Adaptive Concept Drift Detection

Anton Dries[*] Ulrich Rückert[†]

Abstract

An established method to detect concept drift in data streams is to perform statistical hypothesis testing on the multivariate data in the stream. Statistical decision theory offers rank-based statistics for this task. However, these statistics depend on a fixed set of characteristics of the underlying distribution. Thus, they work well whenever the change in the underlying distribution affects these properties measured by the statistic, but they perform not very well, if the drift influences the characteristics caught by the test statistic only to a small degree. To address this problem, we present three novel drift detection tests, whose test statistics are dynamically adapted to match the actual data at hand. The first one is based on a rank statistic on density estimates for a binary representation of the data, the second compares average margins of a linear classifier induced by the 1-norm support vector machine (SVM), and the last one is based on the average zero-one or sigmoid error rate of an SVM classifier. Experiments show that the margin- and error-based tests outperform the multivariate Wald-Wolfowitz test for concept drift detection. We also show that the tests work even if the drift is gradual in nature and that the new methods are faster than the Wald-Wolfowitz test.

1 Introduction

Learning with concept drift poses an additional difficult challenge to existing learning algorithms. Instead of treating all training examples equally, a concept drift aware system must decide to what extent some particular set of examples still represents the current concept. After all, a recent concept drift might have made the examples less relevant or even obsolete for classifier induction. This *concept drift detection* problem is often addressed by statistical methods. More formally, the problem can be framed as follows: Given a sequence of training examples, are the last n_1 examples sampled from a different distribution than the n_2 preceding ones? Depending on the answer to this question, the learning algorithm can then incorporate the examples at hand to a larger or smaller extent in the generation of a classifier. Statistical decision theory has come up with a broad range of established methods that can be used for this purpose. These methods typically compute a statistic that catches the similarity between the two example sets. The value of the statistic is then compared to the expected value under the null hypothesis that both sets are sampled from the same distribution. The resulting *p-value* can be seen as a measure of to what extent concept drift has happened. In order to be accurate, these statistical tests need to extract as much information as possible from the two samples. Sometimes this is done by building the minimum spanning tree of a complete graph that encodes the similarity between examples in the two sets [8], sometimes nearest neighbor methods are applied to compute the statistic [18], and some approaches require a complete matrix of dissimilarity measures between all examples as determined by a kernel [10].

It must be noted, though, that it is impossible to come up with a universally best test statistic. This is because for every test statistic one can construct a pair of distributions, which differ from each other to some degree, but lead to the same distribution of the test statistic. For instance, the multivariate Wald-Wolfowitz test [8] is based on the differences between the data points as measured by a metric. Thus, a concept drift, which keeps the distances constant (such as certain rotations) can not be detected by this test. The question on whether or not a particular test works well in a particular setting depends on the match of the applied test statistic with the underlying distribution. In the following we propose and evaluate three new methods, which adjust the test statistic depending on the actual data. This ensures that the test statistic captures the most important properties of the underlying distributions and adjusts itself well in a broad range of settings. The first method is based on density estimation with a binary representation of the data, the second uses a 1-norm SVM in a PAC-Bayesian framework, while the third one is based on the error rate of a linear classifier induced by a SVM. As a benchmark with high computational complexity, we use the Wald-Wolfowitz test, which is based on the minimum spanning tree of the complete similarity graph. The test has been shown to work very well in

[*] Katholieke Universiteit Leuven, Celestijnenlaan 200 A, B-3001 Leuven, Belgium, anton.dries@cs.kuleuven.be

[†] International Computer Science Institute, 1947 Center Street. Suite 600, Berkeley, CA 94704, rueckert@icsi.berkeley.edu

empirical studies [10], but it requires the computation of a new minimum spanning tree for each new example.

Another consideration is the time complexity of the tests. The kernel-based tests in [10], for instance, have at least quadratic runtime complexity with regard to training set size. This makes them unsuitable for typical data stream applications such as internet transaction monitoring, stock price prediction, or object recognition in video data, where the learning system is expected to work in an online fashion. In these settings the learning system is given new examples in short time intervals and it is asked to update its current model without spending too much time. Costly computations are therefore not possible. In the following we also investigate the trade-off between accuracy and time complexity of concept drift detection with statistical tests in the online setting. In particular, we are interested in whether simple statistics based on averages over the data points can compete with computationally more complex statistics such as rank-based measures.

The paper is organized as follows. We start with a short overview of related work in Section 2 before we present the evaluated concept drift detection methods in Section 3. These methods are then evaluated empirically in Section 4. A short conclusion is given in Section 5. The Appendix contains the proofs of the main theorems.

2 Related Work

Learning with concept drift has been the subject of many studies. We refer to the survey by Tsymbal [20] for a short overview and pointers to the relevant literature. On the theoretical side, early investigations extended results from computational learning theory to relate the strength of concept drift, the hypothesis space complexity and the expected prediction error [14, 11]. On the practical side, early approaches such as the one by Widmer *et al.* [23] often used heuristics and a sliding window to gradually adjust the generated classifier to the current concept. Later approaches more often applied statistical principles, such as the leave-one-out bound [13] to measure and rate concept drift. Ensemble-based methods adjust the weights of the base classifiers instead of modifying a classifier, see e.g. [22, 19].

The task of concept drift detection can be framed as a statistical hypothesis test with two samples and multivariate data. There is quite some work on such problems in the statistical literature. Most prominently, a study by Friedman and Rafsky [8] extended the Wald-Wolfowitz and the Smirnov tests towards the multivariate setting. Later approaches are based on nearest-neighbor analyses [18] or distances between density estimates [1]. Most recently, statistics based on maximum mean discrepancy for universal kernels have become popular [10]. There is also a range of statistical work on abrupt *change detection*, see e.g. [3, 6].

The methods for concept drift detection proposed in this paper are related to the work by Hido et al. [12]. Their *virtual classifier* approach assigns positive and negative class labels to the instances depending on which sample they stem from and then induces a classifier from these labeled examples. While this is similar to the approach we take for the SVM-based statistics, their study is focused more on drift analysis rather than drift detection and the method is based on a costly cross-validation procedure that is not practical for data stream settings. Finally, our CNF based test is somewhat related to work by Vreeken *et al.* [21], where itemset mining techniques are used for estimating the dissimilarity between two samples. The technique to identify concept drift locations by finding peaks in sequences of p-values can be found in a similar way in work by Gama *et al.* [9]. Finally, there is also research on machine learning based methods in *outlier and anomaly detection* [4, 24].

3 Concept Drift Detection

Let us frame the problem of concept drift detection and analysis more formally. We are given a continuous stream of examples $x_1, x_2, \ldots$. Each example is an m-dimensional vector in some pre-defined vector space $\mathcal{X} = \mathbb{R}^m$. At every time point p we split the examples in a set $\underline{X}$ of $\underline{n}$ recent examples and a set $\overline{X}$ containing the $\overline{n}$ examples that appeared prior to those in $\underline{X}$. We would now like to know whether or not the examples in $\overline{X}$ were generated by the same distribution as the ones in $\underline{X}$. The standard tools for drift detection are methods from statistical decision theory. These methods usually compute a statistic from the available data, which is sensitive to changes between the two sets of examples. The measured values of the statistic are then compared to the expected value under the null hypothesis that both samples are from the same distribution. The resulting p-value can be seen as a measure of the strength of the drift. A good statistic must be sensitive to data properties that are likely to change by a large margin between samples from differing distributions. This means it is not enough to look at means or variance-based measures, because distributions can differ significantly even though mean or variance remain in the same range. Since they are also sensitive to higher-order moments, rank-based measures such as the Mann-Whitney or the Wald-Wolfowitz statistics are successful in nonparametric drift detection.

Unfortunately, rank-based statistics for multivariate data often require costly computations. The Wald-Wolfowitz and the Smirnov test, for example, require the computation of the minimum spanning tree of a complete graph with $\underline{n}+\overline{n}$ vertices. In the following, we present and evaluate three different strategies that aim at drift detection based on statistics that are easier to compute. In particular, we follow the lead of [12] and re-use methods from supervised machine learning and statistical learning theory to design and analyze suitable statistics for drift detection.

3.1 A CNF Density Estimation Test The first method is based on density estimation on a binary representation of the data. We start by discretizing the continuous attributes in the data sets into a fixed set of bins. We then assign a binary feature to each of these bins. With this, each example is represented by an m'-dimensional feature vector of binary (i.e. Boolean) features. Let $\mathcal{A}$ denote the set of the m' Boolean attributes and $\mathcal{C}_l := \{A \subset \mathcal{A} || A| = l\}$ be the set of all feature-subsets of size l. Given an example x and a subset A we say that A *covers* x, if at least one feature in A is set to *true* by the example x. This is the same as demanding that the clause $a_1 \vee \ldots \vee a_k$ is satisfied for the subset $A = \{a_1, \ldots, a_k\}$. Let $A_i := \{A \in \mathcal{C}_l | A \text{ covers } x_1 \wedge \ldots \wedge A \text{ covers } x_i\}$ denote the set of subsets that cover all examples $x_1, \ldots, x_i$ observed on or before time step i. In other words, the set A_i contains all clauses that are satisfied by the examples $x_1, \ldots, x_i$.

We now proceed as follows: we split the sequence of examples in three parts. The first $\dot{n}$ examples are stored in the set $\dot{X}$, the next $\overline{n}$ examples are saved in $\overline{X}$ and the newest $\underline{n}$ examples are kept in $\underline{X}$. We would now like to find out whether the examples in $\underline{X}$ are taken from the same distribution as the ones in $\dot{X} \cup \overline{X}$. To do so, we compute the set $A_{\dot{n}} := \{A \in \mathcal{C}_l | A \text{ covers } x_1 \wedge \ldots \wedge A \text{ covers } x_{\dot{n}}\}$ of clauses, which are consistent with all examples in $\dot{X}$. Then, for each example x_i from $\overline{X}$ and $\underline{X}$, let $c_i := |\{A \in A_{\dot{n}} | A \text{ does not cover } x_i\}|$ denote the number of clauses which do not cover example x_i. If the examples in $\underline{X}$ are taken from the same distribution as the ones in $\overline{X}$ (and $\dot{X}$), the c_is should be small and not change too much, because most inconsistent clauses were already removed during the construction of $A_{\dot{n}}$. If, however, $\underline{X}$ is sampled from a different distribution as $\overline{X}$, the c_i for $x_i \in \underline{X}$ should be much larger than the ones in $\overline{X}$. To measure the significance of this difference, we apply a Mann-Whitney test on the sequence of c_is. That is, we sort the c_i by size and add up the ranks of the examples for each sample. The difference between these sums of ranks can then be used to compute a p-value. We call this method the *CNF test*, because it essentially learns a Boolean formula in conjunctive normal form (CNF) from the first part of the data and evaluates the number of clauses that are satisfied for the two samples $\overline{X}$ and $\underline{X}$. It can be computed efficiently in the online setting, because $A_{\dot{n}}$ and the c_i can be updated easily whenever a new example is observed. For the experiments in Section 4, we choose $l = 2$, so that the system collects all consistent clauses with up to two literals.

3.2 A PAC-Bayesian Margin Test The second method is based on a PAC-Bayesian analysis of a linear classifier induced on $\overline{X}$ and evaluated on $\underline{X}$. Assume we have a fixed function $f : \mathcal{X} \to [-1; 1]$. Applying such a function, we can compute the two sequences $f(\overline{x}_1), \ldots, f(\overline{x}_{\overline{n}})$ and $f(\underline{x}_1), \ldots, f(\underline{x}_{\underline{n}})$ and use any established statistical test (Mann-Whitney, etc.) on the two sequences to compute a p-value under the null hypothesis that the two sequences were generated by the same distribution. Ideally, we would like to use a function f that is sensitive to the changes between the two data samples. Unfortunately, it is not valid to select f based on the two data samples $\underline{X}$ and $\overline{X}$ and apply a standard two sample test. This is because f depends on the whole data set $\underline{X} \cup \overline{X}$ and the function values $f(\overline{x}_1), \ldots, f(\overline{x}_{\overline{n}}), f(\underline{x}_1), \ldots, f(\underline{x}_{\underline{n}})$ are thus not independent from each other. However, it is well known from statistical learning theory that the skew introduced by selecting f depending on the data set is not too large, if one chooses f to come from a rather restricted class of functions. In the following we therefore restrict ourselves to the class of linear functions $f : x \mapsto w^T x$, where w is a weight vector with $\sum_{j=1}^m |w_j| = 1$. If we choose f from this class, the following version of the PAC-Bayesian theorem can be applied to compute p-values. For ease of notation, we define $n := \overline{n} + \underline{n}$. The Kullback-Leibler divergence between two vectors is given by $D(w\|v) := \sum_i w_i \frac{\ln w_i}{\ln v_i}$

Theorem 3.1. *Let $v \in [0,1]^m$ with $\sum_{i=1}^m v_i = 1$ be arbitrary, but independent from the two samples. Let $\overline{d} := \frac{1}{\overline{n}} \sum_{i=1}^{\overline{n}} w^T \overline{x}_i$ and $\underline{d} := \frac{1}{\underline{n}} \sum_{i=1}^{\underline{n}} w^T \underline{x}_i$ and define $n' := \frac{\overline{n}\underline{n}}{\overline{n}+\underline{n}}$. Then for any $w \in [0,1]^m$ with $\sum_{i=1}^m w_i = 1$ (where w may depend on the samples), the random variable $D = \overline{d} - \underline{d}$ fulfills the following inequality:*

$$\Pr[D \geq t] \leq n' e^{-(0.5n'-1)t^2 + D(w\|v)}$$

The proof is in the Appendix. The bound can be applied as follows. First, one selects a “prior” weight vector v that assigns larger weights to attributes that are assumed to be relevant. Then, we observe the two data sets and choose a vector w that assigns large weights to attributes that distinguish well between $\overline{X}$ and $\underline{X}$. The

p-value can then be computed from the bound in the theorem. It depends on the difference between "prior" and "posterior" knowledge as encoded by $D(w\|v)$ and the empirical value of the random variable D. Since the bound is valid for any choice of w, we can also choose a w which maximizes D subject to the constraint that $\sum_{i=1}^{m} w_i = 1$. Thus, for our purposes the best w can be obtained by solving the following constrained linear program:

$$\begin{aligned} w = \operatorname*{argmax}_{w\in[0,1]^m} & \frac{1}{\overline{n}} \sum_{i=1}^{\overline{n}} w^T \overline{x}_i - \frac{1}{\underline{n}} \sum_{i=1}^{\underline{n}} w^T \underline{x}_i \\ \text{subject to } & \sum_{i=1}^{m} w_i = 1 \end{aligned}$$

Determining such a w is essentially equivalent to computing the *1-norm SVM* [25] with a linear loss function on a training set, which contains the examples in $\underline{X}$ labeled with a negative class label and the examples in $\overline{X}$ labeled with a positive label. It is easy to see that the optimal w for this optimization problem assigns full weight to the single attribute, whose average differs most between $\underline{X}$ and $\overline{X}$. For the experiments in Section 4, we therefore apply a 1-norm SVM with the hinge loss instead of the linear loss. This ensures that the weights are assigned to a larger number of attributes and that the D is based more on the instances near the decision boundary.

Due to its generality (it has to hold for all distributions and all possible w), the bound can be loose especially for data sets with many features. For concept drift detection, however, we are more interested in the change of the p-value over different samples rather than its absolute value. The experiments in section 4 indicate that the random variable D can indeed be applied to detect drift reliably. We call the described method the *margin* method, because D depends essentially on the average of margins $w^T x$ of the examples x.

Note that the original version of theorem 3.1 works only for weight vectors whose components are positive. To extend the result towards the general case where $w \in [-1, 1]^m$ (i.e. w can also contain negative weights), one can work with a $2m$-dimensional weight vector $w' := ([w_1]_+, \dots, [w_m]_+, [w_1]_-, \dots, [w_m]_-)^T$ and use a modified data matrix $X' := (X, -X)$ with twice the number of columns. Here, $[x]_+ := \max\{x, 0\}$ is defined to be zero for negative weights and $|x|$ otherwise. Likewise, $[x]_- := \max\{0, -x\}$ is zero for positive values and $|x|$ otherwise. It is easy to see that the margin of the original weight vector w on an original instance x is equal to the margin of the new weight vector on a duplicated instance: $w^T x = {w'}^T x'$.

3.3 Two Tests Based on Error Rates

The third method is also based on a SVM, but it uses the error rate rather than the average margin. We give two test statistics. The first one is based on the zero-one loss, the second one on the sigmoid loss function. In both cases we again build a training set by assigning the class label $+1$ to all instances in $\overline{X}$ and the class label -1 to all instances in $\underline{X}$. Then, we apply a traditional SVM to learn a linear classifier w from that training set. However, instead of using the margin $w^T x$ of an example x as a test statistic, we apply the *zero-one loss* $l_z(w^T x)$ or the *sigmoid loss* $l_s(w^T x)$:

$$l_z(x) := \begin{cases} 0 & \text{if } x \geq 0 \\ 1 & \text{otherwise} \end{cases}$$

$$l_s(x) := 1 - \frac{1}{1 + e^{-px}}$$

Here, $p > 0$ is a free parameter, which controls the smoothness of the sigmoid loss. The sigmoid loss can be seen as a smooth variant of the zero-one loss. Whereas the zero-one loss is non-continuous at $x = 0$, the sigmoid loss decreases smoothly from one to zero. The larger a value for p is chosen, the more l_s resembles the zero-one loss. However, since the sigmoid loss is always differentiable, it is easier to analyze theoretically and may thus give rise to better error bounds.

Using these two loss functions we can compute the loss for every example x and compare the average loss in $\overline{X}$ with the average loss in $\underline{X}$. If $\overline{X}$ and $\underline{X}$ are drawn from the same distribution, the average loss should not differ too much between the two samples. The following theorem allows the computation of a p-value for the zero-one loss.

THEOREM 3.2. *Consider the case where $n := \underline{n} = \overline{n}$. Let $\overline{e} := \frac{1}{\overline{n}} \sum_{i=1}^{n} l_z(w^T \overline{x}_i)$ and $\underline{e} := \frac{1}{\underline{n}} \sum_{i=1}^{n} l_z(w^T \underline{x}_i)$. Then for any $w \in \mathbb{R}^m$ (possibly depending on $\underline{X}$ and $\overline{X}$), the following holds for the random variable $E = \overline{e} - \underline{e}$:*

$$P(E \geq t) \leq 2 \left(\sum_{i=0}^{m+1} \binom{n}{i} \right) e^{-\frac{1}{8} t^2 n}$$

The proof is based on VC-dimension arguments. For the sigmoid loss, one can resort to Rademacher penalization techniques:

THEOREM 3.3. *Consider the case where $n := \underline{n} = \overline{n}$ and $\|x\|_\infty \leq 1$ for all examples x. Let $\overline{e} := \frac{1}{\overline{n}} \sum_{i=1}^{n} l_s(w^T \overline{x}_i)$ and $\underline{e} := \frac{1}{\underline{n}} \sum_{i=1}^{n} l_s(w^T \underline{x}_i)$. Then for any $w \in \mathbb{R}^m$ with $\|w\|_1 \leq 1$ (possibly depending on $\underline{X}$ and $\overline{X}$), the following holds for the random variable $E = \overline{e} - \underline{e}$:*

$$P(E \geq t) \leq e^{-(t - p\sqrt{\frac{m}{n}})^2 n}$$

Both proofs are in the Appendix. For the experiments in section 4, we use a traditional SVM to induce a linear classifier w that separates the examples in $\overline{X}$ well from those in $\underline{X}$. We choose $p = 100$ for the sigmoid-loss based statistic. Again, the bounds are generally too loose to yield meaningful p-values in most settings, but analyzing E appears to work well in the empirical experiments in section 4. We call the two methods the *zero-one error rate method* and the *sigmoid error rate method*, because they are based on the training error of the SVM classifier induced on the two samples.

3.4 The Wald-Wolfowitz Test Finally, as an established benchmark we make use of the multivariate version of the Wald-Wolfowitz test as described in Friedman *et al.* [8]. The algorithm proceeds in four steps. First, it computes the dissimilarity measure $d(x_i, x_j) := \|x_i - x_j\|_2$ for every pair of examples (x_i, x_j). In the second step, it constructs a complete graph, where each vertex represents an example and each edge is labeled with the dissimilarity between its two adjacent vertices. Third, it computes the minimum spanning tree (MST) for this complete graph. It is clear that this tree contains $\overline{n}+\underline{n}-1$ edges. Finally, it removes every edge between two vertices whose corresponding examples stem from different samples. This partitions the MST into a forest. The number of trees in this forest can be used as a statistic to compute a p-value. We refer the reader to [8] for details.

4 Experiments

In this section we want to evaluate the usefulness of the different approaches to concept drift detection as outlined above. In particular, we would like to investigate the following three questions.

1. How well do the described methods detect concept drift?

2. Are the methods robust against noise, i.e. do they detect drift, even though there is none?

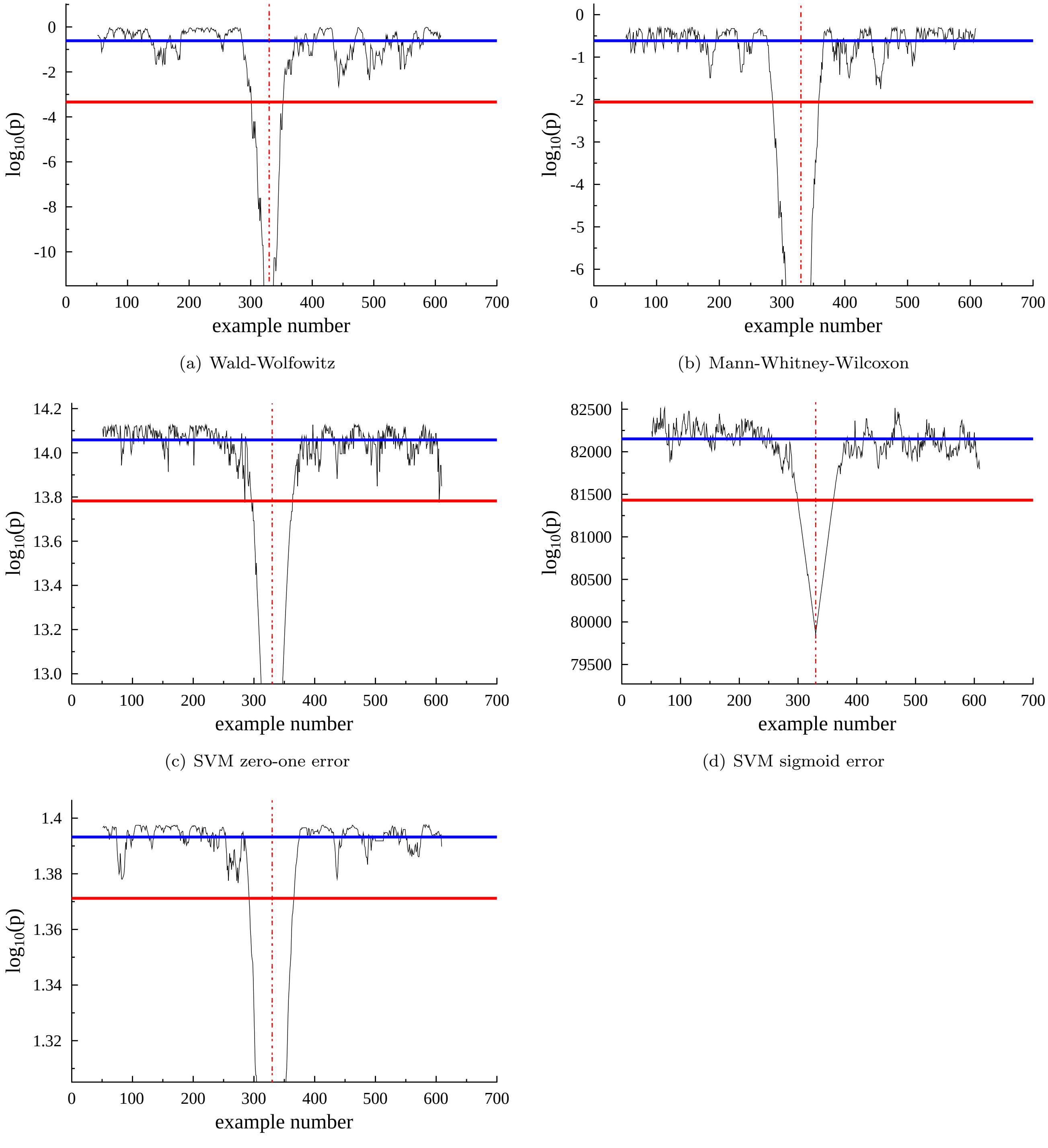
3. Do the methods still work for more gradual transitions from one distribution to the other?

We will also investigate the runtimes for the different approaches.

In order to evaluate the methods' ability to find existing concept drifts, we apply them to a set of benchmark datasets where the location of concept drift is well known. We followed the approach pioneered in [21] to generate drift detection datasets from a set of 27 UCI datasets [2] as follows. First, we order the examples in the dataset by class label so that the most common class label comes first, the second common second, etc.. Then, we shuffle the examples randomly within each class and remove the class label column. The resulting data matrix contains the examples from the most frequent class label first, followed by the examples with the second most frequent class label. Obviously, there is a concept drift in between these two parts. The strength of the drift depends on how much the two classes differ. One can easily make the drift more gradual by introducing an area of overlap that contains a random selection of both classes. For the experiments below, we use only the concept drift between the most frequent and the second most frequent class. Every class contains at least 100 examples. The implementation of the error-based method is based on LibSVM [5].

The five methods output sequences of p-values. A small p-value suggests that a concept drift is likely, whereas a large value indicates that a concept drift at that location is unlikely. The absolute value of the p-values depends on the underlying distributions, the used test statistic and the structural errors introduced by the bounding methods. So, it is difficult to compare these *absolute* values directly. For our purposes, though, it is not necessary to care about the absolute values. Since we are mainly interested in finding the location of a possible concept drift, we are looking for peaks rather than certain absolute values. To detect the peaks in the sequence of p-values, we proceed as follows. First, we compute the logarithms of the p-values. This is sensible, because the methods make use of bounds that are essentially exponential in the number of examples in the samples. It is thus way easier to detect the underlying signal on a log-scale representation. Then, at point t, we compute the average and standard deviation of all (logarithmic) p-values outside of the window from $t - \overline{n}$ to $t + \underline{n}$. This is because we want to exclude the actual area where the drift occurs as it would influence the mean and variance considerably. Finally, we compute how many standard deviations the p-value at point t is away from the average of the examples outside of the drift detection window. If it is more than s standard deviations away, the system signals the discovery of a concept drift. Figure 1 gives an example of concept drift detection by peak identification on the segment data set for $s = 5$.

The results for the experiments on all data sets with $s = 5$ are summarized in Table 1. A bullet (•) in the table indicates that concept drift was detected at approximately the right position. The left value in each column is the difference (in standard deviations) between the p-value at the correct concept drift location and the average p-value over the whole dataset. The right value is the difference (in standard deviations) of the second best location. Since there is only one valid



(a) Wald-Wolfowitz

(b) Mann-Whitney-Wilcoxon

(c) SVM zero-one error

(d) SVM sigmoid error

(e) SVM margin

Figure 1: Results for segment data set. Horizontal lines indicate the mean and 5 stddev thresholds.

dataset	**WW**			**CNF**			**Error (0/1)**			**Error (sigmoid)**			**Margin**		
anneal	• 24.3	4.1	(5)	• 31.7	3.6	(1)	• 9.1	3.6	(1)	7.4	2.8	(0)	• 24.1	3.6	(2)
balance-scale	• 39.0	3.3	(0)	• 10.1	3.3	(1)	• 28.9	6.1	(1)	• 10.2	3.4	(0)	• 24.0	3.1	(0)
breast-w	• 49.8	4.8	(2)	• > 99	5.5	(1)	• 24.6	4.3	(2)	• 8.5	2.6	(1)	• 54.5	3.8	(4)
car	• 9.4	5.9	(5)	• 75.3	6.0	(1)	• 11.1	4.5	(3)	• 6.6	3.3	(2)	• 19.0	6.4	(1)
colic	• 13.2	3.3	(0)	• -0.4	3.1	(0)	• 5.0	2.3	(0)	• 6.0	2.2	(0)	• 25.9	3.7	(0)
credit-a	• 10.9	5.5	(1)	2.1	5.3	(2)	• 11.0	3.6	(0)	• 9.1	3.0	(0)	• 51.6	5.2	(0)
credit-g	3.6	5.4	(2)	• 1.3	6.0	(2)	3.6	2.8	(0)	4.1	2.2	(0)	4.9	4.0	(0)
diabetes	• 3.5	3.5	(1)	• 0.1	6.5	(1)	• 8.6	5.5	(0)	• 4.9	3.2	(0)	• 9.0	4.9	(1)
haberman	0.8	4.4	(0)	1.0	2.8	(2)	0.8	3.7	(0)	1.1	3.3	(0)	1.4	2.4	(0)
heart-c	• 41.7	3.8	(0)	• 10.2	2.2	(0)	• 10.5	2.7	(0)	• 7.3	2.2	(0)	• 27.3	2.5	(0)
heart-h	• 16.4	2.9	(0)	• 13.7	3.9	(0)	• 8.8	2.8	(0)	• 7.4	1.8	(0)	• 24.7	3.3	(0)
heart-statlog	• 12.1	3.5	(0)	• 14.9	2.5	(0)	• 14.8	3.5	(0)	• 8.3	2.0	(0)	• 8.6	2.8	(0)
ionosphere	17.5	4.0	(0)	8.3	3.2	(0)	10.3	3.3	(0)	5.7	2.8	(0)	10.0	4.1	(0)
kr-vs-kp	• 20.1	6.4	(10)	• 11.5	6.1	(5)	• 10.7	3.6	(3)	• 9.0	3.2	(3)	• 11.0	6.5	(8)
letter	• 37.5	5.1	(5)	• > 99	4.4	(7)	• 31.3	6.1	(6)	• 13.0	3.2	(4)	• 47.8	4.7	(6)
mfeat-morph	• 40.2	3.3	(0)	• > 99	2.8	(0)	• > 99	3.3	(0)	• 47.7	2.6	(0)	• > 99	3.8	(0)
nursery	• 35.0	6.4	(23)	-0.3	7.3	(1)	• 12.4	3.8	(1)	• 7.5	2.8	(0)	• 11.8	5.3	(11)
optdigits	• 37.6	4.3	(2)	• > 99	4.7	(3)	• 10.2	3.4	(0)	• 10.6	3.3	(0)	• 79.2	5.1	(2)
page-blocks	• 30.0	6.8	(22)	• 22.4	4.6	(11)	• 31.4	6.2	(11)	• 10.9	3.4	(4)	• 39.7	6.3	(23)
pendigits	42.1	5.6	(9)	• > 99	5.0	(9)	• 31.9	5.0	(2)	• 12.7	3.0	(1)	• 81.2	6.0	(10)
segment	• 37.2	3.7	(1)	• 50.7	3.9	(1)	• 47.9	5.1	(1)	• 15.9	2.8	(1)	• 81.8	3.7	(3)
sick	• 8.8	6.0	(13)	• 7.2	6.5	(10)	• 6.1	4.5	(3)	• 4.7	3.6	(0)	• 14.2	6.0	(7)
tic-tac-toe	• 34.6	4.5	(0)	-0.3	6.4	(0)	• 14.2	3.6	(0)	• 7.2	2.9	(0)	• 13.9	3.6	(1)
vehicle	• 36.8	5.3	(0)	• 40.4	4.0	(0)	• 42.3	5.2	(0)	• 14.7	2.5	(0)	• 40.8	2.8	(0)
vote	0.1	3.5	(1)	• > 99	3.8	(0)	• 15.6	3.0	(0)	• 8.4	1.9	(0)	• 60.3	2.8	(0)
waveform-5000	• 28.5	7.2	(17)	• 13.3	5.2	(13)	• 9.1	3.8	(1)	• 10.5	3.4	(1)	• 30.7	5.5	(8)
yeast	• 7.6	4.3	(3)	2.9	3.9	(2)	6.0	4.7	(0)	• 4.2	3.0	(0)	10.7	4.6	(1)

Table 1: Results for the experiments. Columns represent correct detection, difference in standard deviations between p-value at correct point and average, largest value outside drift window (in standard deviations) and number of false detections.

concept drift location in the datasets, this is a worst-case measure of the fluctuation caused by noise. Finally, the number in brackets gives the number of incorrectly detected concept drifts. As can be seen from the table, the Wald-Wolfowitz method is much more sensitive than the other tests and could work even with larger values for the threshold s. This is followed by the CNF based statistic and the three SVM-based approaches. In our experiments, we found that the optimal threshold value for the Wald-Wolfowitz test and the CNF test is around ten.

To compare the performance of the five tests, Figure 2 gives a precision-recall plot with an incrementing threshold from one to 100. The precision and recall values are averaged over all 27 data sets. The precision and recall at the thresholds five, ten and fifteen are marked with a diamond, a circle and a triangle respectively. As can be seen from the plot, the margin-based test provides the best compromise between precision and recall, while the error-based tests work especially well for large recall settings. The Wald-Wolfowitz test, while being the most sensitive amongst the evaluated methods, is always worse than the margin- and error-based tests. This demonstrates that sensitivity of the test is not necessarily the best measure to aim for. Instead, the success of the margin- and error-based methods indicates that it might make more sense to choose a less sensitive test, but to ensure that the applied test statistic is selected to match well with the actual data.

Table 2 shows a comparison of the runtimes of the four methods on five datasets. As can be seen, Wald-Wolfowitz is by far the computationally most expensive, while the CNF-based method is fastest. The SVMs are in between, but can probably be sped up significantly by resorting to online SVMs.

To answer the second question, we shuffled the examples in all datasets randomly, so that the data does not feature any distinguishable concept drift. Running the methods on those shuffled datasets gave p-values that were very similar to the second-largest-peak values

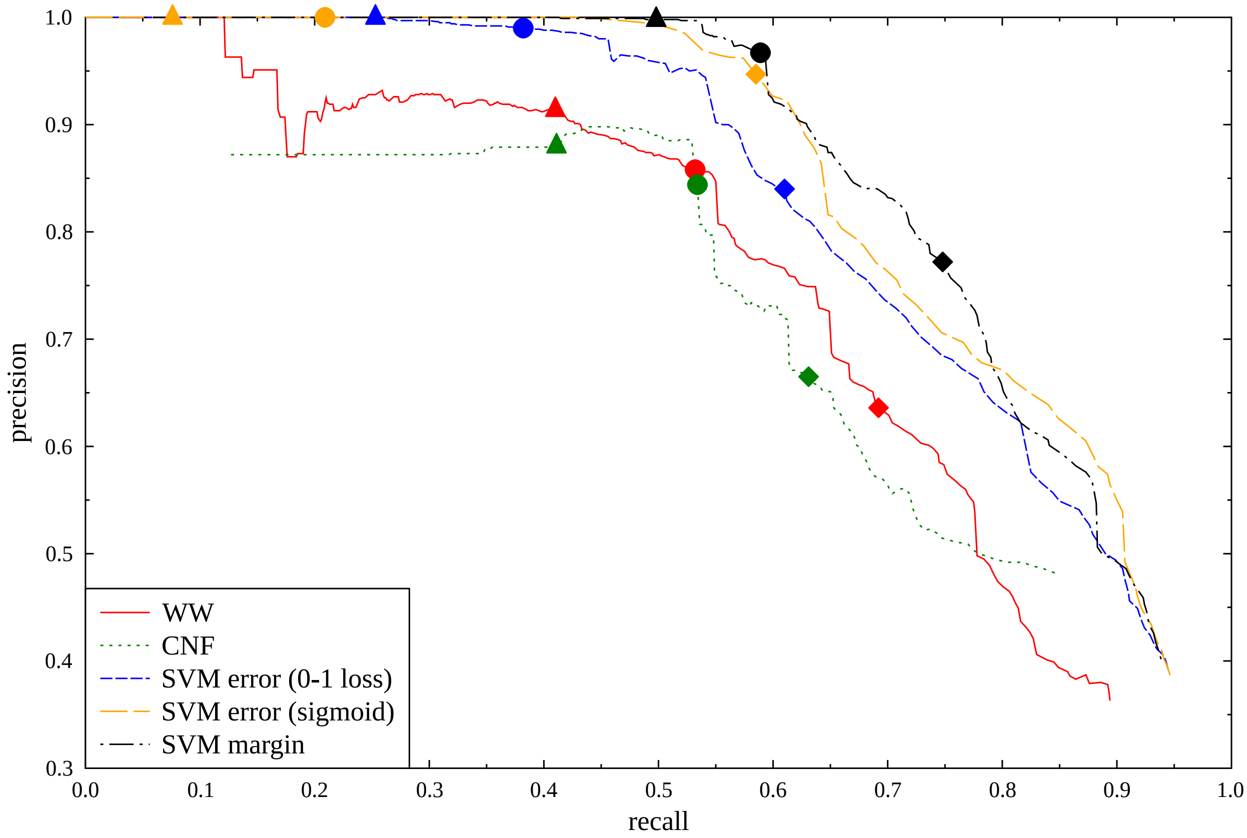


Figure 2: Precision-recall curve for the five concept drift detection methods for the thresholds between one and 100.

on the right in the columns of Table 1. This means that on most datasets settings with concept drift can be reliably distinguished from those without any drift, if the threshold is selected suitably. Finally, for the third question, we investigated wether these bounds still work when the change is more gradual. To control the speed of the drift we use the method outlined by Widmer and Kubat in [23]. We define a parameter Δx that determines the length of the interval in which the first and second class overlaps. For each position i in this interval we generate an example from the first class with probability $i/(\Delta x + 1)$ (and from the second class otherwise). This corresponds to a gradual transition from one concept to the other. The preceding experiments can then be seen as a special case where $\Delta x = 0$. For $\Delta x = 20$ there was no significant change with respect to the previous experiments. For $\Delta x = 50$ the number of standard deviations decreased and this led to reduced performance for the Wald-Wolfowitz and MMW statistics and to a lesser extend for the SVM bounds. For most data sets, however, the concept drift was still correctly detected. For $\Delta x = 100$ none of the methods could detect any concept drift. This is probably due to the fact that the drift is too slow and the changes are spread out over more than the size of the window. We expect that increasing the window size might improve this, at the cost of higher computational complexity.

5 Conclusion

In this paper we evaluated five different methods for concept drift detection in the online setting. Traditional statistical methods for drift detection such as the multivariate Wald-Wolfowitz test are often based on rank statistics that require costly computations and do not adapt to the specific properties of the underlying data distribution. In contrast, we presented three new test methods, whose statistics adapt to the data and which allow for a faster update of the statistic whenever the drift detection window moves. The first method is based on a density estimation technique on a binary representation of the data. The second method measures the

Dataset	N	m	CNF based	Marg. based	Error based	Wald-Wolf.
page-blocks	5192	11	10	114	132	414
nursery	4625	27	15	133	153	464
sick	3810	34	12	124	143	351
waveform	3390	41	11	142	159	332
kr-vs-kp	3241	41	17	119	133	291

Table 2: Comparison of runtimes (in seconds). N is the number of examples, m the number of features.

average margin of a linear classifier induced by a 1-norm SVM, while the third one is based on the average error rate of a linear classifier generated by a SVM. Empirical experiments show that these methods are better able to detect concept drifts and are not too sensitive to noise in most cases. All of them are faster than the Wald-Wolfowitz test and remain applicable if the concept drift is more gradual in nature. As an additional advantage, the SVM-based methods provide a weight vector that can be used for *concept drift analysis* in the style of [12]. In particular, the weights in the vector indicate which features were affected most by the concept drift. This information can be presented to the user or made use of for classifier modification. One of the most promising directions for future research is the application of online SVMs to further speed up the update step.

6 Appendix: Proofs

In the proofs we will make use of the following two results. The first is McDiarmid's bound, a powerful concentration inequality that can be used to bound functions of independent random variables.

Theorem 6.1. (McDiarmid, [17]) *Let $X_1, X_2, \ldots, X_n$ be independent (not necessarily identically distributed) random variables. Define a function $g : X_1 \times \ldots \times X_n \to \mathbb{R}$. If there are some nonnegative constants $c_1, \ldots, c_n$ so that for all $1 \leq i \leq n$ and for all $x_1, \ldots, x_n, x_i'$:*

$$|g(x_1, \ldots, x_n) - g(x1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_n)| \leq c_i$$

then the random variable $G := g(X_1, \ldots, X_n)$ fulfills for all $\varepsilon > 0$:

$$\mathbf{Pr}\left[G - \mathbf{E}[G] \geq \varepsilon\right] \leq \exp\Big(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\Big) \text{ and}$$
$$\mathbf{Pr}\left[\,\mathbf{E}[G] - G \geq \varepsilon\right] \leq \exp\Big(-\frac{2\varepsilon^2}{\sum_{i=1}^n c_i^2}\Big)$$

The second rather technical lemma is used in the proofs of the PAC-Bayesian theorems:

Lemma 6.1. *For $\beta > 0, K > 0$, and $R, S, x \in \mathbb{R}^m$ satisfying $R_j \geq 0, S_j \geq 0, x_j \geq 0, \sum_{i=1}^m R_j = 1$, we have that if*

$$\sum_{j=1}^n R_j e^{\beta x_j^2} \leq K$$

then

$$\sum_{j=1}^n S_j x_j \leq \sqrt{\frac{D(R\|S) + \ln K}{\beta}}$$

For a proof, see lemma 21 in [16].

6.1 Proof of Theorem 3.1 Let $D_r := |\frac{1}{\overline{n}}\sum_{i=1}^{\overline{n}}[\overline{x}_i]_r - \frac{1}{\underline{n}}\sum_{i=1}^{\underline{n}}[\underline{x}_i]_r|$ the contribution of the rth feature in the definition of D. This is a random variable depending on the two samples $\overline{X}$ and $\underline{X}$. As a first step, we prove that

$$\underset{\overline{X},\underline{X}}{\mathbf{Pr}}\left[\sum_{i=1}^m v_i e^{(0.5n'-1)D_r^2} \leq \frac{n'}{\delta}\right] \geq 1 - \delta \tag{6.1}$$

Changing one example in the first sample $\overline{X}$ changes the value of D_r by at most $\frac{2}{\overline{n}}$. Likewise, changing an example in the second sample $\underline{X}$ changes the value of D_r by at most $\frac{2}{\underline{n}}$. Since $\mathbf{E}[D_r] = 0$, McDiarmid's inequality (Theorem 6.1) ensures that

$$\underset{\overline{X},\underline{X}}{\mathbf{Pr}}\left[D_r \geq x\right] \leq 2\exp\Big[-0.5x^2n'\Big] \tag{6.2}$$

Now, we investigate the distribution of the random variable D_r. Let $f : [0,2] \to \mathbb{R}$ denote the density function of D_r so that $\mathbf{Pr}[D_r \leq x] = \int_0^x f(a)da$. Since we want to find an upper bound for $\mathbf{E}_{\overline{X},\underline{X}}\, e^{(0.5n'-1)D_r^2}$, we look for a density f_{max} that achieves the maximum of this term. More precisely, we look for the density f which maximizes $\int_0^\infty e^{(0.5n'-1)D_r^2} f(D_r)\ dD_r$, subject to the constraint (6.2) that $\int_x^\infty f(D_r)\ dD_r \leq 2e^{-0.5n'x^2}$. The maximum is achieved when $\int_x^\infty f(D_r)\ dD_r = 2e^{-0.5n'x^2}$. Taking the derivative yields that $f_{max}(D_r) = 2n'D_r e^{-0.5n'D_r^2}$. Therefore,

$$\begin{aligned}\underset{\overline{X},\underline{X}}{\mathbf{E}}\, e^{(0.5n'-1)D_r^2} &\le \int_0^\infty e^{(0.5n'-1)D_r^2} f_{max}(D_r)\ dD_r \\ &= \int_0^\infty 2n'D_r e^{(0.5n'-1)D_r^2} e^{-0.5n'D_r^2}\ dD_r \\ &= \int_0^\infty 2n'D_r e^{-D_r^2}\ dD_r \\ &= n'\end{aligned}$$

Since this upper bound is valid for all indices r, it holds also for the linear combination of the D_rs:

$$\underset{\overline{X},\underline{X}}{\mathbf{E}}\left[\sum_{i=1}^m v_i e^{(0.5n'-1)D_r^2}\right] \le n'$$

Inequality (6.1) follows from this and Markov's inequality. Applying Lemma 6.1 to (6.1) with $K = \frac{n'}{\delta}, R = R_w, S = R_v, x = (D_1, \ldots, D_m)^T, \beta = 0.5n' - 1$ yields:

$$\underset{\overline{X},\underline{X}}{\mathbf{Pr}}\left[\sum_{j=1}^m w_j D_j \ge \sqrt{\frac{D(R_w\|R_v) + \ln\frac{n'}{\delta}}{0.5n'-1}}\right] \le \delta$$

The result follows from the definition of D_j and setting

$$\delta = n'e^{-t^2(0.5n'-1)+D(R_w\|R_v)}\ .$$

6.2 Proof of Theorem 3.2 The proof is a slight modification of the well known Vapnik-Chervonenkis theorem. We start with a symmetrization argument based on Rademacher variables. Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be a sequence of n *Rademacher* random variables, which adopt the values -1 and +1 with equal probability 0.5. Define $L_i(w) := l_z(w^T\overline{x}_i) - l_z(w^T\underline{x}_i)$. Then,

$$\begin{aligned}\mathbf{Pr}[E \ge t] &= \mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n L_i(w)\right] \ge t\right] \\ &= \mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i L_i(w)\right] \ge t\right]\end{aligned}$$

This holds, because having a negative Rademacher variable is equivalent to swapping two examples between $\overline{X}$ and $\underline{X}$. Since $\overline{X}$ and $\underline{X}$ are drawn i.i.d., the expectation remains the same. Applying the union bound, we get:

$$\mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i L_i(w)\right] \ge t\right] \le$$
$$2\,\mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i l_z(w^T\overline{x}_i)\right] \ge \frac{t}{2}\right]$$

Now, we consider this probability conditional to a fixed data sample $\overline{x}_1, \ldots, \overline{x}_n$. While the supremum in the probability is over all possible w, Sauer's lemma (see, for instance, theorem 13.3 in [7]) states that linear classifiers can separate the dataset into two classes in at most $d(n,m) := \sum_{i=0}^{m+1}\binom{n}{i}$ distinct ways. This is based on the fact that the hypothesis space of hyperplanes has VC-dimension $m+1$. This means the supremum in the probability is just a maximum over $d(n,m)$ different random variables.

$$\mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i l_z(w^T\overline{x}_i)\right] \ge \frac{t}{2}\middle|\overline{x}_1, \ldots, \overline{x}_n\right] \le$$
$$d(n,m)\sup_{w\in\mathbb{R}^m}\mathbf{Pr}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i l_z(w^T\overline{x}_i) \ge \frac{t}{2}\middle|\overline{x}_1, \ldots, \overline{x}_n\right]$$

Finally, changing one σ_i changes the sum in the probability by at most $\frac{2}{n}$. Thus, McDiarmid's theorem states that

$$\mathbf{Pr}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i l_z(w^T\overline{x}_i) \ge \frac{t}{2}\middle|\overline{x}_1, \ldots, \overline{x}_n\right] \le e^{-\frac{1}{8}t^2 n}$$

Taking the expectation on both sides, we have that

$$\begin{aligned}\mathbf{Pr}[E \ge t] &\le 2\,\mathbf{Pr}\left[\sup_{w\in\mathbb{R}^m}\left[\frac{1}{n}\sum_{i=1}^n \sigma_i l_z(w^T\overline{x}_i)\right] \ge \frac{t}{2}\right] \\ &\le 2\left(\sum_{i=0}^{m+1}\binom{n}{i}\right)e^{-\frac{1}{8}t^2 n}\end{aligned}$$

6.3 Proof of Theorem 3.3 We investigate the random variable $E' = \sup_{w\in\mathbb{R}^m} E$. Changing one example in the first sample $\overline{X}$ changes the value of E' by at most $\frac{1}{\overline{n}}$. Likewise, changing an example in the second sample $\underline{X}$ changes the value of E' by at most $\frac{1}{\underline{n}}$. McDiarmid's inequality (Theorem 6.1) ensures that

$$\begin{aligned}\mathbf{Pr}\left[E' - \underset{\overline{X},\underline{X}}{\mathbf{E}}\ E' \ge s\right] &\le \exp\left[-\frac{2s^2}{\sum_{i=1}^{\overline{n}}\frac{1}{\overline{n}^2} + \sum_{i=1}^{\underline{n}}\frac{1}{\underline{n}^2}}\right] \\ &= \exp\big(-s^2 n\big) \qquad (6.3)\end{aligned}$$

Setting $s = t - \mathbf{E}[E']$ it suffices to show that $\mathbf{E}[E'] \le 2\sqrt{m/n}$ to gain the result. We prove this upper bound for $\mathbf{E}[E']$ using a symmetrization argument. Let $\sigma = (\sigma_1, \ldots, \sigma_n)$ be a sequence of n *Rademacher* random variables, which adopt the values -1 and +1 with equal probability 0.5. Then,

$$\begin{aligned}\underset{\overline{X},\underline{X}}{\mathbf{E}}[E'] &= \underset{\overline{X},\underline{X}}{\mathbf{E}}\sup_{w\in\mathbb{R}^m}\Big[\frac{1}{n}\sum_{i=1}^n l_s(w^T\overline{x}_i) - l_s(w^T\underline{x}_i)\Big] \\ &= \underset{\overline{X},\underline{X},\sigma}{\mathbf{E}}\sup_{w\in\mathbb{R}^m}\Big[\frac{1}{n}\sum_{i=1}^n \sigma_i(l_s(w^T\overline{x}_i) - l_s(w^T\underline{x}_i))\Big]\end{aligned}$$

This holds because having a negative Rademacher variable is equivalent to swapping two examples between $\overline{X}$ and $\underline{X}$. Since $\overline{X}$ and $\underline{X}$ are drawn i.i.d., the expectation remains the same. With this, we have:

$$\underset{\overline{X},\underline{X}}{\mathbf{E}}[E'] = \underset{\overline{X},\underline{X},\sigma}{\mathbf{E}} \sup_{w\in\mathbb{R}^m} \left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i(l_s(w^T\overline{x}_i) - l_s(w^T\underline{x}_i)\right]$$

$$\leq 2 \underset{\overline{X},\sigma}{\mathbf{E}} \sup_{w\in\mathbb{R}^m} \left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i l_s(w^T\overline{x}_i)\right] \tag{6.4}$$

$$\leq p \underset{\overline{X},\sigma}{\mathbf{E}} \sup_{w\in\mathbb{R}^m} \left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i w^T\overline{x}_i\right] \tag{6.5}$$

$$\leq p \underset{\overline{X},\sigma}{\mathbf{E}} \left\|\frac{1}{n}\sum_{i=1}^{n} \sigma_i \overline{x}_i\right\|_\infty \tag{6.6}$$

Here, (6.4) is a consequence of Jensen's inequality and the convexity of the supremum, (6.5) is an application of theorem 4.12 in [15] and the fact that $l_s(.)$ is Lipschitz with Lipschitz constant $\frac{p}{4}$. Finally, (6.6) is an application of Hölder's inequality and the fact that $\sup_{w\in\mathbb{R}^m}\|w\|_1 = 1$. The right hand side of (6.6) can be further bounded as follows:

$$\underset{\overline{X},\underline{X}}{\mathbf{E}}[E'] \leq p \underset{\overline{X},\sigma}{\mathbf{E}} \left\|\frac{1}{n}\sum_{i=1}^{n} \sigma_i \overline{x}_i\right\|_\infty$$

$$\leq p \underset{\overline{X},\sigma}{\mathbf{E}} \sqrt{\sum_{j=1}^{m} \left|\left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i \overline{x}_i\right]_j\right|^2} \tag{6.7}$$

$$\leq p \sqrt{\sum_{j=1}^{m} \underset{\overline{X},\sigma}{\mathbf{E}} \left|\left[\frac{1}{n}\sum_{i=1}^{n} \sigma_i \overline{x}_i\right]_j\right|^2} \tag{6.8}$$

$$\leq p \sqrt{\sum_{j=1}^{m} \frac{1}{n^2} \underset{\sigma}{\mathbf{E}} \left|\sum_{i=1}^{n} \sigma_i\right|^2} \tag{6.9}$$

$$= p \sqrt{\sum_{j=1}^{m} \frac{1}{n^2} \sum_{i,j=1}^{n} \underset{\sigma}{\mathbf{E}} \sigma_i \sigma_j}$$

$$= p \sqrt{\frac{m}{n}}$$

Inequality (6.7) follows because $\|x\|_\infty \leq \|x\|_2$ for all $x \in \mathbb{R}^m$, (6.8) is an application of Jensen's inequality and the concavity of the square root, while (6.9) holds, because $\|x_i\|_\infty \leq 1$ for all x_i. The result follows from this upper bound and by setting $s = t - \mathbf{E}[E']$ in (6.3).

Acknowledgements

We would like to thank Luc De Raedt and Siegfried Nijssen for helpful discussions. Large parts of this work were carried out while Ulrich Rückert was at Technische Universität München, Germany.

References

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

[2] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[3] Michèle Basseville and Igor V. Nikiforov. *Detection of abrupt changes: theory and application.* Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[4] P. Burge and J. Shawe-Taylor. Detecting cellular fraud using adaptive prototypes. In *Proceedings AAAI-97 Workshop on AI Approaches to Fraud Detection and Risk Management*, pages 9–13. AAAI Press, 1997.

[5] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001.

[6] F. Desobry and M. Davy. Support vector-based online detection of abrupt changes. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP '03). 2003 IEEE International Conference on*, pages IV– 872–5 vol.4. IEEE, 2003.

[7] Luc Devroye, Laszlo Györfi, and Gabor Lugosi. *A Probabilistic Theory of Pattern Recognition (Stochastic Modelling and Applied Probability).* Springer, New York, February 1996.

[8] Jerome H. Friedman and Lawrence C. Rafsky. Multivariate generalizations of the Wald-Wolfowitz and Smirnov two-sample tests. *Annals of Statistics*, 7(4):697–717, 1979.

[9] João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with drift detection. *Advances in Artificial Intelligence - SBIA 2004*, pages 286–295, 2004.

[10] Arthur Gretton, Karsten M. Borgwardt, Malte Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample-problem. In *NIPS*, 2006.

[11] D.P. Helmbold and P.M. Long. Tracking drifting concepts by minimizing disagreements. *Journal of Machine Learning*, 14(1):27–45, 1994.

[12] Shohei Hido, Tsuyoshi Idé, Hisashi Kashima, Harunobu Kubo, and Hirofumi Matsuzawa. Unsupervised change analysis using supervised learning. *Advances in Knowledge Discovery and Data Mining*, pages 148–159, 2008.

[13] Ralf Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intelligent Data Analysis*, 8(3):281–300, 2004.

[14] Anthony Kuh, Thomas Petsche, and Ronald L. Rivest. Learning time-varying concepts. In *Proceedings of the 1990 conference on Advances in neural information processing systems*, pages 183–189, San Francisco, CA, USA, 1990. Morgan Kaufmann Publishers Inc.

[15] M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes.* Springer, New York, 1991.

[16] David A. McAllester. PAC-bayesian model averaging. In *Proceedings of the twelfth annual conference on Computational learning theory.* Morgan Kaufmann Publishers, 1999.

[17] Colin McDiarmid. On the method of bounded differences. In *Surveys in Combinatorics*, pages 148–188. Cambridge University Press, 1989.

[18] Tajvidi N. Permutation tests for equality of distributions in high-dimensional settings. *Biometrika*, 89:359–374(16), June 2002.

[19] K.O. Stanley. Learning Concept Drift with a Commitee of Decision Trees. Technical Report AI-03-302, Department of Computer Sciences, University of Texas at Austin, 2003.

[20] A. Tsymbal. The Problem of Concept Drift: Definitions and Related Work. Technical Report TCD-CS-2004-15, Computer Science Department, Trinity College Dublin, 2004.

[21] Jilles Vreeken, Matthijs van Leeuwen, and Arno Siebes. Characterising the difference. In *Knowledge Discovery and Data Mining Conference*, pages 765–774, New York, NY, USA, 2007. ACM.

[22] Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Knowledge Discovery and Data Mining Conference*, pages 226–235, New York, NY, USA, 2003. ACM Press.

[23] Gerhard Widmer and Miroslav Kubat. Learning in the presence of concept drift and hidden contexts. *Journal of Machine Learning*, 23(1):69–101, 1996.

[24] Kenji Yamanishi and Jun ichi Takeuchi. Discovering outlier filtering rules from unlabeled data: combining a supervised learner with an unsupervised learner. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 389–394. ACM, 2001.

[25] Ji Zhu, Saharon Rosset, Trevor Hastie, and Robert Tibshirani. 1-norm support vector machines. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Neural Information Processing Systems.* MIT Press, 2004.