

Suitability of Different Metric Choices for Concept Drift Detection*

Fabian Hinder

Cognitive Interaction Technology (CITEC)
Bielefeld University
Inspiration 1, D-33619 Bielefeld, Germany
fhinder@techfak.uni-bielefeld.de

Valerie Vaquet

Cognitive Interaction Technology (CITEC)
Bielefeld University
Inspiration 1, D-33619 Bielefeld, Germany
vvaquet@techfak.uni-bielefeld.de

Barbara Hammer

Cognitive Interaction Technology (CITEC)
Bielefeld University
Inspiration 1, D-33619 Bielefeld, Germany
bhammer@techfak.uni-bielefeld.de

February 22, 2022

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 unsupervised approaches for drift detection rely on measuring the discrepancy between the sample distributions of two time windows. This may be done directly, after some preprocessing (feature extraction, embedding into a latent space, etc.), or with respect to inferred features (mean, variance, conditional probabilities etc.). Most drift detection methods can be distinguished in what metric they use, how this metric is estimated, and how the decision threshold is found. In this paper, we analyze structural properties of the drift induced signals in the context of different metrics. We compare different types of estimators and metrics theoretically and empirically and investigate the relevance of the single metric components. In addition, we propose new choices and demonstrate their suitability in several experiments.

*We gratefully acknowledge funding by the BMBF TiM, grant number 05M20PBA.

1 Introduction

One popular assumption in classical machine learning is that the observed data is generated i.i.d. according to some unknown underlying and stationary probability \mathbb{P}_X . Yet, stationarity is often violated for realistic learning tasks such as machine learning based on (streaming) social media entries or measurements of IoT devices, which are subject to continuous change [2, 23]. Here, concept drift, i.e. changes of the underlying distribution \mathbb{P}_X occurs, caused e.g. by seasonal changes, changed demands, ageing of sensors, etc. Learning with drift can be dealt with in different ways. Often, data are treated via windowing techniques, and the model is continuously adapted based on the characteristics of the data in an observed time window. Thereby, many approaches deal with supervised scenarios and they aim for a small interleaved train-test error. In recent years, some approaches deal with concept drift in unsupervised settings [7, 17]. One fundamental problem, which is part of supervised learning schemes as well as unsupervised drift modelling and which will be in the focus of this publication, is the challenge of drift detection and determination of the time point when drift occurs.

According to [17] most drift detection schemes proceed in four stages: 1) collecting data, 2) building a descriptor of the data in two time windows, 3) computing a similarity based on the obtained the descriptor, 4) normalize the similarity, e.g. by considering an appropriate statistical test. This work focuses on the second and third stage of this scheme, which constitute the most crucial ones. The first stage can be solved in many problem-specific ways without a major effect on the next stages. The decision process in stage four can be bounded independently of the concrete realization: the difference of the output of stage three under the null hypothesis (no drift) and the alternative (drift) constitutes such a bound.

The aim of the present work is to determine the influence of the two major ingredients of stage 2 and 3, namely the used descriptor (stage 2) and the similarity measure applied to the descriptor (stage 3) and to evaluate their influence on the capability to detect drift and localize it in time. We will empirically show that the chosen similarity measure is of minor importance. The descriptor has an impact. In lay terms, it is more important how to estimate rather than what to estimate. This claim will be investigated from a theoretical and an empirical perspective using different estimation schemes.

Beyond this general comparison, we provide a new method to construct dataset-specific models to solve stages two and three in an efficient way: random projection-based and moment tree-based binning. This is of particular interest since dataset-agnostic similarity measures face the challenge of an inherent trade-off between decision accuracy and convergence speed.

This work is structured as follows: first (Section 2) we recall relevant work from the literature and define the problem setup – in particular, we describe different approaches to tackle the four stages (Section 3). We also provide a general argument when an estimator is capable of drift detection (see Theorem 2). In the last section (Section 5) we evaluate the metrics and estimators – showing their strengths and weaknesses – and show the suitability of our proposed approaches.

2 Problem Setup

In the usual time invariant setup of machine learning, one considers a generative process \mathbb{P}_X , i.e. a probability measure, on the data space \mathcal{X} . In this context, one views the realizations of \mathbb{P}_X -distributed, independent random variables X_1, \dots, 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, the posterior distribution. We will not distinguish between these settings and only consider distributions in general, subsuming supervised and unsupervised modeling [24].

Many processes in real-world applications are time dependent, so it is reasonable to incorporate time into our considerations. One prominent way to do so, is to consider an index set \mathcal{T} , representing time, and a collection of probability measures p_t on \mathcal{X} , indexed over \mathcal{T} , which may change over time [9]. We will usually assume $\mathcal{T} = [0, 1]$. In the following, we investigate the relationship of those p_t . Drift refers to the fact that p_t varies for different time points, i.e.

$$\exists t_0, t_1 \in \mathcal{T} : p_{t_0} \neq p_{t_1}.$$

In this context, we consider a sequence of samples $(X_1, T_1), (X_2, T_2), \dots$, with $X_i \sim p_{T_i}$ and $T_i \leq T_{i+1}$, as a stream. Notice, that we will usually use the shorthand drift instead of concept drift.

In this contribution we will mainly focus on the case of one single abrupt drift, i.e. there exist probability measures P and Q and a time point $t_0 \in \mathcal{T}$, such that

$$p_t = \begin{cases} P, & t \leq t_0 \\ Q, & t > t_0 \end{cases}.$$

In this context we can ask two questions, which are referred to as drift detection:

1. *Whether* there is drift, i.e. does $P \neq Q$ hold?
2. If so, *when* does the drift occur, i.e. what is t_0 ?

2.1 A General Scheme for Drift Detection

As most drift detection methods are applied in a streaming context, one usually considers time-dependent data samples $S(t)$, observed during a time period $W(t)$. To detect drift, one estimates the similarity of the distributions of a (presumably before drift) *reference time-interval* (or window) $W_-(t)$ and a *current time-interval* $W_+(t)$, which are obtained by splitting $W(t)$. The estimation is done using the sub-samples $S_-(t)$ and $S_+(t)$ called *windows* of $S(t)$ that correspond to $W_-(t)$ and $W_+(t)$, respectively. The way this is done varies depending on the specific algorithm. In this section we discuss some of the most prominent choices for the relevant stages 1-4 of this drift detection scheme as described in [17].

Stage 1: Acquisition of data: As stated above most approaches are based on sliding windows, however, the concrete implementation can vary. In particular, the reference window is realized in different ways: as sliding window, stationary, growing window, implicitly within a model, etc. To illustrate the idea we describe the examples of (a variant of) ADWIN and a simple version of an implicit reference window:

Example 1. ADWIN [3] uses only one sliding window $S(t_{\text{now}})$. To test for drift this window is split successively into two halves, $S_-(t; t_{\text{now}})$ and $S_+(t; t_{\text{now}})$. Then, these are compared using a suitable distance measure \hat{d} , i.e. the statistic of ADWIN is given by $\sup_t \hat{d}(S_-(t; t_{\text{now}}), S_+(t; t_{\text{now}}))$. In the original version ADWIN “preprocesses” the data by comparing the result of a fixed classification model against reference labels. However, extensions with other statistical tests are straightforward.

Example 2. A simple approach with implicit reference window consists of a reference mean $\hat{\mu}_{\text{ref}} = \mu(S_-(t))$ and a sliding window $S_+(t)$ of fixed size corresponding to $W_+(t)$. If there is no drift, the mean in the current window and the reference should be the same, i.e. $\mu(S_+(t)) \approx \hat{\mu}_{\text{ref}}$. Based in this assumption a drift detection can be performed using a t -test. Once a sample drops out of the current window $S_+(t)$ it is used to update $\hat{\mu}_{\text{ref}}$.

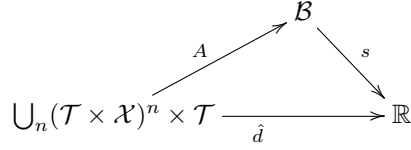


Figure 1: Two stage scheme of estimating distribution similarity from data. Distance of distributions (\hat{d}) can be estimated by building a descriptor (A) and then computing a similarity (s).

Apart from these examples, some approaches use preprocessing such as a deep latent space embedding. We do not explain those possibilities in more detail. Instead, we focus on the case of two windows only and try to evaluate the suitability of different distance measures for the task at hand.

Stage 2: Building a descriptor: Comparing two distributions directly based on a sample is usually complicated. Therefore, the process is split into two parts which are visualized in Fig. 1: First a descriptor of the distributions is built (this corresponds to A in Fig. 1), and then the similarity of the distribution is computed based on that descriptor (s in Fig. 1). Possible descriptors are grid- or tree-based binnings, neighbor-, and kernel approaches. We list some of the most popular descriptors together with suitable similarity measures in Section 3.

Stage 3: Computing similarity: As stated in the last paragraph, computing the similarity of two samples is often reduced to a comparison of descriptors which are based on those samples (s in Fig. 1). Although, several approaches for building descriptors exist, many admit the same or at least comparable similarity measures. For example, if we consider binning descriptors, it does not matter whether the bins are obtained from a grid or a tree, or if the grid or tree is adjusted to the presented data or not.

Stage 4: Normalization: As the obtained similarities typically depend on both, the method, i.e. stages 1-3, and also the concrete distribution at hand, it is necessary to normalize the result to obtain a useful scale. One of the most common ways to do this is by a relation of the similarity to the statistic of a statistical test; in this case the p -value offers a normalized scale. In the literature a large variety of approaches are considered. However, independently of the concrete normalization, the presence of drift can be observed from the output of stage 3; more formally the post hoc optimal normalization after stage 3 provides an upper bound on the quality of any concrete normalization. Therefore, we will focus on the output after stage 3 in the following.

Beyond stage 4: Ensemble and hierarchical approaches: Some authors [17] suggest to combine multiple drift detectors. They are usually arranged in an ensemble, e.g. by combining multiple p -values after stage 4 into a single one, or hierarchical, e.g. by combining a computationally inexpensive but imprecise detector with a precise but computationally expensive validation. Although, those approaches differ on a technical level, they do not from a theoretical perspective, as the suggested framework is sufficiently general.

2.2 Formal Setup and Research Question

Before we can formally specify question 1 and 2, we first have to define the sampling process:

Definition 1. Let \mathcal{X} be a data space and $\mathcal{T} \subset \mathbb{R}$. Let (p_t, P_T) be a drift process [14] on \mathcal{X} and \mathcal{T} , i.e. a distribution P_T on \mathcal{T} and a Markov kernel p_t from \mathcal{T} to \mathcal{X} . A *window* S drawn from p_t during a time interval $W \subset \mathcal{T}$ is a sample $S = \{(x_1, t_1), \dots, (x_n, t_n)\}$ drawn i.i.d. from $p_t P_T[\cdot | W]$, assuming $P_T(W) > 0$.

We use the following notation: If the choice of W is not specified, we will assume $W = \mathcal{T}$. For the sub-intervals $W_-(t) := (-\infty, t] \cap W$ and $W_+(t) := (t, \infty) \cap W$, we define the sub-windows $S_-(t) := \{(x', t') \in S \mid t' \in W_-(t)\}$ and $S_+(t)$ analogously.

Question 1: “Whether”: It was shown in [13, Theorem 2] that drift is equivalent to different sub-window distributions, i.e. it exists a $t \in W$ such that $p_{W_-(t)} \neq p_{W_+(t)}$, here p_W denotes the distribution during W . Since we do not observe the underlying distributions, but only a window S , it is reasonable to quantify this using estimating of the distance

$$\hat{d}(S_-(t), S_+(t)) := (s \circ A)(S, t)$$

which should be (significantly) larger than 0 if and only if there is drift. Here we decompose \hat{d} as described before into a descriptor A and a similarity s . Control of the uncertainty of the sampling process when detecting drift can be formalized as follows:

Definition 2. Let (p_t, P_T) be a drift process, and S denote a window drawn from it. An *estimator* (A, s) is a pair of measurable maps, one mapping windows to descriptors, i.e. $A : \cup_n (\mathcal{X} \times \mathcal{T})^n \times \mathcal{T} \rightarrow \mathcal{B}$, the other mapping descriptors to similarities, i.e. $s : \mathcal{B} \rightarrow \mathbb{R}$. We refer to \mathcal{B} as the *description space*.

An estimator is *drift detecting*, iff it raises correct alarms with a high probability in the following sense: There exists a $0 < \delta < 1/2$ and a number n such that with probability at least $1 - \delta$ over all choices of S , with $|S| > n$, it holds $s \circ A(S, t) > 0$ for some t if and only if there is drift.

An estimator is *surely drift detecting*, iff it raises correct alarms with arbitrarily high certainty, that is the above statement holds for all $0 < \delta < 1$.

Notice, that this definition is applicable for general drift, including gradual, incremental, and periodic. Furthermore, the difference between drift detection and sure drift detection only occurs in the limit of the size of S . As long as we are restrained to windows of fixed sample size, both notions are effectively the same.

Question 2: “When”: We are interested in finding the time point t_0 where the drift actually occurs. This is often estimated by the point \hat{t}_0 with largest difference of the sub-windows, i.e.

$$\hat{t}_0 = \arg \max_{t \in \mathcal{T}} \hat{d}(S_-(t), S_+(t)).$$

The precision of this estimator can be quantified by mean ratio of samples between the true drift event t_0 and its estimate \hat{t}_0 . This can be captured in the following definition:

Definition 3. Let (p_t, P_T) be a drift process with a single abrupt drift event at t_0 with $0 < P_T(W_-(t_0)) < 1$. Let S be a window drawn from p_t .

We define the *precision* as $1 - P_T([t_0, \hat{t}_0] \cup (\hat{t}_0, t_0])$.¹

¹Recall that $[a, b) = (a, b] = \emptyset$ for $a \geq b$.

We say that an estimator (A, s) is *precise*, iff for all $0 < \delta < 1$ and $\epsilon > 0$ there exists a number n , such that with at least probability $1 - \delta$ over all choices of S , with $|S| > n$, the precision is larger than $1 - \epsilon$, assuming drift was detected.

Notice, that the restriction to a single drift event in the definition of precision is necessary to avoid the ambiguity of which event \hat{t}_0 is to be compared to.

As finding the best split \hat{t}_0 requires the evaluation of multiple potential split points t , an efficient computation is important. As we have to compute the similarity at each time point, efficiency holds if the same descriptor can be used for multiple splits points, i.e. $\hat{d}(S_-(t), S_+(t)) = (s \circ A)(S, t) = s(A_0(S), t)$, where A_0 is independent of the split point. Using this idea we obtain the following definition:

Definition 4. We say that an estimator (A, s) is *c-complex*, iff $s \in \mathcal{O}(c)$ regarding computational complexity and A factorizes as $A_0 \times \text{id}_{\mathcal{T}}$, that means it holds $(s \circ A)(S, t) = s(A_0(S), t)$.

Notice, that the computational efficiency of s crucially depends on the codomain of A_0 . For example factorization also holds if we choose the set of all functions from \mathcal{T} to \mathbb{R} and s as the evaluation map.

The notion of complexity restricts how much $A(S, t)$ can be adapted to the split point t . Yet, the incorporation of the temporal information contained in S is desirable as it usually leads to better descriptors. We therefore say that an estimator is “arrival time respecting” if the descriptor uses temporal information beyond the split point:

Definition 5. An estimator (A, s) is *arrival time respecting*, iff the obtained descriptor depends on the timing within the sub-windows, i.e. it exists a split of a window $S = S_- \dot{\cup} S_+$, and permutations π_-, π_+ , such that $A(S_- \cup S_+, t) \neq A(\tilde{S}_- \cup \tilde{S}_+, t)$, where $\tilde{S}_- = \{(x_i, t_{\pi_-(i)}) \mid i = 1, \dots, n_-\}$ is the time-permuted version of $S_- = \{(x_i, t_i) \mid i = 1, \dots, n_-\}$ and analogous for \tilde{S}_+ .

3 Similarity Estimators

We are interested in popular instantiations of stages 2 and 3 and their properties. Binning can be considered as one of the simplest strategies to estimate a probability distribution. Essentially, the input space is segmented and the number of samples per bin is counted. The ratio of these samples as compared to all provides an estimate for the actual probability. Based thereon, distance measures like total variation [24], Hellinger distance [6], or Kullback-Leibler divergence [20, 5] can be computed. We will also consider the Jensen-Shannon metric which is based on the Kullback-Leibler divergence. Binning on a grid was used in the work [24], for example, to estimate the rate of change in data streams.

Since the number of required bins grows exponentially with the number of dimensions, one might consider multiple, separate binnings in low dimensional projections for high dimensional data. Typical choices are projections onto the *coordinate axis / marginals* [6] and onto the *principal components* [20]. While this strategy reduces the descriptor complexity, it is not capable of capturing drift that affects the correlation of features or of components with small variance, respectively. As this poses a problem for drift detection in the real world, we propose a new technology: *random projection* binning considers binnings along randomly chosen projection axes.

Instead of using an equally spaced grid structure, one can also consider a recursive splitting of the dataset similar to a decision tree with leaves forming the bins. Depending on the way of splitting, these are *Random Trees*, where the dimension and the split point are chosen completely randomly, or *kdt-Trees* [5], where one successively splits the dimensions along the center. As such splits often lead

Table 1: Summary of estimators (Drift Detecting: No ✗, Drift detecting (✓), Surely drift detecting ✓, Arrival Time Respecting: No ✗, Yes ✓)

Descriptor (\mathcal{B})	Metric(s)	DD	ATR		Complexity
Marginal Bin. [6]		✗	✗	$\mathcal{O}(1)$	cumulative histogram
Random Proj. Bin.	Total variation [24],	(✓)	✗	$\mathcal{O}(1)$	cumulative histogram
Random Tree (Bin.)	Hellinger [6],	✓	✗	$\mathcal{O}(1)$	cumulative histogram
$k dq$ -Tree [5] (Bin.)	Jensen-Shannon,	✓	✗	$\mathcal{O}(1)$	cumulative histogram
Moment Tree (Bin.)	D_{KL} [5]	✓	✓	$\mathcal{O}(1)$	cumulative histogram
k -NN	LDD [16], D_{KL} [19]	✓	✗	$\mathcal{O}(k)$	neighborhood graph
Kernel embedding of distribution [21]	MMD [11]	✓	✗	$\mathcal{O}(W)$	Cholesky decomposition of kernel matrix

to slow convergence, we propose to use a comparably new alternative: *Moment Trees* [15], which are designed for conditional density estimation. Here, they are trained to predict the (distribution of) time given data, i.e. $\mathbb{P}_{T|X}$. Notice, that due to the relation of a supervised problem, one can perform a parameter tuning, which is not possible for the other approaches.

Neighborhood-based approaches offer a popular and robust choice in non-parametric methods which have been widely used for various estimators, including Kullback-Leibler divergence [19]. In drift detection the *Local Drift Degree (LDD)* [16] is one method that is explicitly based on k -nearest neighbors (k -NN).

Another, non-parametric approach are kernels. *Maximum Mean Discrepancy (MMD)* [11] is a kernel-based metric, which was also applied to drift detection [21]. These methods are summarized in Table 1. We investigate their theoretical properties and experimental behavior in the following.

4 Theoretical Analysis

We will now discuss some of the properties of the approaches presented in Section 3 from a theoretical point of view. We will see that, regarding question 1 and 2, common estimators for drift detection are well suited. In the following we will always assume a drift process (p_t, P_T) on \mathcal{X}, \mathcal{T} , with $\mathcal{T} \subset \mathbb{R}$.

Linear projections: Many, in particular, simple methods use projections as a first step. However, as already discussed by [20] not every possible projection is also suitable. Indeed, most approaches from the literature are not:

Remark 1. Linear projections with respect to marginals [6] or principal components [20] are not drift detecting, independent of the further processing. This stays true if p_t is compactly supported.

Conversely, random projections are sufficient for drift detection:

Theorem 1. *Let $\mathcal{X} = \mathbb{R}^d$ and assume that p_t is compactly supported, then random projection (with $w \sim \mathcal{N}(0, \mathbb{I})$) with random bins is drift detecting.*

Proof. All proofs can be found in the appendix. □

However, we will observe that they do not perform well for a large dimensionality. We conjecture that this is a consequence of the fact that they are not arrival time respecting, and not adapted to the specific problem at hand.

Learneable models: Many popular machine learning models are also applied to estimate similarities in drift detection. Interestingly, the uniform learnability that qualifies them as valid machine learning models, also assures that the derived estimators are surely drift detecting and precise:

Theorem 2. *Let \mathcal{X} be a measurable space and let \mathcal{H} be a hypothesis class of binary classifiers on \mathcal{X} . Consider the estimators induced by*

$$\frac{1}{2} - \inf_{h \in \mathcal{H}} \mathbb{E}[\ell_w(h, (X, \mathbf{1}[T \in W_+(t)]))],$$

where ℓ_w denotes the 0-1-loss with class reweighting, i.e. $\ell_w(y', (x, y)) = 1/\mathbb{P}[Y = y]$ if $y \neq y'$ and 0 otherwise. If \mathcal{H} is PAC-learnable, then the estimator is precise. If in addition, for all binary classification tasks on \mathcal{X} there exists a $h \in \mathcal{H}$ that performs better than random, then the estimator is also surely drift detecting.

To connect this result to the existing literature, observe that (for $\mathcal{X} = \mathbb{R}^d$ and universal \mathcal{H}) the estimator is equivalent to the total variation norm. Thus, if an estimator is based on a uniform learneable model class, it is surely drift detecting and precise, but in general this requires us to retrain the model for each split point t . At this point the fact that some models do not need adaptation can increase efficiency. Indeed, for Random Trees and kdq -Trees we find the following statements:

Corollary 1. *On $\mathcal{X} = [0, 1]^d$ Random Trees and kdq -Trees with total variation norm are surely drift detecting, precise, and $\mathcal{O}(1)$ -complex with cumulative histograms as descriptors.*

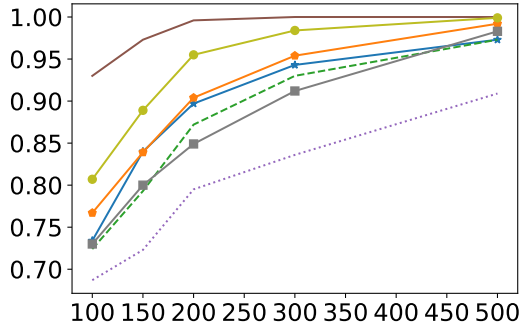
To obtain a similar result for Moment Trees, we make use of the fact that they can be used for conditional density estimation [15]: The obtained tree is suitable for all classification task for the form $\mathbf{1}[T > t]$, which is exactly what is considered by Theorem 2. We therefore conjecture that Moment Trees with total variation norm is drift detecting, precise, arrival time respecting, and $\mathcal{O}(1)$ -complex with cumulative histograms as descriptors.

5 Empirical Evaluation

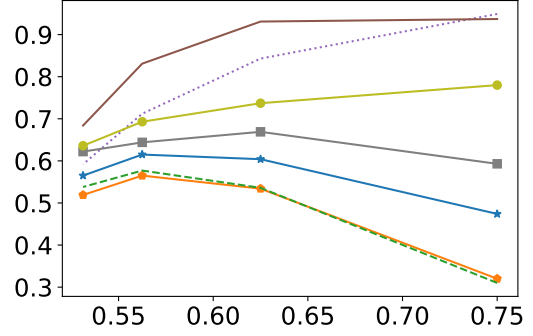
Based on the theory provided in Section 4, we can derive worst case bounds similar to standard results from classical learning theory for drift detection. Yet, we are also interested in average case bounds obtained from empirical estimations.

We apply the estimators as described in Table 1. For the binning approaches we used different numbers of bins, and equidistant and equilikely bins. In case of Random Projection, we also vary the number of projections. In case of the k -NN and tree approaches we vary the number of neighbors and trees. In case of Moment Trees we consider different degrees, ensembles of independently grown Decision Trees and Random Forests. For MMD we use the biased estimator with Gauss kernel. Notice, that due to the setup no parameter tuning can be performed during a run. In any case we consider all possible combinations according to Table 1. For arrival time respecting methods we also consider skipping the last 10% of the reference window during training.

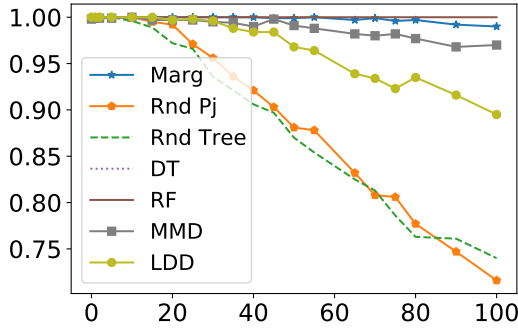
We use the following datasets: “Rotating hyperplane” (RHP) [18], “SEA” [22], “stagger” [10], “RandomRBF” (rbf) [18], “Electricity Market Prices” (elec) [12], “Forest Covertype” (cover) [4] and “Nebraska Weather” (weather) [8]. For labeled datasets, the label is integrated as an additional feature, hence real drift becomes distributional drift. To obtain a sample window with drift we sample two concepts ($S_- \sim p_{W_-} \times U([0, 1/2])$ and $S_+ \sim p_{W_+} \times U([1/2, 1])$) and concatenate them ($S = S_- \cup S_+$); we then permute these samples to obtain a counterpart without drift ($\tilde{S} = \{(x_i, t_{\pi(i)}) \mid i = 1, \dots, n\}$)



(a) Effect of window length on p_{perm} on weather dataset.



(b) Effect of displacement ($t = t_0 + \Delta$, $t_0 = 50\%$) on $p_{\text{pa}}(\Delta)$ on weather dataset.



(c) Effect of additional noise (Gaussian) dimensions on p_{perm} on electricity dataset.

Figure 2: Effect of parameters on statistical power (p_{perm}) and precision accuracy ($p_{\text{pa}}(\Delta)$). Estimators are: Marginal binning, Random Projection, Random Tree, Independent Moment Trees (DT), Random Forest Moment Trees (RF), MMD, LDD.

where $S = \{(x_i, t_i) \mid i = 1, \dots, n\}$. In case of real world datasets we obtain two different concepts (before and after drift) by randomly sampling from before and after a given time stamp (we used a two sample test to assure that the obtained batches are indeed different, while the random selection assures no drift within the sub-windows). Analysis of different split points on the same window use the same binning/tree; for other windows (including drift vs. no drift) we create a new binning/tree.

We investigate the effect of windows length, additional noise dimensions, offsets/imbalance (removing oldest 0%, 12.5%, 25% of whole window; drift is at 50%), and displacement of the split point $t = t_0 + \Delta$ (split at $t = 50\%$, 53%, 56%, 62%, 75% of the whole window; drift is at $t_0 = 50\%$). We repeat each experiment 1000 times.

Question 1: “Whether” We evaluate how well an estimator $\hat{d} = s \circ A$ detects drift. For this purpose, we estimate the probability that the estimation with drift is larger than the one without, i.e. $p_{\text{perm}} = \mathbb{P}[\hat{d}(S_-, S_+) > \hat{d}(\tilde{S}_-, \tilde{S}_+)]$, and we evaluate the probability that the estimation with and without drift can be distinguished using a threshold, i.e. $p_{\text{thre}} = \sup_b \mathbb{P}[\hat{d}(S_-, S_+) > b \geq \hat{d}(\tilde{S}_-, \tilde{S}_+)]$, where S_-, S_+ and \tilde{S}_-, \tilde{S}_+ are obtained from S and \tilde{S} , respectively, using the same split point t . Since p_{perm} is the probability that a random permutation increases the estimate, it is an upper bound

Table 2: Empirical upper bounds: p_{perm} (left), p_{thre} (right). Estimators are: Moment Tree (Random Forest), Random Projection Binning, Marginal Binning, Random Tree, MMD, and LDD.

dataset	RF		Rnd Pj		Marg		Rnd Tree		MMD		LDD	
SEA	0.56	0.53	0.51	0.49	0.59	0.55	0.62	0.56	0.54	0.49	0.53	0.50
cover	1.00	0.92	0.99	0.88	1.00	0.96	0.99	0.90	1.00	0.95	0.97	0.86
elec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	1.00	1.00
rbf	1.00	0.99	1.00	0.99	1.00	0.93	1.00	0.96	0.98	0.90	1.00	1.00
RHP	0.97	0.86	1.00	0.93	0.48	0.48	1.00	0.93	0.50	0.52	1.00	0.96
stagger	1.00	1.00	1.00	0.97	1.00	0.96	1.00	1.00	1.00	0.92	1.00	0.98
weather	0.99	0.84	0.86	0.70	0.85	0.69	0.83	0.67	0.80	0.65	0.91	0.73

for the statistical power (TP/T) of any normalization. Similarly, p_{thre} is an upper bound for the balanced accuracy $((\text{TP}/T + \text{TN}/N)/2)$ of (distribution dependent) threshold-based normalization. Unlike a comparison to 0, this procedure does not suffer from possible estimator biases.

The results for one setup (length 150, split at drift point, no offset, total variation norm and LDD (in case of k -NN), where hyper-parameters are selected to optimize p_{thre} in a previous run) are presented in Table 2. An analysis of feature importances shows that the used descriptor has the largest impact on the results, followed by the dataset. Window length and split point displacement are in medium range, the effects of the used distance measure, and the offset are marginal.

As can be seen, all methods perform about equally good. Exceptions are Random Trees and Marginal Binning, which are the only methods that are better than random on the SEA dataset (Moment Tree and LDD are also able to solve SEA for larger windows sizes), and Moment Trees (RF) which is the only method that could solve the weather dataset. To show the impact of the window length, we plot the results for different window length for the weather dataset (see Fig. 2a). As can be seen for most methods, using more samples increases the statistical power. The results on the impact of noise for the Electricity dataset are presented in Fig. 2c. Only Moment Trees can handle the noisy version.

Question 2: “When” To evaluate precision of an estimator \hat{d} we empirically evaluate the probability that the estimation at the real split point t_0 is larger than the one at the displaced split point $t_0 + \Delta$, i.e. $p_{\text{pa}}(\Delta) = \mathbb{P}[\hat{d}(S_-(t_0), S_+(t_0)) > \hat{d}(S_-(t_0 + \Delta), S_+(t_0 + \Delta))]$. We refer to this as precision accuracy. Notice, that this corresponds to an ADWIN [3] like split point search. The feature importances provides the same results as before. The results (under the same parameters as in Table 2) are shown in Table 3. We also illustrate the behavior for the weather dataset in Fig. 2b for different Δ .

As can be seen the larger the split point displacement (Δ), the higher the precision accuracy. Furthermore, except for two datasets and only with $\Delta = 3\%$, Moment Trees show the best performance. Furthermore, they tend to approach perfect precision accuracy rather quickly.

6 Conclusion

In this paper we studied the theoretical and empirical properties of several metrics that are used in drift detection. We also introduced two new metric estimators based on Random Projection Binning and Moment Trees. We found that in most cases the estimation method is more important than the used distance measure, when it comes to drift detection. Also, most datasets can be solved by

Table 3: Precision accuracy: $\Delta = 3\%$ (left) and $\Delta = 12\%$ (right). Estimators are: Moment Tree (Random Forest), Random Projection, Marginal Binning, Random Tree, MMD, and LDD.

dataset	RF		Rnd Pj		Marg		Rnd Tree		MMD		LDD	
SEA	0.42	0.58	0.42	0.33	0.52	0.52	0.42	0.38	0.50	0.48	0.50	0.53
cover	0.80	0.97	0.69	0.79	0.76	0.90	0.77	0.89	0.80	0.93	0.71	0.83
elec	1.00	1.00	0.94	1.00	0.94	1.00	0.95	1.00	0.80	0.90	0.95	1.00
rbf	0.96	1.00	0.93	0.99	0.81	0.91	0.89	0.97	0.81	0.90	0.94	1.00
RHP	0.73	0.93	0.75	0.90	0.48	0.48	0.76	0.90	0.50	0.47	0.75	0.92
stagger	0.93	1.00	0.84	0.94	0.71	0.82	0.94	0.99	0.82	0.91	0.78	0.94
weather	0.68	0.93	0.52	0.53	0.56	0.60	0.54	0.54	0.62	0.67	0.64	0.74

all methods, when it comes to drift detection. Regarding localizing the drift point, Moment Trees outperform the other methods.

References

- [1] Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(9), 2008.
- [2] A. Bifet and J. Gama. Iot data stream analytics. *Ann. des Télécomm.*, 75(9-10), 2020.
- [3] Albert Bifet and Ricard Gavaldà. 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, 2007.
- [4] Jock A. Blackard, Denis J. Dean, and Charles W. Anderson. Coverttype data set, 1998.
- [5] Tamraparni Dasu, Shankar Krishnan, Suresh Venkatasubramanian, and Ke Yi. 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*. Citeseer, 2006.
- [6] Gregory Ditzler and Robi Polikar. 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, 2011.
- [7] Gregory Ditzler, Manuel Roveri, Cesare Alippi, and Robi Polikar. Learning in nonstationary environments: A survey. *IEEE Comp. Int. Mag.*, 10(4):12–25, 2015.
- [8] R. Elwell and R. Polikar. Incremental learning of concept drift in nonstationary environments. *IEEE Transactions on Neural Networks*, 22(10):1517–1531, Oct 2011.
- [9] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37, March 2014.
- [10] Joao Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. In *Brazilian symposium on artificial intelligence*, pages 286–295. Springer, 2004.

- [11] Arthur Gretton, Karsten Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alex Smola. A kernel method for the two-sample-problem. *Advances in neural information processing systems*, 19, 2006.
- [12] Michael Harries and New South Wales. Splice-2 comparative evaluation: Electricity pricing. 1999.
- [13] Fabian Hinder, André Artelt, and Barbara Hammer. A probability theoretic approach to drifting data in continuous time domains. *CoRR*, abs/1912.01969, 2019.
- [14] Fabian Hinder, André Artelt, and Barbara Hammer. Towards non-parametric drift detection via dynamic adapting window independence drift detection (DAWIDD). In Hal Daumé III and Aarti Singh, editors, *Proceedings 37th ICML*, volume 119 of *PMLR*, pages 4249–4259, Virtual, 13–18 Jul 2020. PMLR.
- [15] Fabian Hinder, Valerie Vaquet, Johannes Brinkrolf, and Barbara Hammer. Fast Non-Parametric Conditional Density Estimation using Moment Trees. In *IEEE Computational Intelligence Magazine*. IEEE, 2021.
- [16] A. Liu, Y. Song, G. Zhang, and J. Lu. Regional concept drift detection and density synchronized drift adaptation. In *IJCAI*, 2017.
- [17] Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31(12):2346–2363, 2018.
- [18] Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdesslem. Scikit-multiflow: A multi-output streaming framework. *Journal of Machine Learning Research*, 19(72):1–5, 2018.
- [19] Fernando Pérez-Cruz. Estimation of information theoretic measures for continuous random variables. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *NIPS*, volume 21. Curran Associates, Inc., 2009.
- [20] Abdulhakim A. Qahtan, Basma Alharbi, Suojin Wang, and Xiangliang Zhang. A pca-based change detection framework for multidimensional data streams: Change detection in multidimensional data streams. In *Proceedings of the 21th ACM SIGKDD ICKDDM*, KDD '15, page 935–944, New York, NY, USA, 2015. Association for Computing Machinery.
- [21] Stephan Rabanser, Stephan Günnemann, and Zachary Lipton. Failing loudly: An empirical study of methods for detecting dataset shift. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- [22] W. Nick Street and YongSeog Kim. 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, 2001.
- [23] S. Tabassum, F. S. F. Pereira, S. Fernandes, and J. Gama. Social network analysis: An overview. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(5), 2018.
- [24] Geoffrey I. Webb, Loong Kuan Lee, François Petitjean, and Bart Goethals. Understanding concept drift. *CoRR*, abs/1704.00362, 2017.

A Proofs for Section 4 (Theoretical Analysis)

A.1 A proof of Theorem 1

Theorem 1 is actually a direct consequence of the following lemma:

Lemma 1. *Let X and X' be \mathbb{R}^d -valued, compactly supported random variables. If $\mathbb{P}_X \neq \mathbb{P}_{X'}$, then the set of projections that cannot distinguish them, i.e. $\{w \in \mathbb{S}^{d-1} \mid \mathbb{P}_{w^\top X} = \mathbb{P}_{w^\top X'}\}$, is a null-set (with respect to the Haar measure).*

Proof. Denote by $\mu_{\alpha,D} = \int \prod_i x_i^{\alpha_i} dD(x)$ the α th moment of D . Let $P = \mathbb{P}_X$ and $Q = \mathbb{P}_{X'}$. Since P and Q are both compactly supported, there exist moments of minimal degree d which differ for P and Q , i.e. $\exists \alpha \in \mathbb{N}_0^n : d = |\alpha| := \sum_i \alpha_i, \mu_{\alpha,P} \neq \mu_{\alpha,Q}$. Consider the function $p(w) = \int (w^\top x)^d dP(x) - \int (w^\top x)^d dQ(x)$. It is easy to see that $p(\lambda w) = \lambda^d p(w)$ for any $\lambda \in \mathbb{R}$, so in particular the zero sets of p form lines through the origin. Furthermore, by multiplying out and sorting, we see that p is a polynomial whose coefficients are essentially given by the difference of the moments of P and Q of total degree d . Assuming that $p(w) = 0$ for all w would imply that $p = 0$ and thus all coefficients of p are zero, which is the case if and only if $\mu_{\alpha,P} = \mu_{\alpha,Q}$ for all α ; this is a contradiction. Thus, there has to exist a w for which $p(w) \neq 0$. As the zeros of a (non-zero) polynomial always form a Lebesgue null-set and thus cannot intersect the sphere in more than a Haar null-set due to the resulting non-null-set of the corresponding rays, the statement follows. \square

Proof of Theorem 1. If has no drift, then for any projection w and any split point b it holds $\mathbb{P}[w^\top X > b \mid T \in W_-(t)] = \mathbb{P}[w^\top X > b \mid T \in W_+(t)]$. We may compare the empirical estimates of those probabilities using a statistical test and then consider the (inverse) p -value minus a constant that relates to the desired threshold.

So assume p_t has drift. By [13, Theorem 2] there exists a t , such that $p_{W_-(t)} \neq p_{W_+(t)}$; it then follows from Lemma 1 that the set of all projections for which both measures coincide is a Haar zero set. As scaling does not change this we may sample our projection vectors from a normal distribution.

Let F and G denote the cdfs of the projected $p_{W_-(t)}$ and $p_{W_+(t)}$, respectively. As $F \neq G$ there exists a b such that $F(b) \neq G(b)$; we will prove, that there exists some $\epsilon > 0$ such that $F(b') \neq G(b')$ for all $b' \in [b, b + \epsilon)$ which implies the statement. Assume this is not the case, then for every $\epsilon > 0$ there exists a point ξ with $b < \xi < b + \epsilon$ and $F(\xi) = G(\xi)$. We may use those ξ to construct a sequence x_n such that $x_n > x_{n+1} > b$, $x_n \rightarrow b$ and $F(x_n) = G(x_n)$ (start with an arbitrary ϵ to find $x_0 = \xi$, then choose x_{n+1} as the ξ obtained for $\epsilon = (x_n - b)/2$). Since F and G are cadlag, it holds that $F(b) = \lim_n F(x_n) = \lim_n G(x_n) = G(b)$, which is a contradiction. However, $[b, b + \epsilon)$ is a Lebesgue non-null-set, thus with probability larger than zero we pick a random split b such that $\mathbb{P}[w^\top X > b \mid T \in W_-(t)] \neq \mathbb{P}[w^\top X > b \mid T \in W_+(t)]$. We may now proceed as in the non-drifting case. \square

A.2 A proof of Theorem 2 and Corollary 1

Proof of Theorem 2. We first show that there exists an empirical estimator that converges (nearly) uniformly over $\mathcal{T} \times \mathcal{H}$ to $\mathbb{E}[\ell_w(h, \mathbf{1}[T \in W_+(t)])]$ as $|S| \rightarrow \infty$. Then we conclude that this implies the statement.

Step 1: Convergence Interpret \mathcal{H} as a set of functions from \mathcal{X} to $\{-1, 1\}$. We may rewrite

$$\begin{aligned}\mathbb{E}[\ell_w(h, \mathbf{1}[T \in W_+(t)])] &= \frac{1}{2}(\mathbb{P}[h(X) = 1|T \in W_-(t)] + \mathbb{P}[h(X) = -1|T \in W_+(t)]) \\ &= \frac{1}{2} \left(\mathbb{E} \left[\frac{h(X) + 1}{2} \middle| T \in W_-(t) \right] - \mathbb{E} \left[\frac{h(X) - 1}{2} \middle| T \in W_+(t) \right] \right) \\ &= \frac{1}{4} (\mathbb{E}[h(X)|T \in W_-(t)] - \mathbb{E}[h(X)|T \in W_+(t)]) + \frac{1}{2}\end{aligned}$$

and then use the canonical estimator

$$\frac{1}{|S_{\pm}(t)|} \sum_{x \in S_{\pm}(t)} h(x) \approx \mathbb{E}[h(x)|T \in W_{\pm}(t)],$$

for each of the expectations separately.

Denote by $n = |S|$, F_T the cdf of P_T , by $\hat{F}_T(t|S) = |S_-(t)|/n$ its canonical estimator, and by

$$\hat{\mathcal{T}}(\tau|S) := \left(\hat{F}_T(\cdot|S) \right)^{-1}([\tau, 1]) \setminus \left(\hat{F}_T(\cdot|S) \right)^{-1}([1 - \tau, 1])$$

In particular, $\{\hat{F}_T(t|S) \mid t \in \hat{\mathcal{T}}(\tau|S)\} \subset [\tau, 1 - \tau]$ for all $\tau \in (0, 1/2)$. We will show that there exists a sequence τ_n such that the canonical estimator converges uniformly over $\hat{\mathcal{T}}(\tau_n|S) \times \mathcal{H}$ and $\tau_n \rightarrow 0$ as $n \rightarrow \infty$.

Fix a $0 < \tau < 1/2$ and $\lambda > 1$. For $A_{\tau, \lambda} = \{F_T(t) \mid t \in \hat{\mathcal{T}}(\tau|S)\} \subset [\tau/\lambda, 1 - \tau/\lambda]$ it holds

$$\begin{aligned}\mathbb{P}(A_{\tau, \lambda}^C) &= \mathbb{P} \left(\left\{ \inf_{t \in \hat{\mathcal{T}}(\tau|S)} F_T(t) < \tau/\lambda \right\} \cup \left\{ \sup_{t \in \hat{\mathcal{T}}(\tau|S)} F_T(t) > 1 - \tau/\lambda \right\} \right) \\ &= \mathbb{P}(\{F_T(\inf \hat{\mathcal{T}}(\tau|S)) < \tau/\lambda\} \cup \{F_T(\sup \hat{\mathcal{T}}(\tau|S)) > 1 - \tau/\lambda\}) \\ &= \mathbb{P}(\{F_T(\inf \hat{\mathcal{T}}(\tau|S)) - \tau < \tau/\lambda - \tau\} \\ &\quad \cup \{F_T(\sup \hat{\mathcal{T}}(\tau|S)) - (1 - \tau) > 1 - \tau/\lambda - (1 - \tau)\}) \\ &\leq \mathbb{P}(\{F_T(\inf \hat{\mathcal{T}}(\tau|S)) - \underbrace{\hat{F}_T(\inf \hat{\mathcal{T}}(\tau|S))}_{\geq \tau} < \underbrace{\tau/\lambda - \tau}_{= -\tau(1-1/\lambda)}\} \\ &\quad \cup \{F_T(\sup \hat{\mathcal{T}}(\tau|S)) - \underbrace{\hat{F}_T(\sup \hat{\mathcal{T}}(\tau|S))}_{\leq 1-\tau} > \underbrace{1 - \tau/\lambda - (1 - \tau)}_{=\tau(1-1/\lambda)}\}) \\ &\leq \mathbb{P} \left[\max_{t \in \{\inf \hat{\mathcal{T}}(\tau|S), \sup \hat{\mathcal{T}}(\tau|S)\}} |F_T(t) - \hat{F}_T(t|S)| \geq \tau \left(1 - \frac{1}{\lambda} \right) \right] \\ &\leq \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S)} |F_T(t) - \hat{F}_T(t|S)| \geq \tau \left(1 - \frac{1}{\lambda} \right) \right] \\ &\leq \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}} |F_T(t) - \hat{F}_T(t|S)| \geq \tau \left(1 - \frac{1}{\lambda} \right) \right] \\ &\leq 2 \exp(-2n\tau^2(1 - 1/\lambda)^2),\end{aligned}$$

where the last inequality is the Dvoretzky-Kiefer-Wolfowitz inequality and the second equality follows from the fact that F_T is monotonously increasing.

Now for all $\omega \in A_{\tau, \lambda}$, for all $t \in \hat{\mathcal{T}}(\tau|S)$ it holds $\mathbb{P}[T \in W_{\pm}(t)] \geq \tau/\lambda, |S_{\pm}(t)|/n \geq \tau$. Furthermore, for all $h \in \mathcal{H}$ it holds

$$\begin{aligned}
& \left| \frac{1}{|S_{\pm}(t)|} \sum_{x \in S_{\pm}(t)} h(x) - \mathbb{E}[h(X)|T \in W_{\pm}(t)] \right| \\
& \leq \left| \left(\frac{|S_{\pm}(t)|}{n} \right)^{-1} \cdot \frac{1}{n} \sum_{i=1}^n h(x_i) \mathbb{I}_{W_{\pm}(t)}(t_i) - \mathbb{P}[T \in W_{\pm}(t)]^{-1} \cdot \mathbb{E}[h(X) \mathbb{I}_{W_{\pm}(t)}(T)] \right| \\
& \leq \left| \left(\frac{|S_{\pm}(t)|}{n} \right)^{-1} - \mathbb{P}[T \in W_{\pm}(t)]^{-1} \right| \cdot \underbrace{|\mathbb{E}[h(X) \mathbb{I}_{W_{\pm}(t)}(T)]|}_{\leq 1} \\
& \quad + \left| \frac{1}{n} \sum_{i=1}^n h(x_i) \mathbb{I}_{W_{\pm}(t)}(t_i) - \mathbb{E}[h(X) \mathbb{I}_{W_{\pm}(t)}(T)] \right| \underbrace{\left(\frac{|S_{\pm}(t)|}{n} \right)^{-1}}_{\leq 1/\tau},
\end{aligned}$$

which can be seen using the triangle and Jensen's inequality. As $x \mapsto 1/x$ and $x \mapsto 1/(1-x)$ are Lipschitz continuous with Lipschitz constant λ^2/τ^2 on $[\tau/\lambda, 1 - \tau/\lambda]$, we may upper bound the first summand by

$$\begin{aligned}
& \left| \left(\frac{|S_{-}(t)|}{n} \right)^{-1} - \mathbb{P}[T \in W_{-}(t)]^{-1} \right| \leq \lambda^2/\tau^2 \cdot \left| \frac{|S_{-}(t)|}{n} - \mathbb{P}[T \in W_{-}(t)] \right| \quad \text{and} \\
& \left| \left(\frac{|S_{+}(t)|}{n} \right)^{-1} - \mathbb{P}[T \in W_{+}(t)]^{-1} \right| \leq \lambda^2/\tau^2 \cdot \left| \frac{|S_{+}(t)|}{n} - \mathbb{P}[T \in W_{+}(t)] \right| \\
& \quad = \lambda^2/\tau^2 \cdot \left| \frac{|S_{-}(t)|}{n} - \mathbb{P}[T \in W_{-}(t)] \right|,
\end{aligned}$$

using that $|S_{+}(t)|/n = 1 - |S_{-}(t)|/n$ and $\mathbb{P}[T \in W_{+}(t)] = 1 - \mathbb{P}[T \in W_{-}(t)]$. From the Dvoretzky-Kiefer-Wolfowitz it follows

$$\begin{aligned}
& \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S), h \in \mathcal{H}} \left| \left(\frac{|S_{-}(t)|}{n} \right)^{-1} - \mathbb{P}[T \in W_{-}(t)]^{-1} \right| \right. \\
& \quad \left. + \left| \left(\frac{|S_{+}(t)|}{n} \right)^{-1} - \mathbb{P}[T \in W_{+}(t)]^{-1} \right| \geq \varepsilon, A_{\tau, \lambda} \right] \\
& \leq \mathbb{P} \left[2\lambda^2/\tau^2 \sup_{t \in \hat{\mathcal{T}}} \left| \frac{|S_{-}(t)|}{n} - \mathbb{P}[T \in W_{-}(t)] \right| \geq \varepsilon, A_{\tau, \lambda} \right] \\
& = \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}} \left| \frac{|S_{-}(t)|}{n} - \mathbb{P}[T \in W_{-}(t)] \right| \geq 2\lambda^2/\tau^2 \varepsilon, A_{\tau, \lambda} \right] \\
& \leq \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}} \left| \frac{|S_{-}(t)|}{n} - \mathbb{P}[T \in W_{-}(t)] \right| \geq 2\lambda^2/\tau^2 \varepsilon \right] \\
& \leq 2 \exp(-2n(2\varepsilon\tau^2/\lambda^2)^2)
\end{aligned}$$

Now, consider the induced hypothesis class

$$\begin{aligned}
\mathcal{H}_{-}(t) &:= \{(x, s) \mapsto h(x) \mathbb{I}_{W_{-}(t)}(s) \mid t' \leq t\}, \quad \text{and} \\
\hat{\mathcal{H}}_{-}(\tau|S) &= \mathcal{H}_{-}(\inf \hat{\mathcal{T}}(\tau|S)).
\end{aligned}$$

Notice, that for $t < t'$ we have $\mathcal{H}_-(t) \subset \mathcal{H}_-(t') \subset \mathcal{H}_-(1) =: \mathcal{H}_-$. Furthermore, for a set $D \subset \mathcal{X} \times \mathcal{T}$ with projections $D_{\mathcal{X}}, D_{\mathcal{T}}$ we have $|(\mathcal{H}_-(t))|_D| \leq |\mathcal{H}_{|D_{\mathcal{X}}}| \cdot (|D_{\mathcal{T}}| + 1)$ for all t , so in particular for all $\hat{\mathcal{H}}_-(\tau|S)$ independent of S , up to $|S|$, and τ . In particular, we can upper bound the second summand using Vapnik's inequality w.r.t. $\mathcal{H}_-(\tau|S)$ by

$$\begin{aligned}
& \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S), h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n h(x_i) \mathbb{I}_{W_-(t)}(t_i) \right. \right. \\
& \quad \left. \left. - \mathbb{E}[h(X) \mathbb{I}_{W_-(t)}(T)] \right| \mathbb{P}[T \in W_-(t)]^{-1} \geq \varepsilon, A_{\tau, \lambda} \right] \\
& \leq \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S), h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^n h(x_i) \mathbb{I}_{W_-(t)}(t_i) - \mathbb{E}[h(X) \mathbb{I}_{W_-(t)}(T)] \right| \geq \varepsilon \tau / \lambda, A_{\tau, \lambda} \right] \\
& = \mathbb{P} \left[\sup_{h' \in \hat{\mathcal{H}}_-(\tau|S)} \left| \frac{1}{n} \sum_{i=1}^n (h'(x_i, t_i) + 1) - \mathbb{E}[(h'(X, T) + 1)] \right| \geq \varepsilon \tau / \lambda, A_{\tau, \lambda} \right] \\
& \leq \mathbb{P} \left[\sup_{h' \in \hat{\mathcal{H}}_-(\tau|S)} \left| \frac{1}{n} \sum_{i=1}^n (h'(x_i, t_i) + 1) - \mathbb{E}[(h'(X, T) + 1)] \right| \geq \varepsilon \tau / \lambda \right] \\
& \leq 4\Gamma(2n) \exp(-n(\varepsilon \tau / \lambda)^2 / 72)
\end{aligned}$$

Notice, that all those statements also hold for the analogous hypothesis class $\mathcal{H}_+(t)$ (defined using $W_+(t)$) with $t > t'$ and 0 (instead of 1) which corresponds to the approximation of $\mathbb{E}[h(X)|T \in W_+(t)]$.

Combining both bounds, we obtain an upper bound for the error

$$\begin{aligned}
& \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S)} \sup_{h \in \mathcal{H}} \left| \mathbb{E}[\ell_w(h, (X, \mathbf{1}[T \in W_+(t)]))] \right. \right. \\
& \quad \left. \left. - \frac{1}{n} \sum_{i=1}^n \ell_w(h, (x_i, \mathbf{1}[t_i \in W_+(t)])) | S \right| \geq \varepsilon_1 + \varepsilon_2 \right] \\
& \leq \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau|S)} \sup_{h \in \mathcal{H}} \left| \mathbb{E}[\ell_w(h, (X, \mathbf{1}[T \in W_+(t)]))] \right. \right. \\
& \quad \left. \left. - \frac{1}{n} \sum_{i=1}^n \ell_w(h, (x_i, \mathbf{1}[t_i \in W_+(t)])) | S \right| \geq \varepsilon_1 + \varepsilon_2, A_{\tau, \lambda} \right] + \mathbb{P}(A_{\tau, \lambda}^C) \\
& \leq 2 \cdot 4\Gamma(2n) \exp(-n(\varepsilon_1 \tau / \lambda)^2 / 72) + 2 \exp(-2n(2\varepsilon_2 \tau^2 / \lambda^2)^2) \\
& \quad + 2 \exp(-2n\tau^2(1 - 1/\lambda)^2).
\end{aligned}$$

Setting $\lambda = \sqrt{\tau}$, $\varepsilon_1 = \sqrt{144} \cdot \tau$, $\varepsilon_2 = \sqrt{\tau}/2$ and assuming $0 \leq \tau \leq (3 - \sqrt{5})/2$ we obtain

$$\begin{aligned}
& \leq 2 \cdot 4\Gamma(2n) \exp(-2n\tau^3) + 2 \exp(-2n\tau^3) + 2 \exp(-2n \underbrace{\tau^2(1 - 1/\sqrt{\tau})^2}_{\geq \tau^3}) \\
& \leq 2 \cdot (4\Gamma(2n) + 2) \cdot \exp(-2n\tau^3).
\end{aligned}$$

Thus, by setting $\tau_n = (3 - \sqrt{5})/(2\sqrt[4]{n})$ we have $\tau_n \rightarrow 0$, $\varepsilon_n = \sqrt{144}\tau_n + \sqrt{\tau_n}/2 \rightarrow 0$ and

$$\begin{aligned} & \mathbb{P} \left[\sup_{t \in \hat{\mathcal{T}}(\tau_n|S)} \sup_{h \in \mathcal{H}} \left| \mathbb{E}[\ell_w(h, (X, \mathbf{1}[T \in W_+(t)])) \right] \right. \\ & \quad \left. - \frac{1}{n} \sum_{i=1}^n \ell_w(h, (x_i, \mathbf{1}[t_i \in W_+(t)]|S) \right| \geq \varepsilon_n \right] \\ & \leq 2 \cdot (4\Gamma(2n) + 2) \cdot \exp\left(-18n^{\frac{7}{4}}\right) \rightarrow 0 \end{aligned}$$

as $n \rightarrow \infty$ since $\log(\Gamma(2n)) \in \mathcal{O}(\log(n))$, due to Sauer's lemma and the fundamental theorem of learning theory.

Step 2: Application If there is no drift, then the above estimator will tend to zero. If there is drift, then there is a t_0 with $0 < F_T(t_0) < 1$ such that the distributions before and after differ. Since $\tau_n \rightarrow 0$ there exists an n such that $t_0 \in \hat{\mathcal{T}}(\tau_n|S)$ with probability at least $1 - \delta/3$ so that the estimate will converge to a value larger or equal to

$$\sup_{h \in \mathcal{H}} \left| \int h(x) dp_{W_-(t)}(x) - \int h(x) dp_{W_+(t)}(x) \right| \stackrel{!}{\geq} 0$$

and in particular eventually exceed ε_n with probability at least $1 - \delta/3$, assuming \mathcal{H} provides estimators that are better than a random guess so that $>$ holds in $!$. So we detect the drift with at least $1 - 2/3\delta$.

On the other hand, if there is only one drift point, then the difference $|p_{W_-(t)}(A) - p_{W_+(t)}(A)|$ is maximal at $t_0 = t$ for any A . It follows that the estimator is precise. \square

Proof of Corollary 1. We can consider a decision tree as a composition $h = v \circ l$ of a map $l : \mathcal{X} \rightarrow \{1, \dots, n\}$, assigning every point to the leaf it belongs to, and a map $v : \{1, \dots, n\} \rightarrow \{0, 1\}$, assigning a value to every leaf. For a fixed tree structure l , we can obtain a hypothesis class \mathcal{H}_l by composing l with every possible value map v . As \mathcal{H}_l is always finite and l is constructed independently of S , we can apply Theorem 2 to show the statement. Thus, it remains to show that \mathcal{H}_l contains a model capable of detecting the drift \mathbb{P} -a.s., but since we know that both models are consistent (consider the proof of [1, Theorem 2] for Random Trees and notice that the proof can easily be adapted for kdq -Trees) and convergence in probability implies convergence a.s. along a subsequence, which is equivalent in this case as if l' is a sub-tree of l then $\mathcal{H}_{l'} \subset \mathcal{H}_l$, this is clear. \square