Towards non-parametric drift detection via Dynamic Adapting Window Independence Drift Detection (DAWIDD)

Fabian Hinder [1] **André Artelt** [1] **Barbara Hammer** [1]

Abstract

The notion of concept drift refers to the phenomenon that the distribution, which is underlying the observed data, changes over time; as a consequence machine learning models may become inaccurate and need adjustment. Many online learning schemes include *drift detection* to actively detect and react to observed changes. Yet, reliable drift detection constitutes a challenging problem in particular in the context of high dimensional data, varying drift characteristics, and the absence of a parametric model such as a classification scheme which reflects the drift. In this paper we present a novel concept drift detection method, Dynamic Adapting Window Independence Drift Detection (DAWIDD), which aims for non-parametric drift detection of diverse drift characteristics. For this purpose, we establish a mathematical equivalence of the presence of drift to the dependency of specific random variables in an according drift process. This allows us to rely on independence tests rather than parametric models or the classification loss, resulting in a fairly robust scheme to universally detect different types of drift, as it is also confirmed in experiments.

1. Introduction

One fundamental assumption in classical machine learning is the fact that observed data are i.i.d. according to some unknown underlying probability measure P_X, i.e. the data generating process is stationary. Yet, this assumption is often violated as soon as machine learning faces real world problems: models are subject to seasonal changes, changed demands of individual costumers, ageing of sensors, etc. In such settings, life-long model adaptation rather than classical batch learning is required for optimum performance. Since drift, i.e. the fact that data is no longer identically distributed, is a major issue in many real-world applications of machine learning, many attempts were made to deal with this setting (Ditzler et al., 2015).

Depending on the domain of data and application, the presence of drift is modelled in different ways. As an example, covariate shift refers to the situation of training and test set having different marginal distributions (Gretton et al., 2009). Learning for data streams extends this setting to an unlimited (but usually countable) stream of observed data, mostly in supervised learning scenarios (Gama et al., 2014). Here one distinguishes between virtual and real drift, i.e. non-stationarity of the marginal distribution only or also the posterior. Learning technologies for such situations often rely on windowing techniques, and adapt the model based on the characteristics of the data in an observed time window. Active methods explicitly detect drift, usually referring to drift of the classification error, and trigger model adaptation this way, while passive methods continuously adjust the model (Ditzler et al., 2015).

Interestingly, a majority of approaches deals with supervised scenarios, aiming for a small interleaved train-test error; this is accompanied by first approaches to identify particularly relevant features where drift occurs (Webb et al., 2017), and a large number of methods, which aim for a detection of drift and an identification of change points in given data sets (Aminikhanghahi and Cook, 2017). These techniques often rely on strong assumptions as regards the process, e.g. they detect a substantial decrease of the classification accuracy of a specific classification scheme on the given data, or they judge important characteristics of the distribution which are estimated on time windows of fixed size – as a consequence, these methods face problems if the underlying drift characteristics do not align with these assumptions. The purpose of our contribution is threefold:

(I) We formalize two different notions of drift which are used in the literature, namely drift as change of probabilities, and drift as change of a loss function, which is used for popular drift detection methods, and we show the equivalence of these notions; (II) we provide a novel mathematical

[1]Cognitive Interaction Technology (CITEC), Bielefeld University, Inspiration 1, D-33619 Bielefeld, Germany. Correspondence to: Fabian Hinder <fhinder@techfak.uni-bielefeld.de>, André Artelt <aartelt@techfak.uni-bielefeld.de>, Barbara Hammer <bhammer@techfak.uni-bielefeld.de>.

Proceedings of the 37^{th} International Conference on Machine Learning, Vienna, Austria, PMLR 119, 2020.

characterization of drift in terms of independence of random variables in a drift process and we prove the equivalence of this formalization to the notion of drift as used in the literature; due to page limitations, all proofs in this contribution can be found in the supplement; (III) based thereon, we provide a new drift detection method that relies on independence tests, and this way, neither depends on an underlying machine learning model nor relies on assumptions on the underlying form of drift such as rate of change.

This paper is organized as follows: In section 2 (Concept Drift Definition) we give a formal definition of concept drift and analyse its mathematical properties; in particular, we formalize in how far drift detection via an observed change of a model error – a common procedure for many drift detection technologies – can be linked to the standard definition of drift as change of an underlying probability (section 2.1); after summarizing existing drift detection technologies, we derive an equivalent formalization of the presence of concept drift as the dependency of random variables (section 2.2). This fact constitutes the foundation for a novel drift detection method, which we construct in section 3: Dynamic Adapting Window Independence Drift Detection (DAWIDD). We compare this new method to popular alternatives in section 4. Note that the method as presented in sections 3 and 4 relies on section 2 only as concerns its mathematical substantiation and formal guarantees, but, otherwise, the proposed method DAWIDD can be accessed without delving deep into the presented mathematical details.

2. Concept Drift Definition

In the usual, time invariant setup of machine learning one considers a generative process P_X, i.e. a probability measure, on $\mathbb{R}^d$. In this context one views the realizations of P_X-distributed random variables $X_1, ..., X_n$ as samples. Depending on the objective, learning algorithms try to infer the data distribution based on these samples or, in the supervised setting, a posterior distribution. We will not distinguish these settings and only consider distributions in general, this way subsuming the notion of both, real drift and virtual drift.

Many processes in real-world applications are not time independent, so it is reasonable to incorporate time into our considerations. One prominent way to do so is to consider an index set $\mathfrak{T}$, representing time, and a collection of probability measures p_t on $\mathbb{R}^d$ indexed over $\mathfrak{T}$, which may change over time (Gama et al., 2014). In the following we investigate the relationship of those p_t, with drift referring to a property of the relationship of several p_t at different time points t. However, rather than using $\mathfrak{T} = \{1, 2, ..., N\}$, as done for example by (Bifet and Gavaldà, 2007; Gama et al., 2004; Ditzler and Polikar, 2011), we will consider the more general case $\mathfrak{T} = [0, 1]$ so that we may incorporate the actual clock-time, rather than a simple index, into our considerations. This yields the following definition:

Definition 1. A *drift process* (p_t, P_T) is a probability measure P_T on $[0, 1]$ together with a collection of probability measures p_t on $\mathbb{R}^d$ with $t \in [0, 1]$, such that $t \mapsto p_t(A)$ is measurable for every measurable $A \subset \mathbb{R}^d$.

When P_T is clear, we sometimes just write p_t for simplicity.

Remark 1. For every drift process (p_t, P_T) there exists a probability measure P on $\mathbb{R}^d \times [0, 1]$ which is uniquely determined by the property

$$P(B \times A) = \int_B p_t(A) \mathrm{d}P_T(t)$$

for all $B \subset [0, 1], A \subset \mathbb{R}^d$ measurable (Friedman, 1980). In the following we will denote this measure as $p_t \otimes P_T$. The converse is also true: For every probability measure P on $\mathbb{R}^d \times [0, 1]$ there exists a uniquely determined drift process (p_t, P_T) such that (Parthasarathy, 1967)

$$P = p_t \otimes P_T.$$

Remark 1 shows the main benefit in considering time directly as part of the observation, rather than just an index, since it enables us to link drift processes to distributions on the product space of data and time, which in turn implies that sampling a sequence of observations $X_1, ..., X_n$ at several time points t_i with different distribution p_{t_i} is equivalent to sampling $(X_1, T_1), ..., (X_n, T_n)$ i.i.d. from $p_t \otimes P_T$.

We will now define drift: A very common notion specifies drift as the fact that distributions vary over time (Gama et al., 2014), i.e. there exist $t, s \in [0, 1]$ such that $p_t \neq p_s$. A canonical extension of this definition for continuous time is given by the following setting, which defines the absence of drift as differences of distributions not being observable in a null set:

Definition 2. Let (p_t, P_T) be a drift process. We say that p_t has *no drift* iff $p_t \neq p_s$ holds on a P_T null set only, i.e. $(P_T \times P_T)(\{(s, t) \in [0, 1]^2 \mid p_t \neq p_s\}) = 0$.

Definition 2 is given by comparing p_t at pairs of time points; an obvious question is whether this can be simplified to the fact that the probability distribution is constant, i.e. p_t does not depend on t (up to a null set) – this corresponds to the standard setting of classical (drift free) machine learning. Note that this is not an immediate consequence due to null-sets and requires additional argumentation:

Lemma 1. *Let (p_t, P_T) be a drift process. The following are equivalent:*

1. *p_t has no drift*
2. *it exists P_X such that $p_t = P_X$ for P_T-a.s.*

3. it exists P_X such that $p_t \otimes P_T = P_X \times P_T$
Furthermore, if the probability measure P_X exists it is uniquely determined and it holds $P_X = \int p_t P_T(\mathrm{d}t)$.

Proof. All proofs are omitted due to space restrictions and may be found in the supplemental material. □

2.1. Change of Loss as Indicator for Drift

So far we have only considered drift from a theoretical, generative process point of view; now we will consider drift in the context of machine learning models: machine learning models in the context of drift often learn a constant model over a time window. It is common practice to detect drift by a change of such model, e.g. a changed error or loss; more precisely suppose we observe samples $x_1, ..., x_n \in \mathbb{R}^d$ ordered by time of occurrence and let $\hat{\ell}$ be an empirical loss function. If for some i we have $\hat{\ell}(h|x_1, ..., x_i) \ll \hat{\ell}(h|x_{i+1}, ..., x_n)$, where h is the model we consider, a drift alert is given. Drift detection methods often aim to detect significant changes of the loss and, more specifically, identify this change-point i. Following (Gama et al., 2014) popular drift detection methods can be assigned to three groups:

1. *Sequential Analysis based approaches* follow the idea of the Sequential Probability Ratio Test (SPRT) (Wald, 1945) by comparing the sum of characteristics computed from the input signals against a threshold; if the threshold is exceeded, drift is detected. The idea is that expected input signals are small, while unexpected signals tend to be large; therefore the sum tends to be larger if more unexpected events happen. Prominent examples are the cumulative sum (CUSUM) and its variant the Page-Hinkley test (PH) (PAGE, 1954).

2. *Statistical Process Control based approaches* are mainly applied to classification tasks where the outcome of a prediction can be modeled as a Bernoulli process linking the problem to time-series. Concept drift is then detected using statistical parameters such as mean or variation, which is common practice in the analysis of drift in time-series. Prominent examples are the Drift Detection Method (DDM) (Gama et al., 2004), Early Drift Detection Method (EDDM) (Baena-García et al., 2006), Exponentially Weighted Moving Average (EWMA) (Ross et al., 2012), and Reactive Drift Detection Method (RDDM) (Barros et al., 2017).

3. *Two time-window based approaches* make use of a pair of windows; one (usually fixed) representing the past distribution - one (usually sliding along the stream) representing the current distribution. If those windows differ significantly (i.e. fail on a statistical test for equal distribution) drift is detected. A similar approach is to successively split a single window into two and compare those. Prominent examples are the Adaptive Windowing (ADWIN) (Bifet and Gavaldà, 2007), the Drift Detection Methods based on Hoeffding's Bound (HDDMA-test and HDDMW-test) (Frías-Blanco et al., 2015), Hellinger Distance Drift Detection Method (HDDDM) (Ditzler and Polikar, 2011) and the comparable Change Detection Test (CDT) (Bu et al., 2018).

Most of the methods mentioned above only work for one-dimensional, or even binary, data. It is therefore very common to consider the loss of some machine learning model, instead of the actual data itself. This is beneficial since it reduces a potentially very high dimensional problem to a one dimensional or, in case of classification, binary one. We will refer to those methods as loss-based.

There do exist methods that do not refer to the loss of a learning model, and rely on the data instead: e.g HDDDM compares marginal distributions based on histograms and CDT approximates the density functions using RBF-networks; yet, the majority of technologies detects drift based on a loss-function of a machine learning model. In the following, we want to substantiate this practice by stating under which conditions this is equivalent to observing drift as defined above. For this purpose, we rely on a general definition of a loss function ℓ as an indicator for how well a hypothesis $h \in \mathcal{H}$ fits some ground truth, i.e. probability distribution P, which is usually unknown and therefore approximated using observations – leading to an empirical estimation $\hat{\ell}$ of ℓ, which is often done by using the empirical mean.

Definition 3. Let $\mathcal{H}$ be a hypothesis class and $\mathfrak{X}$ be a measure space (usually $\mathfrak{X} = \mathbb{R}^d$). An *empirical loss function* is a map $\hat{\ell} : \mathcal{H} \times (\coprod_{n=0}^{\infty} \prod_{i=1}^{n} \mathfrak{X}) \to \mathbb{R}$, such that for every set of $\mathfrak{X}$-valued random variables $X_1, ..., X_n$ and hypothesis $h \in \mathcal{H}$ we obtain a measurable map $\hat{\ell}(h|X_1, ..., X_n) : \Omega \to \mathbb{R}$ which measures the error of h on the random samples delivered by $X_1, ..., X_n$.

We say that an empirical loss function $\hat{\ell}$ *decomposes into sums* for $X_1, X_2, ..., X_N$ (with $N \in \mathbb{N} \cup \{\infty\}$) if $\hat{\ell}(h|X_1, ..., X_n) = \frac{1}{n} \sum_{i=1}^{n} \hat{\ell}(h|X_i)$ holds for all $n \leq N$.

We say that an empirical loss function is *uniformly bounded* if there exists an $K < \infty$ such that $|\hat{\ell}(h|x_1, ..., x_n)| < K$ for all $x_1, ..., x_n \in \mathfrak{X}$ and $h \in \mathcal{H}$.

As an example: the negative log likelihood decomposes into sums under the assumption of independency; the mean squared error (MSE) or the 0-1-loss always decomposes into sums, with the letter being uniformly bounded by $K = 1$.

Using this definition we may formalize the idea of "change of loss between time-windows" giving rise to theorem 1 which basically states that the change of a specific, empirical loss function is, up to approximation error, bounded

by the total variance of the mean distributions during the considered time windows.

Theorem 1. *Let $\hat{\ell}$ be an empirical loss function on a hypothesis class $\mathcal{H}$ which is uniformly bounded by some $K < \infty$. Let $X_1, ..., X_n$ and $X'_1, ..., X'_m$ be two sets of independent random variables for which $\hat{\ell}$ decomposes into sums. Then for all $h \in \mathcal{H}$ and $\varepsilon > 0$ it holds*

$$\mathbb{P}[|\hat{\ell}(h|X_1, ..., X_n) - \hat{\ell}(h|X'_1, ..., X'_m)| \geq \varepsilon] \leq \frac{K}{\varepsilon}\sqrt{\left(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}}\right)^2 + \left\|\frac{1}{n}\sum_{i=1}^{n}\mathbb{P}_{X_i} - \frac{1}{m}\sum_{i=1}^{m}\mathbb{P}_{X'_i}\right\|_{\text{TV}}^2},$$

where $\|\cdot\|_{\text{TV}}$ denotes the total variation norm and $\mathbb{P}_X$ the underlying distribution of X. In particular if all X_i resp. X'_j are also identically distributed the bound becomes

$$\mathbb{P}[|\hat{\ell}(h|X_1, ..., X_n) - \hat{\ell}(h|X'_1, ..., X'_m)| \geq \varepsilon] \leq K/\varepsilon\sqrt{(n^{-1/2} + m^{-1/2})^2 + \left\|\mathbb{P}_{X_1} - \mathbb{P}_{X'_1}\right\|_{\text{TV}}^2}.$$

In a typical application one considers resp. compares the empirical loss of a fixed model using the samples contained in two time windows, in theorem 1 those samples are referred to as $X_1, ..., X_n$ and $X'_1, ..., X'_m$, respectively. It is hence of particular importance that we do not assume any relation between X_i and X'_j, in particular we do not assume them to be independent. Indeed, we actually allow that the samples are reused in the estimation, i.e. $X_i = X'_j$. Furthermore, we also do not assume that the samples within a single time window, i.e. the X_is resp. X'_js, are identically distributed, i.e. we allow drift within a single time window. In this sense theorem 1 upper bounds the probability of a significant change of loss by a change of the underlying distribution.

Unfortunately, we cannot apply theorem 1 to unbounded loss-functions, like MSE, directly. However, we can still use it to compare the resulting distributions. More precisely we have:

Lemma 2. *Let $\hat{\ell}$ be an empirical loss function and $X_1, ..., X_n$ be random variables for which $\hat{\ell}$ decomposes into sums. Denote by $F_{\hat{\ell}(h|X_1,...,X_n)}(x)$ the empirical cumulative distribution over $\hat{\ell}(h|X_1, ..., X_n)$, i.e.*

$$F_{\hat{\ell}(h|X_1,...,X_n)}(x) = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}_{(\hat{\ell}(h|X_i),\infty)}(x).$$

Then for every $x \in \mathbb{R}$ we have that $F_{\hat{\ell}(h|X_1,...,X_n)}(x)$ is again an empirical loss function that decomposes into sums with $K = 1$.

Remark 2. Theorem 1 together with lemma 2 imply that every quantity that is derived from an empirical loss function that decomposes into sums, may only be non-constant if the underlying data distribution changes, which is therefore directly linked to all loss-based methods, i.e. if a loss-based method detects a change, which is not due to approximation errors, then the underlying distributions differ.

We would like to link theorem 1 to the notion of drift in the sense of definition 2. As it turns out the change of loss functions that decompose into sums is, up to approximation errors, bounded above by the total variation of the expected distribution during that time-windows, i.e. we have the following corollary:

Corollary 1. *Let (p_t, P_T) be a drift process and $\hat{\ell}$ be an empirical loss function on a hypothesis class $\mathcal{H}$ which is uniformly bounded by some $K < \infty$. Let $(X_1, T_1), ..., (X_n, T_n) \sim p_t \otimes P_T$ and $(X'_1, T'_1), ..., (X'_n, T'_n) \sim p_t \otimes P_T$ be independent random variables. Then for all $h \in \mathcal{H}$, $A, B \subset [0, 1]$ measurable with $P_T(A), P_T(B) > 0$ and $\varepsilon > 0$ it holds*

$$\mathbb{P}[|\hat{\ell}(h|\mathbf{X}) - \hat{\ell}(h|\mathbf{X}')| \geq \varepsilon|\mathbf{T} \in A, \mathbf{T}' \in B] \leq \frac{K}{\varepsilon}\sqrt{(n^{-1/2} + m^{-1/2})^2 + \|p_A - p_B\|_{\text{TV}}^2},$$

where $p_A = P_T(A)^{-1}\int_A p_t(\cdot)P_T(\mathrm{d}t)$ and p_B analogous and we used the shorthands $\hat{\ell}(h|\mathbf{X}) = \hat{\ell}(h|X_1, ..., X_n)$ and $\mathbf{T} \in A :\iff T_1 \in A, ..., T_n \in A$ and analogous for $\mathbf{X}'$ and $\mathbf{T}'$.

As in theorem 1 we allow drift during the time windows A and B. In contrast to theorem 1, corollary 1 allows us to control the bound using the observed time-value, which is beneficial since, thought we do not know the underlying distribution of X_i, we do know the value of T_i.

Furthermore, lemma 1 shows that statistically significant change of the empirical loss between two time windows A and B implies a change of the underlying distributions p_A resp. p_B between those time windows. This gives rise to the question whenever we can always detect such changes of the underlying distribution using a loss-based method. The answer to that question is "yes", as we show in the following lemma:

Lemma 3. *Let p_t be a drift process. If p_t has (model) drift then we may find a model h and an empirical loss function $\hat{\ell}$ such that*

$$|\hat{\ell}(h|X_1, ..., X_n) - \hat{\ell}(h|X'_1, ..., X'_n)| \xrightarrow{a.s.} \|p_A - p_B\|_{\text{TV}},$$

with $X_1, X_2, ... \sim p_A$, $X'_1, X'_2, ... \sim p_B$ independent.

Though, this shows that we can detect a change of the underlying distribution using model loss, one should notice that the choice of $\hat{\ell}$ and h may be very exotic and depend on A and B so that it is not clear – in a concrete setup –

which occurrences of change of distribution are detected under which circumstances.

However, by corollary 1 and lemma 3 we have proven that a significant change of empirical loss and a change of underlying distributions between two time windows are equivalent; this leads to the following definition:

Definition 4. We say that a drift process (p_t, P_T) has *model drift* iff there exists measurable sets $A, B \subset [0,1]$ with $P_T(A), P_T(B) > 0$, such that $p_A \neq p_B$ or equivalent $\|p_A - p_B\|_{\text{TV}} > 0$, with $p_A = P_T(A)^{-1} \int_A p_t(\cdot) P_T(\mathrm{d}t)$ and analogous for p_B.

So by reconsidering remark 2 and lemma 3 we see that model drift and change of our empirical loss function (up to approximation errors) and therefore any statistical quantity derived from it, are equivalent. In particular this shows that model drift is indeed exactly the class of effects that is detected by loss-based methods. It is therefore natural to ask whenever model drift implies drift (in the sense of definition 2) and vice versa – which is true as shown by the following theorem:

Theorem 2. *Let (p_t, P_T) be a drift process. Then it holds that p_t has drift if and only if p_t has model drift.*

So we see that loss based methods try to interfere drift. However, since this is only executable if the "right" model was chosen it remains to derive a notion of drift that gives rise to feasible algorithms.

2.2. Drift as Dependency between Data and Time

In addition to these notions of drift from the literature, we will now discuss drift under a novel aspect, which will be particularly suited to derive efficient algorithms, namely in the context of independence of random variables, which has the pleasant property of being directly observable in data using well established tools.

In the classical machine learning setup one considers samples as realizations of (independent) identically distributed random variables. In the context of drift, this distribution changes, as discussed above. To put this into the context of dependence of variables, we can equip each sample with a timestamp of its occurrence: instead of $\mathbb{R}^d$-valued random variables X, we consider $\mathbb{R}^d \times [0,1]$-valued random variables (X, T). If there is no drift then the distribution of data X should not depend on time T, i.e. X and T should be statistically independent:

Definition 5. Let (p_t, P_T) be a drift process and let $(X, T) \sim p_t \otimes P_T$ a pair of random variables. We say that p_t has *dependency drift* iff X and T are statistically dependent, i.e. are not independent random variables.

Notice that, other than drift or model drift, dependency drift is directly observable in data. Furthermore we can understand dependency drift using common statistical explanation methods. It turns out that this is indeed an alternative characterization of drift:

Theorem 3. *Let (p_t, P_T) be a drift process. Then p_t has drift if and only if it has dependency drift.*

This result allows us to reduce many problems that occur in the context of drift to already known and well studied problems; the problem of drift detection for instance is reduced to the problem to test independence of random variables. The latter problem is well investigated and highly efficient algorithms exist for independence tests.

In addition, notice that we can quantify dependency drift using total variation between joint $p_t \otimes P_T$ and marginal distribution $P_X \times P_T$. This allows us to compare the different notions of drift in a quantitative fashion: If A, B are time windows then p_A and p_B are always harder to distinguish than $P_X \times P_T$ and $p_t \otimes P_T$, i.e. the loss obtained form learning independent vs. dependent bounds the one between time window A and time window B, which in turn upper bounds the difference of loss for a fixed model as seen in theorem 1, i.e. up to statistical imprecisions we have

$$\begin{aligned} 1/K|\hat{\ell}(h|\mathbf{X}) - \hat{\ell}(h|\mathbf{X}')| &\leq \|p_A - p_B\|_{\text{TV}} \\ &\leq \|P_X \times P_T - p_t \otimes P_T\|_{\text{TV}}. \end{aligned}$$

On the other hand, if there is abrupt drift between time windows A and B and it is the only drift in $[0,1]$ then it holds

$$\begin{aligned} 1/K|\hat{\ell}(h|\mathbf{X}) - \hat{\ell}(h|\mathbf{X}')| &= \|p_A - p_B\|_{\text{TV}} \\ &= \|P_X \times P_T - p_t \otimes P_T\|_{\text{TV}}, \end{aligned}$$

assuming we used a model that is maximally vulnerable to the drift (lemma 3). We therefore see that dependency drift, if measured using total variation, is an optimal upper bound of the change of loss in the sense that it measures the entire drift – whereas considering models or time windows may miss at least part of it due to inappropriate choices of model h or time windows A and B – but not more.

3. Drift Detection via Independence Tests on Dynamically Adapted Windows

In section 2 we gave a definition of concept drift which turned out to be exactly the kind of effect that is detected by drift detectors, as discussed in section 2.1. Furthermore in section 2.2 we discussed that concept drift may be described as the statistical dependency between random variables, more precisely data-points and clock-time of observation. In this section we use the latter result to construct a new drift detection method Dynamic Adapting Window Independence Drift Detection (DAWIDD). It is a direct consequence of theorem 3 to design drift detection as a

Algorithm 1 Dynamic Adaptive Window Independence Drift Detector (DAWIDD)

Input: (x_i) data stream, p p-value for statistical test, $n_{\min}$ minimal number of samples in window, $n_{\max}$ maximal number of samples in window
Initialize Window $W \leftarrow []$
repeat
  Receive new sample x_i at time t_i from stream (x_i)
  $W \leftarrow W \cup \{(x_i, t_i)\}$
  if Test(W, p) rejects H_0 **then**
    output Drift Alert
    Drop $|W| - n_{\min}$ elements from the tail of W
  end if
  while $|W| > n_{\max}$ **do**
    Drop element from W keeping distribution
  end while
until At end of stream (x_i)

dependency test: given $X_1, ..., X_n$ as samples; we add a clock-time random variable T_i, i.e. instead of $X_1, ..., X_n$ we consider $(X_1, T_1), ..., (X_n, T_n)$. These are identically distributed due to the fact that time has now become part of the samples. According to theorem 3, $X_1, ..., X_n$ are not identically distributed, i.e. they have drift, if and only if X_i and T_i are not statistically independent. This leads to the following idea: By performing an independence test between X_i and T_i we obtain a drift detector.

To run independence tests we need a certain amount of samples for significance. These may be kept in a sliding window, i.e. new samples (together with clock-time or a suitable surrogate) are added to the window as they arrive; an independence test is performed to detect drift; if drift is detected, samples are removed from the window to reach minimum; on the other hand, if the window exceeds a certain size, samples are randomly removed. This is numerically more stable than sliding along the stream, i.e. removing the oldest samples, since it better preserves signals which might indicate a gradual drift. The validity of random deletion is substantiated by the following lemma, that states that a removal of samples after a fixed time or number of samples will always change independence properties but random removal, which theoretical allows a sample to stay in the window forever, cannot:

Lemma 4. *Let (p_t, P_T) and (q_t, Q_T) be drift processes. Suppose $P_T(A) = 0 \Rightarrow Q_T(A) = 0$ for all measurable $A \in \mathfrak{B}([0,1])$ and that $p_t = q_t$ for P_T-a.s. all $t \in [0,1]$. Then it holds: if p_t has no drift then q_t has no drift.*

Using ADWIN (Bifet and Gavaldà, 2007) as an example and including our considerations we obtain algorithm 1. Notice however that we do not need to successively split our sliding window since we do not rely on a two-sample test.

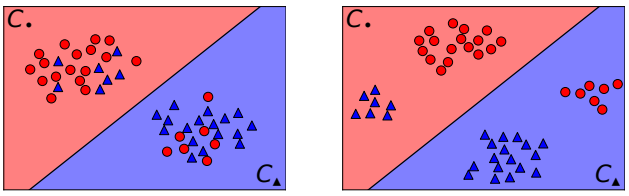
Assumptions regarding type of drift Mathematically, DAWIDD can deal with any type of change, which follows from theorem 3 and lemma 4: theorem 3 states that any form of drift is equivalent to dependency of random variables X and T representing data and time respectively, and therefore can be detected using independence tests. lemma 4 states that we do not have to keep all data points: it is suffices if every data point has strictly positive probability to be used by the test, which can be realized by sufficient sampling strategies, rather than using a sliding window. Hence algorithm 1 does not relies on assumptions on the underlying form of drift such as rate of change. A restriction is, of course, the validity of the used independence test on the observed data and the used sampling strategy.

3.1. Theoretical Comparison to existing Drift Detectors

Now that we defined our algorithm we may compare it to other approaches on a conceptual basis. Our considerations are summarized in Table 1.

Comparison to supervised drift detection Methods such as ADWIN (Bifet and Gavaldà, 2007), DDM (Gama et al., 2004) and EDDM (Baena-García et al., 2006) use the classification error as an indicator of drift (see section 2.1). They assume that drift leads to a change in accuracy. There do exist scenarios in which this assumption does not hold: Consider a binary classification data set with two clusters as shown in Figure 1a. Both clusters are mixtures of samples from both classes, but with different dominance. A linear classifier yields a decision boundary as shown in Figure 1a. Drift is constructed by moving all samples from one class along the decision boundary in the direction of the upper right corner, whereby we do not cross the decision boundary. The final scenario is shown in Figure 1b. Loss-based drift detectors do not detect this drift because the classification error does not change when moving the data points this way, unless the classifier is retrained. DAWIDD detects this drift since it does not rely on the classification error.

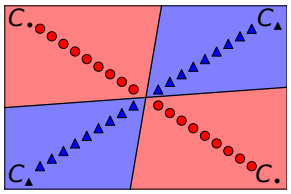
Comparison to unsupervised drift detection Another class of methods for drift detection is based on distribu-



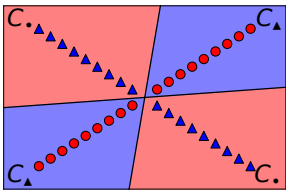
(a) A linear classifier fitted to the original data set.

(b) Drifted data set. Note that the accuracy does not change.

Figure 1. Fooling error based drift detection methods.



(a) A quadratic discriminant model fitted to the original data set.



(b) Drifted data set. Mean, variance and feature-wise marginals did not change.

Figure 2. Fooling simple distributional drift detectors.

tional changes (Kifer et al., 2004; Matteson and James, 2014; Dette and Wied, 2016; Vorburger and Bernstein, 2006; Ditzler and Polikar, 2011; Dasu et al., 2006; Song et al., 2007; Gretton et al., 2006). These methods try to detect drift by detecting changes in the sampling distribution of the data stream. Many of these methods (Kifer et al., 2004; Matteson and James, 2014; Dette and Wied, 2016; Vorburger and Bernstein, 2006; Ditzler and Polikar, 2011) use some kind of windowing - they split the data stream (or parts of it) into two windows and compute statistics on these windows. However, relying on two windows can be problematic because we have to select the right length of the window so that quickly occurring abrupt drifts are recognized – usually, it is assumed that the distribution of the samples in a window is fixed. Another problem of some of these methods is that they try to reduce computational complexity by assuming that the drift will show in the mean, variance or feature-wise marginals (Ditzler and Polikar, 2011; Vorburger and Bernstein, 2006). This is problematic because one can construct drifting data sets where the mean, variance and the feature-wise marginal distribution do not change - such drifts can not be perceived by methods that make these simplifying assumptions. For instance we can construct a data set where the points are arranged like a cross so that each class has its own diagonal - see Figure 2a. If the cross is symmetric and if the samples are placed symmetrically around the center, then we can swap the labels of the two diagonals - see Figure 2b - but the mean, variance and the feature-wise marginal distributions do not change. Therefore, these methods do not recognize the drift. However, our method is able to detect this drift since it does not make any simplifying assumptions about the distributional changes.

Speed of drift As discussed so far most drift detection methods focus, directly or indirectly, on the difference of (mean) distributions between two time-windows (see section 2.1). This makes it hard to detect both very slow and very fast (gradual) drift, i.e. the distribution does not change abrupt but in a continuous fashion. The issue with slow drift is that it may only be detected, due to too small changes, if the chosen window is large enough; however a large window - besides needing more memory - will make the drift seemingly faster. Fast drift on the other hand is even more problematic: If the window is chosen too large then the drift may vanish because it does not affect mean values, i.e. the changes cancel out – if the window is chosen too small the amount of data that can be used for estimations is small, resulting in statistical inaccuracies. Since DAWIDD considers a property that is window intrinsic it suffers from neither of both problems.

Table 1. Comparison of different drift detectors

Method	Model free	Fast drift	Slow drift	General distr.
DAWIDD	✓	✓	✓	✓
ADWIN	✗	(✓)	✗	✓
DDM	✗	✗	✗	✓
HDDDM	✓	✗	(✓)	✗

4. Experiments

We evaluate and compare DAWIDD with different state-of-the art drift detection methods - we use HDDDM (Ditzler and Polikar, 2011), DDM (Gama et al., 2004), EDDM (Baena-García et al., 2006) and ADWIN (Bifet and Gavaldà, 2007), since these methods cover representative different drift-detection schemes. We run our experiments on several standard benchmark data sets. For reasons of simplicity we used a sliding window in our implementation.[1]

Theoretical data We use the following theoretical data sets - each data set contains 4 concepts and thus 3 concept drifts: Rotating hyperplane (Montiel et al., 2018) (200 samples per concept), SEA (Street and Kim, 2001) (400 samples per concept) and RandomRBF (Montiel et al., 2018) (200 samples per concept).

Furthermore we developed two additional data sets: ”Blinking X” and ”mixing Gaussians”. Both are two dimensional and have two classes. The Blinking X data set is shown in

[1]The code is available at https://github.com/FabianHinder/DAWIDD

Table 2. Rank statistic over all data sets. Results were ranked for after every run, with rank 1 always being the best, then mean is taken over all runs and data sets.

Method	TP	FN	FP	Delay
DAWIDD	**2.2**	**2.2**	3.3	**2.4**
HDDDM	3.4	3.4	**1.8**	3.3
EDDM	2.9	2.9	3.3	2.9
DDM	2.9	2.9	3.3	2.8
ADWIN	3.6	3.6	3.3	3.5

Table 3. Results on theoretical and real world benchmark data sets.

	Dataset	Method	TP	FN	FP	Delay
Theoretical	Rotating Hyperplane	DAWIDD	1.7(±0.21)	1.3(±0.21)	0.3(±0.21)	37.18
		HDDDM	0.45(±0.25)	2.55(±0.25)	0.55(±0.55)	16.67
		EDDM	1.25(±0.49)	1.75(±0.49)	1.35(±0.43)	39.64
		DDM	0.45(±0.25)	2.55(±0.25)	1.5(±3.15)	14.33
		ADWIN	0.3(±0.21)	2.7(±0.21)	2.05(±0.55)	35.0
	SEA	DAWIDD	0.7(±0.81)	2.3(±0.81)	6.05(±5.95)	29.93
		HDDDM	0.45(±0.55)	2.55(±0.55)	1.6(±1.44)	33.33
		EDDM	0.15(±0.13)	2.85(±0.13)	2.4(±0.54)	28.0
		DDM	1.0(±1.2)	2.0(±1.2)	7.95(±3.25)	13.45
		ADWIN	0.4(±0.24)	2.6(±0.24)	3.0(±0.6)	28.5
	Random RBF	DAWIDD	1.3(±0.21)	1.7(±0.21)	0.55(±0.25)	41.31
		HDDDM	0.55(±0.25)	2.45(±0.25)	0.0	13.64
		EDDM	0.7(±0.21)	2.3(±0.21)	1.7(±0.81)	34.5
		DDM	0.55(±0.25)	2.45(±0.25)	1.95(±1.75)	19.0
		ADWIN	0.25(±0.19)	2.75(±0.19)	2.45(±1.45)	24.6
	Blinking X	DAWIDD	1.35(±0.43)	1.65(±0.43)	0.65(±0.43)	39.0
		HDDDM	0.15(±0.13)	2.85(±0.13)	0.7(±0.21)	50.0
		EDDM	0.7(±0.21)	2.3(±0.21)	1.95(±1.75)	32.29
		DDM	0.25(±0.19)	2.75(±0.19)	1.8(±1.56)	32.8
		ADWIN	0.15(±0.13)	2.85(±0.13)	0.6(±1.14)	47.0
	Mixing Gaussians	DAWIDD	0.55(±0.25)	3.45(±0.25)	2.45(±0.25)	24.27
		HDDDM	0.7(±0.21)	3.3(±0.21)	0.45(±0.25)	28.57
		EDDM	0.3(±0.21)	3.7(±0.21)	2.85(±0.13)	23.5
		DDM	1.3(±0.81)	2.7(±0.81)	1.45(±1.15)	8.19
		ADWIN	0.5(±0.45)	3.5(±0.45)	3.85(±1.03)	21.4
Real	Weather	DAWIDD	1.4(±0.54)	2.6(±0.54)	6.55(±0.85)	25.0
		HDDDM	0.0	4.0	0.85(±0.13)	–
		EDDM	0.55(±0.25)	3.45(±0.25)	2.55(±0.85)	23.27
		DDM	0.55(±1.15)	3.45(±1.15)	1.7(±2.91)	22.64
		ADWIN	0.15(±0.13)	3.85(±0.13)	1.0(±0.6)	18.0
	Forest Cover Type	DAWIDD	1.4(±0.54)	2.6(±0.54)	7.55(±0.85)	31.82
		HDDDM	0.45(±0.55)	3.55(±0.55)	0.55(±0.25)	28.67
		EDDM	0.4(±0.24)	3.6(±0.24)	2.25(±2.29)	17.38
		DDM	0.3(±0.51)	3.7(±0.51)	1.75(±1.09)	29.5
		ADWIN	0.15(±0.13)	3.85(±0.13)	2.3(±1.71)	29.0
	Electricity Market	DAWIDD	0.15(±0.13)	3.85(±0.13)	1.3(±2.01)	21.0
		HDDDM	0.0	4.0	0.1(±0.09)	–
		EDDM	0.3(±0.21)	3.7(±0.21)	2.5(±0.75)	31.0
		DDM	1.2(±1.56)	2.8(±1.56)	2.85(±1.93)	20.42
		ADWIN	0.4(±0.34)	3.6(±0.34)	2.4(±1.14)	23.88

Figure 2 - each concept corresponds either to samples from the left or right plot in Figure 2, a concept drift simply swaps the labeling. The mixing Gaussians dataset consists of 4 Gaussians containing either 75 or 25 samples - each halve of the plane containing two Gaussians with 100 samples in total. At the beginning one halve contains 75 samples of class 1 (reps. 2) and 25 samples of class 2 (resp. 1) uniformly choose from each of the two respective Gaussians; after the drift occurred each Gaussian only contains samples of a single class. This process is repeated 4 times.

Real world data We use a total number of three real world data sets: "Electricity market prices data set" (Harries et al., 1999), "Forest Covertype data set" (Blackard et al., 1998) and the "Weather data set" (Elwell and Polikar, 2011). We do not rely on classification accuracy, but on the ability of methods for drift detection. Therefore, we present data points such that positions for drift are known: we randomly choose 4 points and permute the samples within the resulting intervals randomly.

Setup The supervised methods use a Gaussian Naive Bayes classifier. We used standard hyperparameter settings. DAWIDD uses a permutation based conditional independence test which uses RBF-SVMs as underlying model (Chalupka et al., 2018).

The task was to pinpoint drift events in streaming data. After a drift event occurs drift should be alerted within a time-window of 50 samples. If drift was detected it is considered as a true positive. Every alert outside such a time-window is considered as a false positive. The number of samples between the occurrence of the drift event and the first alert is referred to as delay.

We report the mean number and variance (if larger than 10^{-2}) of true positives, false negatives, false positives and drift detection delay over 80 repetitions of the experiments presented in Table 3. We also ranked each method in each experiment allowing as to gain an overall résumé shown in Table 2.

4.1. Results

We find that DAWIDD yields a competitive performance and is better than the other methods when it comes to detecting drift. However, notably DAWIDD yields more false alarms than some of the other methods, although not with respect to all data sets, which is due to the used independence test; an improvement of tests of this criterion are subject to future work.

4.2. Hyper Parameters and Stability

We evaluated the performance of DAWIDD with respect to different choices of hyper parameters, such as window size and independence test, as well as its stability with respect to dimensionality and data perturbation.

Window size We found no negative effect when increasing n_{max}, it hence should be chosen as large as possible. n_{min} appears to be a crucial parameter: if it is chosen too large, it can cause false positives, if it is too small, the test accuracy is reduces. In practice, the results are stable in a large range of choices, however.

Independence tests We tried several independence tests including the classical χ^2-test (F.R.S., 1900), the HSIC-test (Gretton et al., 2007) and the original FCI test (Chalupka et al., 2018). FCI suffers from a high computational complexity and appeared to be numerically instable if used with only few samples. Many tests yield a large amount of type I or type II errors. DAWIDD with KCI and FCI outperform all other methods but yield comparably high false positives.

Dimensionality We tested the sensitivity of DAWIDD to data dimensionality. We checked using 1 to 50 dimensional data with normal distributed noise and random linear mixing. The results show that DAWIDD used with KCI independence test is not affected, FCI only if random mixing is used. The RBF-SVM test suffers from dimensionality due to the used Monte Carlo like sampling scheme.

5. Discussion

We have introduced a formalization for the presence of drift in continuous time via a probability theoretic framework, which enables us to show the equivalence of the notion to change of a loss in an idealized window-based machine learning setting, and equivalence to independency of random variables induced by data and time. The latter formulation gave rise to a novel drift detection mechanism, which relies on independence tests, hence provides a step towards model-free methods which neither require the choice of a specific window size nor a specific loss such as the classification loss of a classification prescription. The novel method is yet restricted insofar as independence tests depend on hyper-parameters such as the choice of a kernel, threshold of a hypothesis test, and minimum and maximum window size, yet it performs more robustly if different types of drift are present in particular drift which does not show in an associated marginal or classification loss. One problem is yet given by a comparably high false positive rate, due to the vulnerability of independence test to provide false negatives for limited data sets with possible spurious correlations.

Acknowledgement

We gratefully acknowledge funding from the VW-Foundation for the project *IMPACT* funded in the frame of the funding line *AI and its Implications for Future Society*.

We gratefully acknowledge funding by the BMBF under grant number 01 IS 18041 A.

References

Aminikhanghahi, S. and Cook, D. J. (2017). A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, 51(2):339–367.

Baena-García, M., Campo-Ávila, J., Fidalgo-Merino, R., Bifet, A., Gavald, R., and Morales-Bueno, R. (2006). Early drift detection method.

Barros, R. S., Cabral, D. R., Gonçalves, P. M., and Santos, S. G. (2017). Rddm: Reactive drift detection method. *Expert Systems with Applications*, 90:344 – 355.

Bifet, A. and Gavaldà, R. (2007). Learning from time-changing data with adaptive windowing. In *Proceedings of the Seventh SIAM International Conference on Data*

Mining, April 26-28, 2007, Minneapolis, Minnesota, USA, pages 443–448.

Blackard, J. A., Dean, D. J., and Anderson, C. W. (1998). Covertype data set.

Bu, L., Alippi, C., and Zhao, D. (2018). A pdf-free change detection test based on density difference estimation. *IEEE Transactions on Neural Networks and Learning Systems*, 29(2):324–334.

Chalupka, K., Perona, P., and Eberhardt, F. (2018). Fast conditional independence test for vector variables with large sample sizes. *ArXiv*, abs/1804.02747.

Dasu, T., Krishnan, S., Venkatasubramanian, S., and Yi, K. (2006). An information-theoretic approach to detecting changes in multi-dimensional data streams. In *In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications*.

Dette, H. and Wied, D. (2016). Detecting relevant changes in time series models. *Journal of the Royal Statistical Society Series B*, 78(2):371–394.

Ditzler, G. and Polikar, R. (2011). Hellinger distance based drift detection for nonstationary environments. In *2011 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments, CIDUE 2011, Paris, France, April 13, 2011*, pages 41–48.

Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. (2015). Learning in nonstationary environments: A survey. *IEEE Comp. Int. Mag.*, 10(4):12–25.

Elwell, R. and Polikar, R. (2011). Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531.

Friedman, H. (1980). A consistent fubini-tonelli theorem for nonmeasurable functions. *Illinois J. Math.*, 24(3):390–395.

F.R.S., K. P. (1900). X. on the criterion that a given system of deviations from the probable in the case of a correlated system of variables is such that it can be reasonably supposed to have arisen from random sampling. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 50(302):157–175.

Frías-Blanco, I., d. Campo-Ávila, J., Ramos-Jiménez, G., Morales-Bueno, R., Ortiz-Díaz, A., and Caballero-Mota, Y. (2015). Online and non-parametric drift detection methods based on hoeffding's bounds. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):810–823.

Gama, J., Medas, P., Castillo, G., and Rodrigues, P. P. (2004). Learning with drift detection. In *Advances in Artificial Intelligence - SBIA 2004, 17th Brazilian Symposium on Artificial Intelligence, São Luis, Maranhão, Brazil, September 29 - October 1, 2004, Proceedings*, pages 286–295.

Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. J. (2006). A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 4-7, 2006*, pages 513–520.

Gretton, A., Fukumizu, K., Teo, C. H., Song, L., Schölkopf, B., and Smola, A. (2007). A kernel statistical test of independence.

Gretton, A., Smola, A., Huang, J., Schmittfull, M., Borgwardt, K., and Schölkopf, B. (2009). *Covariate shift and local learning by distribution matching*, pages 131–160. MIT Press, Cambridge, MA, USA.

Harries, M., cse tr, U. N., and Wales, N. S. (1999). Splice-2 comparative evaluation: Electricity pricing. Technical report.

Kifer, D., Ben-David, S., and Gehrke, J. (2004). Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases - Volume 30*, VLDB '04, pages 180–191. VLDB Endowment.

Matteson, D. S. and James, N. A. (2014). A nonparametric approach for multiple change point analysis of multivariate data. *Journal of the American Statistical Association*, 109(505):334–345.

Montiel, J., Read, J., Bifet, A., and Abdessalem, T. (2018). Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5.

PAGE, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1-2):100–115.

Parthasarathy, K. R. (1967). *Probability measures on metric spaces*, volume 3 of *Probability and mathematical statistics ; 3*. Acad. Pr., New York [u.a.].

Ross, G. J., Adams, N. M., Tasoulis, D. K., and Hand, D. J. (2012). Exponentially weighted moving average charts for detecting concept drift. *Pattern Recognition Letters*, 33(2):191 – 198.

Song, X., Wu, M., Jermaine, C., and Ranka, S. (2007). Statistical change detection for multi-dimensional data. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 667–676, New York, NY, USA. ACM.

Street, W. N. and Kim, Y. (2001). A streaming ensemble algorithm (SEA) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, San Francisco, CA, USA, August 26-29, 2001*, pages 377–382.

Vorburger, P. and Bernstein, A. (2006). Entropy-based concept shift detection. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006), 18-22 December 2006, Hong Kong, China*, pages 1113–1118.

Wald, A. (1945). Sequential tests of statistical hypotheses. *The Annals of Mathematical Statistics*, 16(2):117–186.

Webb, G. I., Lee, L. K., Petitjean, F., and Goethals, B. (2017). Understanding concept drift. *CoRR*, abs/1704.00362.