mu(\mu(g, X_{1:i}), X_{1:(i-1)}) \\
&= \int_{X_i} \left[\int_{X \in X_{(i+1):m}} g(X_1, X_2, \ldots, X_m) \prod_{j=i+1}^{m} P_j(X_j) dX \right] P_i(X_i) dX_i \\
&= \int_{X \in X_{i:m}} g(X_1, X_2, \ldots, X_m) \prod_{j=i}^{m} P_j(X_j) dX \\
&= \mu(g, X_{1:(i-1)})
\end{aligned} \tag{A.5}$$

Therefore, considering $\mu(g|X_{1:n})$ as a new random variable Y_n, by Eq. (A.5) we have $\mathbb{E}[Y_n|X_1, X_2, \ldots, X_{n-1}] = Y_{n-1}$ and, consequently, this forms a martingale process. In addition, it produces a martingale of differences given by $V_i = \mu(g|X_{1:n}) - \mu(g|X_{1:(n-1)})$ in such a way that $\mathbb{E}[V_n|X_1, X_2, \ldots, X_{n-1}] = 0$. Proving that V_n is bounded, we can rely on Azuma's Inequality to obtain the probabilistic convergence to the sum U_i of consecutive V_i's such as defined in Eq. (17).

The concept of this proof relies on upper and lower bounds U_i and L_i, respectively, for V_i. For instance, let b represent a new variable, the martingale of differences V_i is given by $\mathbb{E}[g|X_{1:i-1}, b] - \mathbb{E}[g|X_{1:i-1}]$ having $L_i \leq B_i \leq U_i$. So, by defining $U_i = \sup_u \mathbb{E}[g|X_{1:i-1}, u] - \mathbb{E}[g|X_{1:i-1}]$ and $L_i = \inf_l \mathbb{E}[g|X_{1:i-1}, l] - \mathbb{E}[g|X_{1:i-1}]$, we have:

$$\begin{aligned}
&U_i - L_i \leq \sup_{u,l} \mathbb{E}\left[g|X_{1:(i-1)}, u\right] - \sup_{u,l} \mathbb{E}\left[g|X_{1:(i-1)}, l\right] \\
&\leq \sup_{u,l} \left(\int_{X \in X_{(i+1):n}} \left[g(X_{1:(n-1)}, u, X_{(i+1):n}) - g(X_{1:(i-1)}, l, X_{(i+1):n}) \right] \right. \\
&\left. \prod_{j=i+1}^{n} P_j(X_j) dX \right) \\
&\text{(por desigualdade de Jensen)} \\
&\leq \int_{X \in X_{(i+1):n}} \sup_{u,l} \left[g(X_{1:(n-1)}, u, X_{(i+1):n}) - g(X_{1:(i-1)}, l, X_{(i+1):n}) \right] \\
&\prod_{j=i+1}^{n} P_j(X_j) dX.
\end{aligned} \tag{A.6}$$

Notice that McDiarmid's Inequality assumes:

$$\begin{aligned}
&\sup_{x_1, x_2, \ldots, x_n, x_i'} \left| g(x_1, x_2, \ldots, x_i, \ldots, x_n) - g(x_1, x_2, \ldots, x_i', \ldots, x_n) \right| \\
&\leq c_i,
\end{aligned} \tag{A.7}$$

and, thus:

$$U_i - L_i \leq \int_{X \in X_{(i+1):n}} c_i \prod_{j=i+1}^{n} P_j(X_j) dX = c_i. \tag{A.8}$$

Finally, if perturbations propagated over a function g, due to changes in random variables $(x_1, x_2, \ldots, x_i, \ldots, x_n)$, diverge in a bounded form, then, according to Azuma's Inequality, the application of g over independent random variables probabilistically converges to its expected value as demonstrated.

References

Abdulsalam, H., Skillicorn, D. B., & Martin, P. (2011). Classification using streaming random forests. *IEEE Transactions on Knowledge and Data Engineering, 23*(1), 22–36. doi:10.1109/TKDE.2010.36.

Agarwal, R. (1995). Dynamical systems and applications. *World Scientific series in applicable analysis*. World Scientific. https://books.google.com.br/books?id=iXAUnUX-Sg8C

Albertini, M. K., & de Mello, R. F. (2007). A self-organizing neural network for detecting novelties. In *Proceedings of the 2007 ACM symposium on applied computing SAC '07* (pp. 462–466). New York, NY, USA: ACM. doi:10.1145/1244002.1244110. http://doi.acm.org/10.1145/1244002.1244110

Angel, A. M., Bartolo, G. J., & Ernestina, M. (2016). Predicting recurring concepts on data-streams by means of a meta-model and a fuzzy similarity function. *Expert Systems with Applications, 46*, 87–105. doi:10.1016/j.eswa.2015.10.022. http://www.sciencedirect.com/science/article/pii/S0957417415007174

Azuma, K. (1967). Weighted sums of certain dependent random variables. *Tohoku Mathematical Journal, 19*(3), 357–367.

Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldá, R., & Morales-Bueno, R. (2006). Early drift detection method. *In fourth international workshop on knowledge discovery from data streams*.

Berndt, D. J., & Clifford, J. (1994). Using dynamic time warping to find patterns in time series. In *Proceedings of the 3rd international conference on knowledge discovery and data mining series AAAIWS'94* (pp. 359–370). AAAI Press. http://dl.acm.org/citation.cfm?id=3000850.3000887

Bifet, A., & Gavaldà, R. (2007). *Learning from time-changing data with adaptive windowing* (pp. 443–448)). SIAM. http://epubs.siam.org/doi/605abs/10.1137/1.9781611972771.42. arXiv: http://epubs.siam.org/doi/pdf/10.1137/1.9781611972771.42

Bifet, A., Holmes, G., Kirkby, R., & Pfahringer, B. (2010). MOA: Massive online analysis. *Journal of Machine Learning Research, 11*, 1601–1604. http://portal.acm.org/citation.cfm?id=1859903

Bifet, A., Holmes, G., Pfahringer, B., Kirkby, R., & Gavaldà, R. (2009). New ensemble methods for evolving data streams. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining KDD '09* (pp. 139–148). New York, NY, USA: ACM. doi:10.1145/1557019.1557041.

Bousquet, O., & Elisseeff, A. (2002). Stability and generalization. *Journal of Machine Learning Research, 2*, 499–526. doi:10.1162/153244302760200704.

de Carvalho Pagliosa, L., & de Mello, R. F. (2017). Applying a kernel function on time-dependent data to provide supervised-learning guarantees. *Expert Systems with Applications, 71*, 216–229. doi:10.1016/j.eswa.2016.11.028.

da Costa, F. G., Duarte, F. S. L. G., Vallim, R. M. M., & de Mello, R. F. (2017). Multidimensional surrogate stability to detect data stream concept drift. *Expert Systems with Applications, 87*, 15–29. doi:10.1016/j.eswa.2017.06.005.

Devroye, L., Györfi, L., & Lugosi, G. (1996). *A probabilistic theory of pattern recognition*. Springer.

Faria, E. R., Gama, J. a., & Carvalho, A. C. P. L. F. (2013). Novelty detection algorithm for data streams multi-class problems. In *Proceedings of the 28th annual ACM symposium on applied computing SAC '13* (pp. 795–800). New York, NY, USA: ACM. doi:10.1145/2480362.2480515.

Gama, J., Medas, P., Castillo, G., & Rodrigues, P. (2004). Learning with drift detection. In A. L. C. Bazzan, & S. Labidi (Eds.), *Advances in artificial intelligence – SBIA 2004: 17th Brazilian symposium on artificial intelligence, Sao Luis, Maranhao, Brazil, september 29-ocotber 1, 2004. proceedings* (pp. 286–295). Berlin, Heidelberg: Springer Berlin Heidelberg. doi:10.1007/978-3-540-28645-5_29.

Gama, J. a., Medas, P., & Rocha, R. (2004). Forest trees for on-line data. In *Proceedings of the 2004 ACM symposium on applied computing SAC '04* (pp. 632–636). New York, NY, USA: ACM. doi:10.1145/967900.968033.

Harel, M., Crammer, K., El-Yaniv, R., & Mannor, S. (2014). Concept drift detection through resampling. In *Proceedings of the 31st international conference on international conference on machine learning - volume 32 ICML'14*. JMLR.org. http://dl.acm.org/citation.cfm?id=3044805.3045005

Hayat, M. Z., & Hashemi, M. R. (2010). A dct based approach for detecting novelty and concept drift in data streams. In *2010 international conference of soft computing and pattern recognition* (pp. 373–378). doi:10.1109/SOCPAR.2010.5686734.

Hulten, G., Spencer, L., & Domingos, P. (2001). Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining KDD '01* (pp. 97–106). New York, NY, USA: ACM. doi:10.1145/502512.502529.

Jedrzejowicz, J., & Jedrzejowicz, P. (2015). Distance-based ensemble online classifier with kernel clustering. In R. Neves-Silva, L. C. Jain, & R. J. Howlett (Eds.), *Intelligent decision technologies: Proceedings of the 7th KES international conference on intelligent decision technologies (KES-IDT 2015)* (pp. 279–289). Cham: Springer International Publishing. doi:10.1007/978-3-319-19857-6_25.

Kantz, H., & Schreiber, T. (2003). *Nonlinear time series analysis* (2nd). Cambridge University Press. doi:10.1017/CBO9780511755798.

Klinkenberg, R., & Joachims, T. (2000). Detecting concept drift with support vector machines. In *Proceedings of the seventeenth international conference on machine learning ICML '00* (pp. 487–494). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc..

Krawczyk, B., & Woźniak, M. (2015). One-class classifiers with incremental learning and forgetting for data streams with concept drift. *Soft Computing, 19*(12), 3387–3400. doi:10.1007/s00500-014-1492-5.

Last, M. (2002). Online classification of nonstationary data streams. *Intell. Data Anal., 6*(2), 129–147. http://dl.acm.org/citation.cfm?id=1293986.1293988

Loo, H. R., & Marsono, M. N. (2015). Online data stream classification with incremental semi-supervised learning. In *Proceedings of the second ACM IKDD conference on data sciences CoDS '15* (pp. 132–133). New York, NY, USA: ACM. doi:10.1145/2732587.2732614.

von Luxburg, U., & Schölkopf, B. (2011). *Statistical learning theory: Models, concepts, and results* ((vol.10, pp. 651–706)). Amsterdam, Netherlands: Elsevier North Holland.

Marsland, S., Shapiro, J., & Nehmzow, U. (2002). A self-organising network that grows when required. *Neural Networks, 15*(8), 1041–1058. doi:10.1016/S0893-6080(02)00078-3. http://www.sciencedirect.com/science/article/pii/S0893608002000783

Masud, M. M., Chen, Q., Gao, J., Khan, L., Han, J., & Thuraisingham, B. (2010). Classification and novel class detection of data streams in a dynamic feature space. In *Proceedings of the 2010 european conference on machine learning and knowledge discovery in databases: Part ii ECML PKDD'10* (pp. 337–352). Berlin, Heidelberg: Springer-Verlag. http://dl.acm.org/citation.cfm?id=1888305.1888328

McDiarmid, C. (1989). On the method of bounded differences. In J. Siemons (Ed.), *Surveys in combinatorics, 1989: Invited papers at the twelfth british combinatorial conference London Mathematical Society Lecture Note Series* (pp. 148–188). Cambridge University Press. doi:10.1017/CBO9781107359949.008.

de Mello, R., & Ponti, M. (2018). *Machine learning: A practical approach on the statistical learning theory*. Springer International Publishing. https://books.google.fr/books?id=N3ectwEACAAJ

Mena-Torres, D., & Aguilar-Ruiz, J. S. (2014). A similarity-based approach for data stream classification. *Expert Syst. Appl., 41*(9), 4224–4234. doi:10.1016/j.eswa.2013.12.041.

Metzger, M. A. (1997). Applications of nonlinear dynamical systems theory in developmental psychology: Motor and cognitive development. *Nonlinear Dynamics, Psychology, and Life Sciences, 1*(1), 55–68. doi:10.1023/A:1022323926870.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika, 41*(1/2), 100–115. http://www.jstor.org/stable/2333009

Prasad, B. R., & Agarwal, S. (2017). Critical parameter analysis of vertical hoeffding tree for optimized performance using samoa. *International Journal of Machine Learning and Cybernetics, 8*(4), 1389–1402. doi:10.1007/s13042-016-0513-3.

Rendón, E., Abundez, I. M., Gutierrez, C., Zagal, S. D., Arizmendi, A., Quiroz, E. M., & Arzate, H. E. (2011). A comparison of internal and external cluster validation indexes. In *Proceedings of the 2011 american conference on applied mathematics and the 5th wseas international conference on computer engineering and applications American-Math'11/CEA'11* (pp. 158–163). Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS). http://dl.acm.org/citation.cfm?id=1959666.1959695

Rios, R. A., Parrott, L., Lange, H., & de Mello, R. F. (2015). Estimating determinism rates to detect patterns in geospatial datasets. *Remote Sensing of Environment, 156*, 11–20. doi:10.1016/j.rse.2014.09.019. http://www.sciencedirect.com/science/article/pii/S0034425714003642

Ross, G. J., Adams, N. M., Tasoulis, D. K., & Hand, D. J. (2012). Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters, 33*(2), 191–198. doi:10.1016/j.patrec.2011.08.019. http://www.sciencedirect.com/science/article/pii/S0167865511002704

Sethi, T. S., & Kantardzic, M. (2017). On the reliable detection of concept drift from streaming unlabeled data. *Expert Syst. Appl., 82*, 77–99.

Sethi, T. S., Kantardzic, M. M., & Hu, H. (2016). A grid density based framework for classifying streaming data in the presence of concept drift. *J. Intell. Inf. Syst., 46*(1), 179–211.

Shannon, C. E. (1949). Communication in the presence of noise. *Proceedings of the IRE, 37*(1), 10–21. doi:10.1109/JRPROC.1949.232969.

Spinosa, E. J., de Leon F. de Carvalho, A. P., & Gama, J. a. (2007). Olindda: A cluster-based approach for detecting novelty and concept drift in data streams. In *Proceedings of the 2007 acm symposium on applied computing SAC '07* (pp. 448–452). New York, NY, USA: ACM. doi:10.1145/1244002.1244107.

Vapnik, V. (1998). *Statistical learning theory*. Wiley.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York, NY, USA: Springer-Verlag New York, Inc..

Wang, L., Ji, H., & Jin, Y. (2013a). Fuzzy passive-aggressive classification: A robust and efficient algorithm for online classification problems. *Information Sciencies, 220*, 46–63.

Wang, S., Minku, L. L., Ghezzi, D., Caltabiano, D., Tino, P., & Yao, X. (2013b). Concept drift detection for online class imbalance learning. In *The 2013 international joint conference on neural networks (ijcnn)* (pp. 1–10). doi:10.1109/IJCNN.2013.6706768.

Widmer, G., & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning, 23*(1), 69–101. doi:10.1007/BF00116900.

Yang, Y.-B., Feng, L.-L., Pan, J., & Yang, X.-H. (2009). An optimal method for the power spectrum measurement. *Research in Astronomy and Astrophysics, 9*(2), 227. http://stacks.iop.org/1674-4527/9/i=2/a=012