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. The statistical 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 the 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 show how uniform convergence bounds in learning theory can be adjusted for adaptive concept drift detection. In particular, 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, sigmoid or stepwise linear error rate of an SVM classifier. We compare these new approaches to the maximum mean discrepancy method, the StreamKrimp system and the multivariate Wald-Wolfowitz test. The results indicate that the new methods are able to detect concept drift reliably and that they perform favorably in a precision-recall analysis.

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. The theory on statistical hypothesis testing 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 the extent to which 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 [20], 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

*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

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 benchmarks we use the Wald-Wolfowitz test, the Maximum Mean Discrepancy (MMD) statistic and the StreamKrimp change detection system. The Wald-Wolfowitz and MMD-tests have been shown to work very well in empirical studies [10]. The StreamKrimp algorithm was recently proposed as a successful change detection system [17].

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 [24] 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 [15, 11]. On the practical side, early approaches such as the one by Widmer *et al.* [27] 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 [14] to measure and rate concept drift. Ensemble-based methods adjust the weights of the base classifiers instead of modifying a single classifier, see e.g. [26, 23].

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 [20] 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]. Most of the traditional work in this field deals with applications in signal processing, where the signals are given as a uni-variate or relatively low-dimensional time series and the noise in the signals is assumed to be Gaussian. A typical example of a classical change detection method is CUSUM [21, 3]. Recent research has addressed more general change detection settings with high-dimensional data and non-Gaussian noise. For instance, Van Leeuwen and Siebes' StreamKrimp [17] computes code tables, which can be compared to detect change in the underlying distribution. This also allows for change detection with nominal data. Kifer *et al.* present an adaptive method for concept drift detection with uni-variate distributions [13]. To get tight bounds, they optimize the test statistic over intervals on the real line rather than function classes. Finally, Song *et al.* present an approach based on kernel density estimators [22], which uses the EM algorithm to identify a suitable kernel width and makes use of bootstrapping to estimate the variance of the null hypothesis.

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.* [25], 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, 28]. The difference to the concept drift detection setting is that outlier detection is concerned with finding single exceptional events in a data series, whereas concept drift detection deals with identifying a (possibly gradual) shift in the underlying data-generating distribution.

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 hypothesis testing. 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 taken 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 tests, 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 new drift detection strategies, which are 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. We use the Maximum Mean Discrepancy (MMD) method [10], StreamKrimp [17] and the Wald-Wolfowitz test as benchmark methods.

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 $l = 3$, so that the system collects all consistent clauses with up to three literals.

3.2 Learning Theory Based Tests

Another approach to hypothesis tests for concept drift detection is to re-use results from statistical learning theory. Such tests are based on a simple observation. 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 fact one can re-use any of the many uniform convergence bounds from statistical learning theory to derive new statistical tests for concept drift detection.

This can be seen as follows. Let us first consider the standard classification or regression setting: a user draws a sample of n labeled data examples $(x_1, y_1), \ldots, (x_n, y_n)$ from a fixed, but unknown distribution D. Here, the x_i are instances taken from some instance space $\mathcal{X}$, and the target labels $y_i \in \mathbb{R}$ assign a numerical value to each instance. In classification settings, we usually have $y_i \in \{-1, 1\}$, in regression one generally chooses $y_i \in \mathbb{R}$. The user would like to find an estimator $c : \mathcal{X} \to \{-1, 1\}$, which minimizes the true loss $\varepsilon_c := \mathbf{E}_{(x,y)\sim D}[l(y, c(x))]$, where $l : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is some predefined loss function. In classification, for example, one often chooses the zero-one loss $l_z(y, y') = \mathbf{I}[y = y']$. Since the distribution D is unknown, it is impossible to find the best possible estimator. Hence, many learning algorithms search for estimators $c \in \mathcal{C}$ that in some form optimize the empirical loss $\hat{\varepsilon}_c := \frac{1}{n} \sum_{i=1}^{n} l(y_i, c(x_i))$ rather than the (unknown) true loss. An important part of statistical learning theory deals with the question of how much the true and training loss of the learned estimator differ in the worst case. Most theoretical results on this question have the form of a uniform convergence bound:

$$\Pr\left[\sup_{c \in \mathcal{C}} |\varepsilon_c - \hat{\varepsilon}_c| \leq S(\mathcal{C}, n, \delta)\right] \geq 1 - \delta \quad (1)$$

This inequality basically states that with high probability (namely, larger than $1 - \delta$), all estimators in a fixed function class $\mathcal{C}$ lead to empirical losses that differ from the true losses by at most $S(\mathcal{C}, n, \delta)$. Different results in statistical learning theory give dif-

ferent values for $S(\mathcal{C}, n, \delta)$, depending on the bounding method and the particular characteristics of the learning algorithm and the setting at hand.

The results relating empirical and true loss in the standard learning setting are directly relevant for the hypothesis testing problem in concept drift detection. As explained above, we wish to find an upper bound for the probability that the empirical measures of some function on the two samples differ too much under the null hypothesis that both samples are drawn from the same unknown distribution. The main difference to the learning setting is that there are two samples rather than only one and that one is interested in the difference between the empirical quantities on the two samples rather than the difference between empirical and true value. Fortunately, it is easy to bridge the gap between the two settings and thus to re-use results in the learning setting for the hypothesis tests in concept drift detection. We start with a typical learning bound as formalized in equation (1). As a first step, define $S^{-1}(\mathcal{C}, n, t) := \sup\{\delta | S(\mathcal{C}, n, \delta) = t\}$ and rewrite equation (1) as follows:

$$\Pr\left[\sup_{c \in \mathcal{C}} |\varepsilon_c - \hat{\varepsilon}_c| \geq t\right] \leq S^{-1}(\mathcal{C}, n, t) \tag{2}$$

By selecting a suitable loss function and function class $\mathcal{F}$ one can apply this inequality to bound the maximal difference between empirical and expected value of a function for sample $\overline{X}$ in the concept drift detection setting:

$$\Pr\left[\sup_{f \in \mathcal{F}} \left| \mathbf{E}[f(\overline{x})] - \frac{1}{n} \sum_{i=1}^{n} f(\overline{x}_i) \right| \geq t\right] \leq S^{-1}(\mathcal{F}, n, t)$$

Since the same bound also applies to the second sample $\underline{X}$, we can take the union bound over the two inequalities and yield:

$$\Pr\left[\sup_{f \in \mathcal{F}} \left| \frac{1}{\overline{n}} \sum_{i=1}^{\overline{n}} f(\overline{x}_i) - \frac{1}{\underline{n}} \sum_{i=1}^{\underline{n}} f(\underline{x}_i) \right| \geq t\right] \leq S^{-1}(\mathcal{F}, \overline{n}, \frac{t}{2}) + S^{-1}(\mathcal{F}, \underline{n}, \frac{t}{2})$$

This inequality can then be used to obtain p-values for the hypothesis test in concept drift detection.

While this recipe provides a comfortable way to transform uniform convergence bounds in the learning setting into bounds for multivariate hypothesis tests, the resulting bounds are often unnecessarily loose. In the following paragraphs, we describe tighter bounds that are derived from first principle rather than being adapted only in hitherto.

3.3 A PAC-Bayesian Margin Test

The first such bound is based on a PAC-Bayesian analysis of a linear classifier induced on $\overline{X}$ and evaluated on $\underline{X}$. We 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 solv-

ing the following constrained linear program:

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

Determining such a w is essentially equivalent to computing the *1-norm SVM* [29] 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 $w' := ([w_1]_+, \ldots, [w_m]_+, [w_1]_-, \ldots, [w_m]_-)^T$ 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: $w^T x = {w'}^T x'$.

3.4 Three 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 three test statistics. The first one is based on the zero-one loss, the second one on the sigmoid loss function, and the third uses a stepwise linear loss. In all three 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* $l_z(w^T x)$, the *sigmoid loss* $l_s(w^T x)$ and the *stepwise linear* loss $l_l(w^T x)$:

$$l_z(x) := \begin{cases} 0 & \text{if } x \geq 0 \\ 1 & \text{otherwise} \end{cases}$$
$$l_s(x) := 1 - \frac{1}{1 + e^{-px}}$$
$$l_l(x) := \begin{cases} 0 & \text{if } x \geq \frac{2}{p} \\ \frac{1}{2} - \frac{p}{4}x & \text{if } |x| < \frac{2}{p} \\ 1 & \text{if } x \leq -\frac{2}{p} \end{cases}$$

Here, $p > 0$ is a free parameter, which controls the smoothness of the sigmoid and stepwise linear losses. 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. The stepwise linear loss is essentially a linear approximation to the sigmoid loss and allows for even tighter bounds.

Using these three 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 improve the bound by using Rademacher randomization:

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 $E = \overline{e} - \underline{e}$:*

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

Finally, the stepwise linear loss allows for an even tighter upper bound for cases where the number of features is comparably large:

Theorem 3.4 *Consider the case where $n := \underline{n} = \overline{n}$. Let $\overline{e} := \frac{1}{\overline{n}} \sum_{i=1}^{n} l_l(w^T \overline{x}_i)$ and $\underline{e} := \frac{1}{\underline{n}} \sum_{i=1}^{n} l_l(w^T \underline{x}_i)$. Then for any $w \in \mathbb{R}^m$ (possibly depending on $\underline{X}$ and $\overline{X}$), the following holds for the random variable $E = \overline{e} - \underline{e}$:*

$$P(E \geq t) \leq 4me^{-\frac{1}{2} t^2 n}$$

All 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 three methods the *zero-one error rate method*, the *sigmoid error rate method* and the *stepwise linear error rate method*, because they are based on the training error of the SVM classifier induced on the two samples.

3.5 Maximum Mean Discrepancy

Recently, Gretton *et al.* proposed the Maximum Mean Discrepancy (MMD) test [10], a new multivariate hypothesis test, which is based on kernels instead of distances. This enables the test to work with non-numerical data (e.g. graphs) and to avoid the time consuming computation of spanning trees. The test is similar to the methods proposed above as it is also based on the mean values of functions of the data vectors. Instead of dealing with the functions directly, however, it projects the data into a high-dimensional feature space before computing the mean values. Using some well-known results on reproducing kernel Hilbert spaces it is possible to formulate the problem of finding the most discriminative function only implicitly, without the need to compute the expensive projection explicitly. More precisely, the MMD test is based on the following statistic:

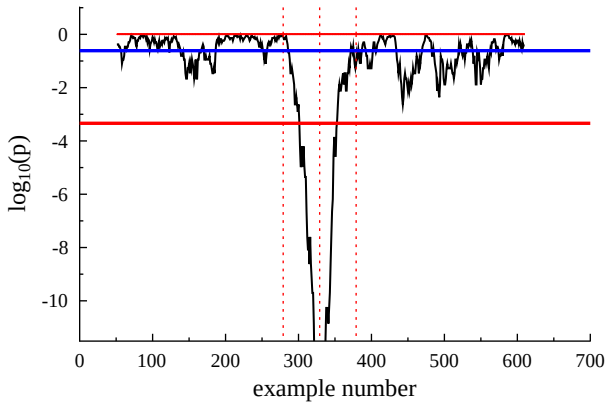
(3)

$$\mathrm{MMD}[\mathcal{F}, \overline{X}, \underline{X}] := \sup_{f \in \mathcal{F}} \left(\frac{1}{\overline{n}} \sum_{i=1}^{\overline{n}} f(\overline{x}_i) - \frac{1}{\underline{n}} \sum_{i=1}^{\underline{n}} f(\underline{x}_i) \right)$$

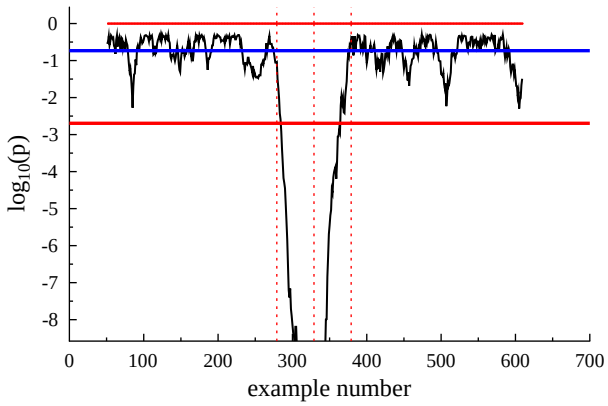
Here, $\mathcal{F}$ is some class of functions $f : \mathcal{X} \to \mathbb{R}$, which are supposed to extract the relevant information from the instances. If one sets $\mathcal{F}$ to the unit ball in a reproducing kernel Hilbert space with universal kernel $k(\mathcal{X}, \mathcal{X}) \to \mathbb{R}$, this statistic can be rewritten as the square root of a simple sum over the kernel values:

$$\mathrm{MMD}[\mathcal{F}, \overline{X}, \underline{X}] = \left[\frac{1}{\overline{n}^2} \sum_{i,j=1}^{\overline{n}} k(\overline{x}_i, \overline{x}_j) - \frac{1}{\overline{n}\underline{n}} \sum_{i,j=1}^{\overline{n},\underline{n}} k(\overline{x}_i, \underline{x}_j) + \frac{1}{\underline{n}^2} \sum_{i,j=1}^{\underline{n}} k(\underline{x}_i, \underline{x}_j) \right]^{\frac{1}{2}}$$

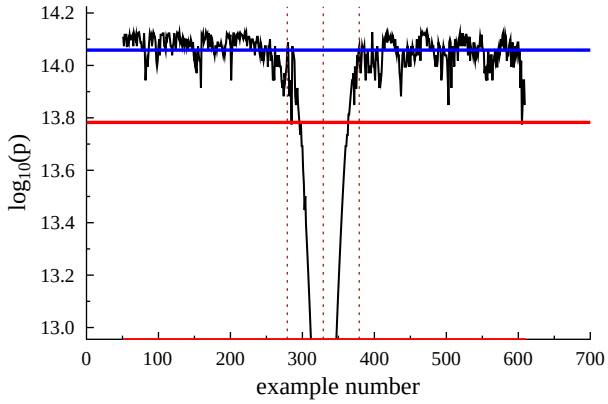
With this trick, the MMD test avoids the potentially very expensive computation of the supremum in (3). As described in [10], the p-values for the MMD statistic can be obtained in various ways. For the experiments in section 4 we chose the uniform convergence bound given by Lemma 6 in [10].



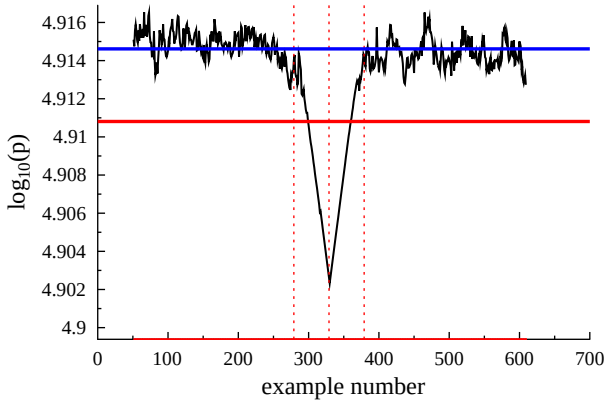
(a) Wald-Wolfowitz



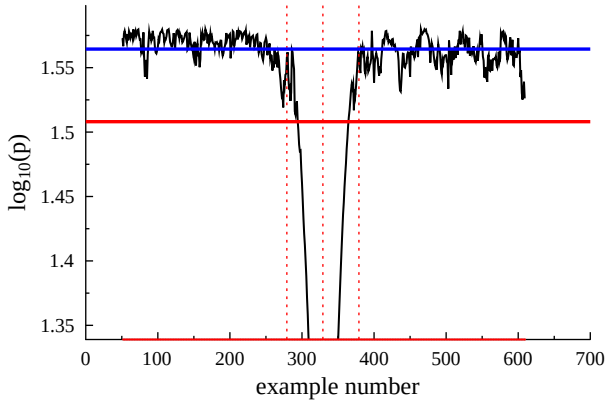
(b) 3-CNF



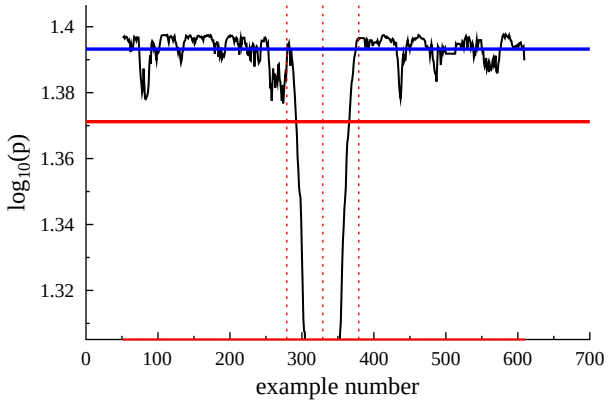
(c) SVM (0/1)



(d) SVM Sigmoid



(e) SVM stepwise linear



(f) Margin

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

3.6 StreamKrimp

Another recent approach to change detection in data streams is StreamKrimp [17]. StreamKrimp deals with streams of transactions, that is, streams where each data unit is a subset of some pre-defined set of possible items. This discrete formalization allows StreamKrimp to handle nominal attributes naturally. StreamKrimp works as follows: It reads the input stream sequentially and computes and updates a code table, which encodes the data-generating distribution in a space-efficient manner. The code table is based on the minimum description length principle (MDL). It assigns an MDL-optimal code word to each frequently occurring itemset in the stream. In this context, "MDL-optimal" means that the description length of the code table plus the length of the code representation of the data stream are minimal. Each new block of data is tested whether it fits well with the distribution encoded in the current code table. If a new block is sufficiently different, the system signals a change in the data distribution. A detailed description of the algorithm can be found in [17]. For the experiments in Section 4, we used the author's implementation of StreamKrimp with a range of different parameter settings.

3.7 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 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. Are the methods detecting concept drift reliably? In particular, how well do they detect existing concept drifts? Do they report false positives, that is, do they signal drifts on locations, where none are present?

2. How do the methods compare to each other with regard to precision and recall?

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

We address these questions in the following paragraphs.

4.1 Concept Drift Detection Reliability

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 [25] 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].

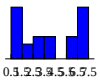
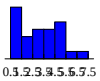
	WW	3-CNF	SVM 0	SVM S	SVM L	Margin	MMD
anneal	0.673	0.793	0.685	0.739	0.720	0.737	**0.802**
balance-scale	0.836	0.871	0.886	0.906	0.899	0.909	**0.912**
breast-w	0.829	0.814	0.811	0.827	0.821	0.886	**0.906**
car	0.445	**0.874**	0.531	0.562	0.555	0.613	0.597
colic	0.583	0.402	0.515	0.573	0.513	0.834	**0.919**
credit-a	0.714	0.654	0.831	**0.872**	0.861	0.723	0.810
credit-g	0.158	0.163	0.483	**0.588**	0.567	0.247	0.468
diabetes	0.233	0.029	0.454	0.492	0.480	**0.513**	0.394
haberman	0.021	**0.460**	0.178	0.147	0.154	0.108	0.155
heart-c	0.856	0.892	0.711	0.755	0.717	**0.914**	0.912
heart-h	**0.845**	0.755	0.770	0.814	0.780	0.808	0.798
heart-statlog	0.627	0.873	**0.940**	0.936	0.929	0.843	0.856
ionosphere	0.556	0.714	**0.852**	0.851	0.847	0.744	0.849
kr-vs-kp	0.499	**0.809**	0.611	0.663	0.650	0.409	0.541
letter	0.705	**0.861**	0.760	0.782	0.770	0.790	0.777
mfeat-morph	0.879	0.957	0.979	0.981	0.979	**0.998**	0.944
nursery	0.448	0.000[a]	0.478	0.492	0.490	0.393	**0.509**
optdigits	0.658	0.839	0.531	0.676	0.611	**0.893**	0.885
page-blocks	0.543	0.801	0.839	**0.866**	0.862	0.795	0.745
pendigits	0.680	0.828	0.806	0.857	0.847	**0.862**	0.847
segment	0.797	**0.948**	0.912	0.947	0.945	0.858	0.894
sick	0.345	**0.622**	0.339	0.402	0.390	0.456	0.459
tic-tac-toe	0.542	**0.900**	0.702	0.706	0.710	0.511	0.437
vehicle	0.940	0.809	0.961	**0.979**	0.977	0.909	0.919
vote	0.338	0.814	0.764	0.782	0.770	0.825	**0.889**
waveform	0.539	0.649	0.504	0.604	0.584	0.589	**0.650**
yeast	0.440	0.297	0.358	0.496	0.449	**0.578**	0.494
Distribution of Positions							

Table 1: Comparison of AUC-PR values for different datasets.

[a]The 0.000 value is due to the relative simplicity of the 3-CNF learner. The concept space of this learner is not powerful enough to learn a stable theory for the initial concept in the data stream. This leads to an empty theory and a steady p-value of 1. Because of this, there is no detection possible after this point (even for a threshold of 0), and this results in a recall of zero for all thresholds. To avoid this problem, a higher value of k can be chosen. Our experiments show that a 6-CNF learner, for example, scores 0.665, which is significantly higher than the other methods. However, increasing the value of k also has a significant impact on processing time and memory consumption.

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. Thus, it is difficult to compare these *absolute* values directly. For our purposes, though, it is not necessary to deal with 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[1]. 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 4 at the end of the article. 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 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. Since StreamKrimp does not output any p-values, we only state, whether the concept drift was detected correctly and the number of wrong drift detections. We obtained the best results by using closed itemsets, blocksize 20, minimum support 50 and a minimal code table difference (CTD) of 0.05. Unfortunately, three datasets caused StreamKrimp to crash, most likely due to the high-dimensionality of those datasets. As can be seen from the table, the Wald-Wolfowitz and MMD methods are much more sensitive than the other tests and could work with even larger values for the threshold s. While Wald-Wolfowitz still fails to find the correct drift location on three data sets, MMD appears to be too sensitive and reports false drifts on fourteen datasets. We investigate the trade-off between precision and recall further in the next section.

We also investigated whether the methods are too sensitive, that is, whether they find concept drift on positions where there is none. To do so, we shuffled the examples in all datasets randomly, so that the data did 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 on the right in the columns of Table 4. This means that on most datasets settings with concept drift can be reliably distinguished from those without any drift, even with a fixed detection threshold.

4.2 Precision-Recall Analysis

So far, we used a fixed threshold for all detection methods on all datasets. Of course, this is unnecessarily strict. In practice one can (and should) adjust the detection threshold to a value that ensures a good trade-off between precision and recall for the application at hand. Precision in this setting is calculated as the number of points in the drift interval that exceed the threshold, divided by the total number of points in the entire dataset that exceed the threshold. Recall is calculated by counting the number of points in the drift interval that exceed the threshold divided by some normalization constant. In our case we select the normalization constant to be the size of the drift window, which means that a recall of 100% is reached when all the points in the drift interval exceed the threshold. In Figure 2 we give a precision-recall plot for the segment data set with an increasing threshold from 0 to 100.

To compare the different methods with regard to

[1] Of course, one can also use more sophisticated methods for change detection in the (univariate) sequence of p-values. However, the main advantage of methods such as the one by Kifer *et al.* [13] is to detect statistically significant changes. Unfortunately, the statistical justification for those tests does not apply in our case, because the p-values are not distributed independently.

	MMD	Margin	SVM Linear	3-CNF	SVM 0/1	WW
1 **SVM Sigmoid**	15/12	14/13	25/2	15/12	24/3	23/4
2 **MMD**		16/11	16/10	17/10	15/12	24/3
3 **Margin**			16/11	16/11	16/11	22/5
4 **SVM Linear**				15/12	22/4	22/5
5 **3-CNF**					14/13	19/8
6 **SVM 0/1**						19/8
7 **WW**						

Table 2: Pairwise win/loss comparison of the methods. Each entry denotes the number of datasets on which the method in the row had a better/worse AUC-PR than the method in the column.

precision and recall, we computed the area under the precision-recall curve (AUC) for each method on each dataset. The results are shown in table 1. The best AUC for each data set is printed in bold. We also computed a ranking of the methods according to the AUC for each dataset. The histograms in the table give the distributions of the methods' ranks in those rankings. Additionally, Table 2 states for each pair of methods, on how many datasets the first method was better/worse than the second method.

The tables shows that there is no clear winner. All in all, the SVM with sigmoid loss performed better than MMD and the margin-based method. The SVM with stepwise linear loss has a consistently lower AUC than the SVM with sigmoid loss. However, the method's performances differ considerably on some datasets. For instance, the MMD method excels especially on the colic data set and the margin-based method outperforms the other methods by a large margin on the yeast data set. It seems that the performance of a method depends on the match between the method's bias and the data-generating distribution. This is similar to the setting in supervised learning, where the success of a learning bias can vary considerably between different data sets. The 3-CNF method performs well on data sets with a large first concept, because it is the only method that uses all data in the stream. For example, CNF ranks first on five of the ten largest datasets (and is a close second on two other).

4.3 Gradual Concept Drift

Many practical applications feature concept drifts that are gradual rather than abrupt. To investigate, whether the presented methods can also detect such drifts, we prepared a range of experiments with increasingly more gradual concept drift. To do so, 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 $\Delta x = 0$. We ran experiments with Δx increasing from 0 to 150 in increments of 10 to investigate the effect of the speed of drift on the performance of the different methods. The datasets 'heart-statlog' and 'ionosphere' are omitted in this analysis, because they are too small to have non-overlapping areas of sufficient size.

In Figure 3 we plot the AUC-PR for increasing values of Δx on the pendigits data set. The figure shows that the AUC decreases approximately linear for all methods with increasing Δx. All methods are affected to the same degree and no method gets significantly better or worse than its competitors. This also holds for most of the other data sets. In Figure 4 we plot the average rank of each method over all data sets for increasingly gradual drifts. The figure shows that no method gets significantly better or worse for varying degrees of overlap. This indicates that the performance of a method is not dependent on the abruptness of the concept drift. Methods that

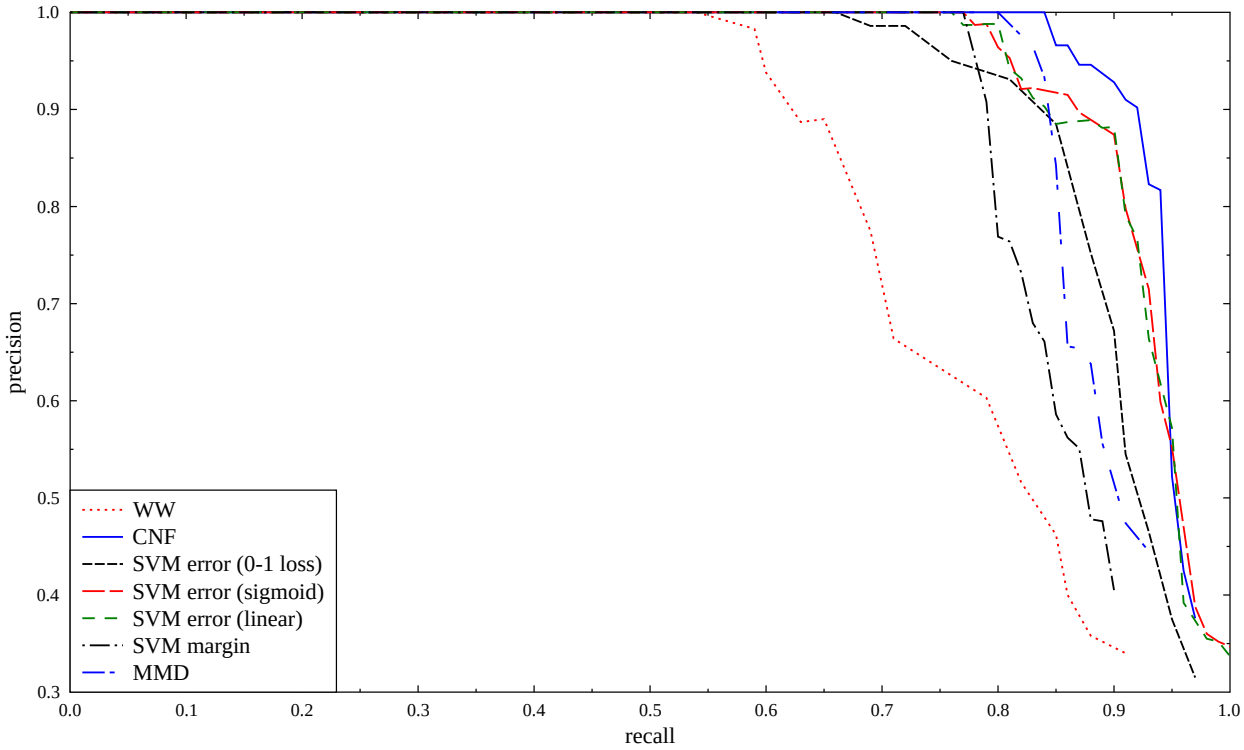


Figure 2: Precision-recall curve for six methods on the segment data set.

perform well on abrupt changes also do well on more gradual ones. We give an overview over the AUC values one can expect for $\Delta x = 100$ in Table 3.

5 Conclusion

In this paper we evaluated six new methods for concept drift detection in the online setting. We compared the methods to the Wald-Wolfowitz test, the Maximum Discrepancy statistic, and the StreamKrimp system. Traditional statistical methods for drift detection such as the multivariate Wald-Wolfowitz test are often based on rank statistics that do not adapt to the specific properties of the underlying data distribution. In contrast, we presented three new approaches, whose statistics adapt to detect data drift reliably. The first method is based on a density estimation technique on a binary representation of the data. The second method measures the 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 able to detect concept drifts reliably and are not too sensitive to noise in most cases. The experiments show that the SVM method with sigmoid loss performs best in a precision-recall analysis. The methods 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

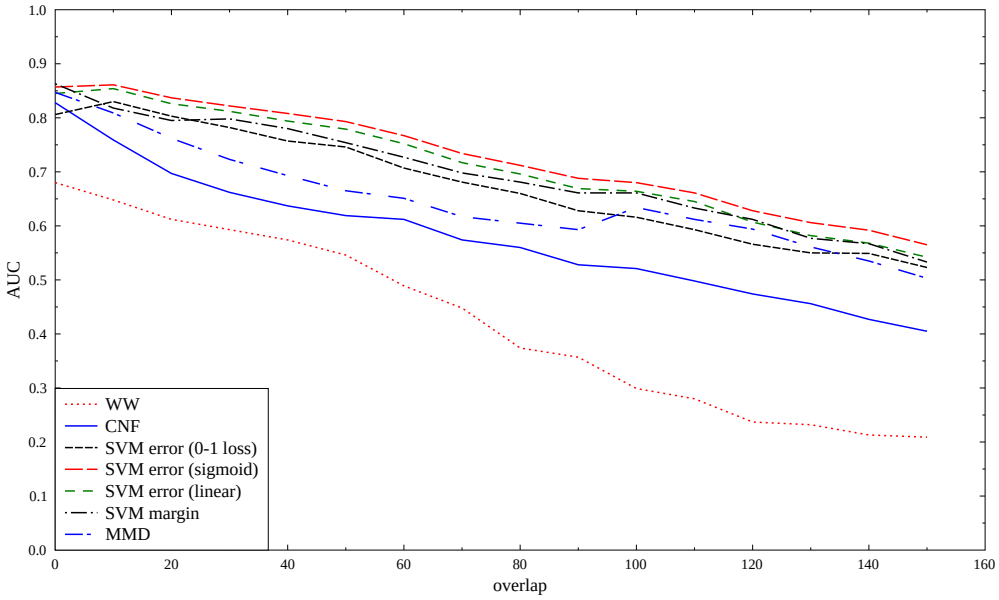


Figure 3: Change of AUC for the pendigits data set with increasing overlap.

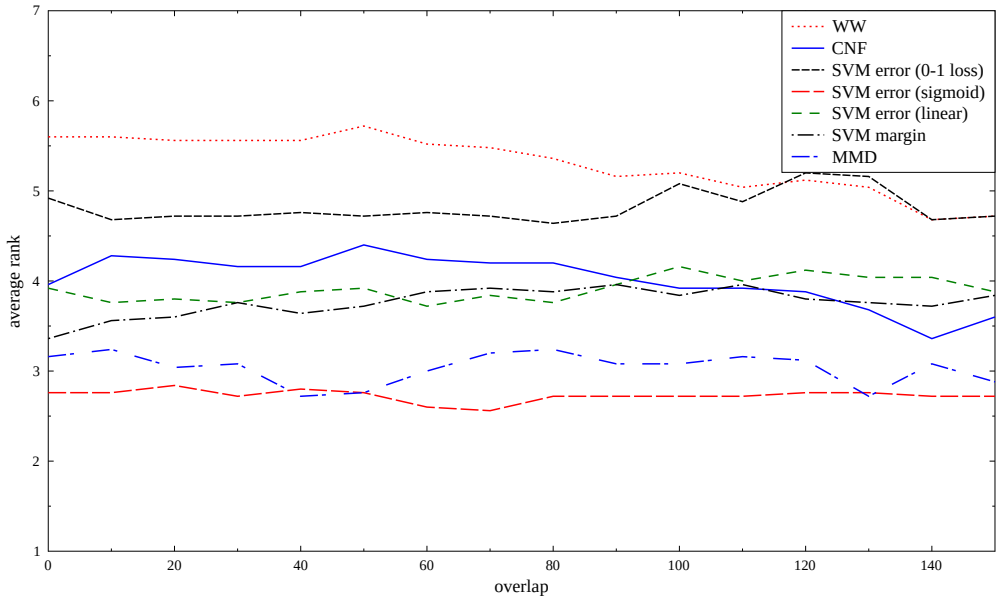


Figure 4: The average rank of each method over all data sets for drifts with increasing overlap. A rank of one means the method is best, a rank of six means the method is worst.

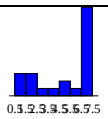
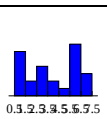
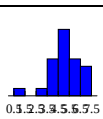
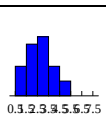
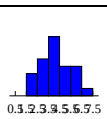
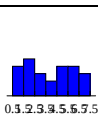
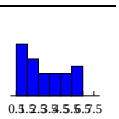
	WW	3-CNF	SVM 0	SVM S	SVM L	Margin	MMD
anneal	0.343	**0.555**	0.412	0.480	0.472	0.363	0.518
balance-scale	0.558	0.646	0.615	**0.667**	0.627	0.660	0.625
breast-w	0.449	0.713	0.745	0.786	0.759	0.803	**0.808**
car	0.176	**0.862**	0.326	0.341	0.329	0.314	0.284
colic	0.463	0.329	0.118	0.203	0.166	**0.712**	0.304
credit-a	**0.550**	0.274	0.486	0.547	0.523	0.511	0.543
credit-g	0.206	0.170	0.288	0.393	0.329	0.125	**0.457**
diabetes	0.187	0.208	0.313	0.334	0.306	**0.531**	0.353
haberman	0.212	**0.368**	0.041	0.052	0.031	0.064	0.040
heart-c	**0.857**	0.700	0.470	0.594	0.490	0.590	0.693
heart-h	0.471	0.618	0.569	0.624	0.594	0.550	**0.679**
kr-vs-kp	0.311	**0.570**	0.341	0.399	0.395	0.149	0.221
letter	0.380	0.558	0.541	0.553	0.540	**0.562**	0.548
mfeat-morph	0.786	0.621	0.773	0.800	0.781	0.824	**0.927**
nursery	0.201	0.000	0.153	0.164	0.161	0.089	**0.373**
optdigits	0.479	0.675	0.387	0.664	0.559	**0.813**	0.734
page-blocks	0.095	0.497	0.449	**0.545**	0.524	0.499	0.426
pendigits	0.299	0.521	0.616	**0.682**	0.665	0.661	0.634
segment	0.469	0.705	0.751	0.774	0.757	0.694	**0.828**
sick	0.060	**0.338**	0.077	0.084	0.082	0.201	0.159
tic-tac-toe	0.360	**0.850**	0.291	0.317	0.324	0.153	0.375
vehicle	0.618	0.657	**0.785**	0.777	0.757	0.666	0.663
vote	0.228	0.515	0.662	0.704	0.672	0.810	**0.883**
waveform	**0.420**	0.323	0.113	0.338	0.281	0.318	0.344
yeast	0.120	0.192	0.315	**0.395**	0.357	0.278	0.233
Distribution of Positions							

Table 3: Comparison of AUC-PR values for different datasets for a gradual drift with $\Delta x = 100$.

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, [19]) *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 [18].

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

$$\underset{\overline{X},\underline{X}}{\mathbf{Pr}}\left[\sum_{i=1}^m v_i e^{(0.5n'-1)D_r^2} \leq \frac{n'}{\delta}\right] \geq 1 - \delta \tag{4}$$

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

$$\underset{\overline{X},\underline{X}}{\mathbf{Pr}}\left[D_r \geq x\right] \leq 2\exp\Big[-0.5x^2 n'\Big] \tag{5}$$

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 (5) 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,

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

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

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

Inequality (4) follows from this and Markov's inequality. Applying Lemma 6.1 to (4) with $K = \frac{n'}{\delta}, R = R_w, S = R_v, x = (D_1, \ldots, D_m)^T, \beta = 0.5n' - 1$ yields:

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

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,

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

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:

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

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 and one can apply the union bound over these $d(n,m)$ cases:

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

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

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

Taking the expectation on both sides, we have that

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

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 $\frac{1}{n}$. Likewise, changing an example in the second sample $\underline{X}$ changes the value of E' by at most $\frac{1}{n}$. McDiarmid's inequality (Theorem 6.1) ensures that

$$\begin{aligned}\mathbf{Pr}\left[E' - \mathop{\mathbf{E}}_{\overline{X},\underline{X}} E' \geq s\right] &\leq \exp\left[-\frac{2s^2}{\sum_{i=1}^{\overline{n}}\frac{1}{\overline{n}^2} + \sum_{i=1}^{\underline{n}}\frac{1}{\underline{n}^2}}\right] \\ &= \exp\left(-s^2 n\right) \qquad (6)\end{aligned}$$

Setting $s = t - \mathbf{E}[E']$ it suffices to show that $\mathbf{E}[E'] \leq 2\sqrt{m/n}$ 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,

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

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:

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

(7)

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

(8)

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

(9)

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

Here, (7) is a consequence of Jensen's inequality and the convexity of the supremum, (8) is an application of theorem 4.12 in [16] and the fact that $l_s(.)$ is Lipschitz with Lipschitz constant $\frac{p}{4}$. Finally, (9) 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 (9) can be further bounded as follows:

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

(10)
$$\leq p\underset{\overline{X},\sigma}{\mathbf{E}} \sqrt{\sum_{j=1}^{m}\left|\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i\overline{x}_i\right]_j\right|^2}$$

(11)
$$\leq p\sqrt{\sum_{j=1}^{m}\underset{\overline{X},\sigma}{\mathbf{E}}\left|\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i\overline{x}_i\right]_j\right|^2}$$

(12)
$$\leq p\sqrt{\sum_{j=1}^{m}\frac{1}{n^2}\underset{\sigma}{\mathbf{E}}\left|\sum_{i=1}^{n}\sigma_i\right|^2}$$

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

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

Inequality (10) follows because $\|x\|_\infty \leq \|x\|_2$ for all $x \in \mathbb{R}^m$, (11) is an application of Jensen's inequality and the concavity of the square root, while (12) 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).

6.4 Proof of Theorem 3.4

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

$$f_p(x) := \begin{cases} \frac{1}{2} & \text{if } x \geq \frac{2}{p} \\ \frac{p}{4}x & \text{if } |x| < \frac{2}{p} \\ -\frac{1}{2} & \text{if } x \leq -\frac{2}{p} \end{cases}$$

$$L_i(w) := l_l(w^T\overline{x}_i) - l_l(w^T\underline{x}_i)$$
$$= f_p(w^T\underline{x}_i) - f_p(w^T\overline{x}_i).$$

Then,

$$\mathbf{Pr}[E \geq t] = \mathbf{Pr}\left[\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}L_i(w)\right] \geq t\right]$$
$$= \mathbf{Pr}\left[\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i L_i(w)\right] \geq t\right]$$

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. under the null hypothesis, the expectation remains the same. Applying the union bound, we get:

$$\mathbf{Pr}[E \geq t] = \mathbf{Pr}\left[\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i L_i(w)\right] \geq t\right]$$
$$\leq 2\,\mathbf{Pr}\left[\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i f_p(w^T\overline{x}_i)\right] \geq \frac{t}{2}\right]$$

We proceed by bounding this probability conditional to a fixed data sample $\overline{X}$. Let

$$P_1 := \mathbf{Pr}\left[\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i f_p(w^T\overline{x}_i)\right] \geq \frac{t}{2}\middle|\overline{X}\right]$$

To bound this, we make use of the fact that the supremum can be written as a supremum over a convex set with a limited number of solutions. More precisely,

define $C_{\overline{X}} := \{(\frac{p}{4}w^T\overline{x}_1, \ldots, \frac{p}{4}w^T\overline{x}_n)^T | w \in \mathcal{B}_1^m\}$. Then, we have for a sample $\overline{X}$:

(13)
$$\sup_{w\in\mathcal{B}_1^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i f_p(w^T\overline{x}_i)\right] = \sup_{c\in C_{\overline{X}}\cap[-0.5,0.5]^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i c_i\right]$$

Now, since $C_{\overline{X}}$ is a m-dimensional convex set, the supremum $\sup_{c\in C_{\overline{X}}}[\sum_{i=1}^{n}\sigma_i c_i]$ is achieved at one of the $2m$ vertices. This means that the supremum in (13) is also over at most $2m$ possible candidates. Thus, we can apply the union bound and McDiarmid's inequality to obtain:

$$\begin{aligned}
P_1 &= \mathbf{Pr}\left[\sup_{c\in C_{\overline{X}}\cap[-0.5,0.5]^m}\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i c_i\right] \geq \frac{t}{2}\middle|\overline{X}\right] \\
&\leq 2m \sup_{c\in C_{\overline{X}}\cap[-0.5,0.5]^m}\mathbf{Pr}\left[\left[\frac{1}{n}\sum_{i=1}^{n}\sigma_i c_i\right] \geq \frac{t}{2}\middle|\overline{X}\right] \\
&\leq 2me^{-2(\frac{t}{2})^2 n}
\end{aligned}$$

Taking expectations yields the result.

Acknowledgements

We would like to thank Luc De Raedt and Siegfried Nijssen for helpful discussions and Matthijs Van Leeuwen for providing us with their latest version of StreamKrimp. Some 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 J. Rasch, Bernhard Schölkopf, and Alexander J. Smola. A kernel method for the two-sample-problem. In Bernhard Schölkopf, John C. Platt, and Thomas Hoffman, editors, *Advances in Neural Information Processing Systems 19, Proceedings of the Twentieth Annual Conference on Neural Information Processing Systems*, pages 513–520. MIT Press, 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] Daniel Kifer, Shai Ben-David, and Johannes Gehrke. Detecting change in data streams. In Mario A. Nascimento, M. Tamer Özsu, Donald Kossmann, Renée J. Miller, José A. Blakeley, and K. Bernhard Schiefer, editors, *Proceedings of the Thirtieth International Conference on Very Large Data Bases, Toronto, Canada, August 31 - September 3 2004*, pages 180–191. Morgan Kaufmann, 2004.

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

[15] 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.

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

[17] Matthijs Leeuwen and Arno Siebes. Streamkrimp: Detecting change in data streams. In *ECML PKDD '08: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I*, pages 672–687, Berlin, Heidelberg, 2008. Springer-Verlag.

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

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

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

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

[22] Xiuyao Song, Mingxi Wu, Christopher Jermaine, and Sanjay Ranka. Statistical change detection for multi-dimensional data. In *KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 667–676, New York, NY, USA, 2007. ACM.

[23] 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.

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

[25] 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.

[26] 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.

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

[28] 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.

[29] 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.

dataset	WW			CNF			Error (0/1)			Error (sigmoid)			Error (linear)			Margin			MMD			Stream Krimp
anneal	•	26.8	4.1 (0)	•	33.5	4.7 (0)	•	9.1	3.6 (0)	•	7.4	2.9 (0)	•	10.3	3.3 (0)	•	24.1	3.6 (0)	•	> 99	7.4 (1)	(0)
balance-scale	•	49.7	3.3 (0)	•	51.0	4.6 (0)	•	28.7	6.1 (0)	•	10.5	3.4 (0)	•	29.6	5.2 (0)	•	24.5	3.1 (0)	•	> 99	7.7 (1)	• (0)
breast-w	•	56.6	4.8 (0)	•	> 99	4.9 (0)	•	24.7	4.3 (0)	•	8.7	2.6 (0)	•	25.3	3.8 (0)	•	54.5	3.8 (0)	•	> 99	4.9 (0)	• (0)
car	•	10.3	5.9 (1)	•	49.3	7.2 (1)	•	12.3	4.5 (0)	•	7.2	3.3 (0)	•	12.7	4.3 (0)	•	19.0	6.4 (0)	•	91.7	9.5 (2)	• (0)
colic	•	13.2	3.3 (0)	•	7.1	3.1 (0)		5.0	2.3 (0)	•	6.0	2.2 (0)	•	6.0	2.3 (0)	•	27.5	3.7 (0)	•	89.0	6.7 (0)	(0)
credit-a	•	10.9	5.5 (0)	•	6.8	3.6 (0)	•	11.0	3.6 (0)	•	9.4	3.0 (0)	•	13.7	3.4 (0)	•	51.6	5.2 (0)	•	> 99	6.2 (1)	(0)
credit-g		4.8	5.4 (0)		3.3	4.3 (0)		4.5	2.8 (0)		4.5	2.2 (0)		5.3	2.4 (0)		4.9	4.0 (0)	•	25.5	8.1 (0)	(0)
diabetes		4.6	3.5 (0)		1.3	5.6 (0)	•	8.6	5.5 (0)		4.9	3.2 (0)	•	8.9	4.7 (0)	•	9.0	4.9 (0)	•	13.0	8.9 (1)	(0)
haberman		1.2	4.4 (0)	•	7.3	2.8 (0)		4.4	3.7 (0)		2.1	3.3 (0)		2.4	3.7 (0)		3.5	2.4 (0)	•	9.5	4.9 (0)	(0)
heart-c	•	51.7	3.8 (0)	•	29.6	2.5 (0)	•	11.9	2.7 (0)	•	8.1	2.2 (0)	•	13.9	2.6 (0)	•	32.5	2.5 (0)	•	> 99	2.7 (0)	• (0)
heart-h	•	23.4	2.9 (0)	•	21.9	4.6 (0)	•	10.2	2.8 (0)	•	8.3	1.8 (0)	•	13.2	2.3 (0)	•	28.4	3.3 (0)	•	> 99	5.1 (0)	• (0)
heart-statlog	•	14.2	3.5 (0)	•	14.7	4.7 (0)	•	16.9	3.5 (0)	•	9.1	2.0 (0)	•	18.1	2.9 (0)	•	10.0	2.8 (0)	•	> 99	3.6 (0)	• (0)
ionosphere	•	20.2	4.0 (0)	•	12.0	3.5 (0)	•	10.8	3.3 (0)	•	5.8	2.8 (0)	•	9.6	3.4 (0)	•	12.3	4.1 (0)	•	> 99	5.7 (0)	ERR
kr-vs-kp	•	24.7	6.4 (0)	•	40.8	6.3 (1)	•	10.7	3.6 (0)	•	9.0	3.3 (0)	•	12.5	3.9 (0)	•	13.0	6.5 (1)	•	39.2	20.0 (2)	(0)
letter	•	39.1	5.1 (0)	•	48.6	5.4 (0)	•	31.3	6.1 (0)	•	13.1	3.2 (0)	•	36.0	4.4 (0)	•	47.8	4.7 (0)	•	> 99	7.1 (2)	• (0)
mfeat-morph	•	47.5	3.3 (0)	•	> 99	2.9 (0)	•	> 99	3.3 (0)	•	49.0	2.6 (0)	•	> 99	3.4 (0)	•	> 99	3.8 (0)	•	> 99	3.0 (0)	• (0)
nursery	•	37.0	6.4 (1)		-0.7	4.9 (0)	•	12.4	3.8 (0)	•	7.6	2.8 (0)	•	13.3	3.7 (0)	•	11.8	5.3 (1)	•	> 99	9.2 (4)	(0)
optdigits	•	40.8	4.3 (0)	•	> 99	4.1 (0)	•	10.2	3.4 (0)	•	10.6	3.3 (0)	•	14.5	3.5 (0)	•	79.2	5.1 (0)	•	> 99	10.0 (2)	ERR
page-blocks	•	33.2	6.8 (1)	•	29.1	5.3 (1)	•	35.5	6.2 (2)	•	11.2	3.4 (0)	•	36.3	5.2 (0)	•	40.0	6.3 (1)	•	> 99	12.8 (7)	• (0)
pendigits	•	43.9	5.6 (1)	•	> 99	5.7 (2)	•	31.9	5.0 (0)	•	12.9	3.0 (0)	•	35.8	4.4 (0)	•	81.2	6.0 (0)	•	> 99	13.3 (3)	• (0)
segment	•	40.4	3.7 (0)	•	33.8	4.0 (0)	•	48.0	5.1 (0)	•	16.1	2.8 (0)	•	58.8	4.0 (0)	•	81.8	3.7 (0)	•	> 99	9.1 (1)	• (0)
sick	•	12.9	6.0 (0)	•	13.0	7.4 (1)	•	6.1	4.5 (0)		4.7	3.6 (0)	•	6.4	4.5 (0)	•	14.2	6.0 (0)	•	57.4	11.8 (4)	(0)
tic-tac-toe	•	46.3	4.5 (0)	•	> 99	5.5 (0)	•	14.3	3.6 (0)	•	7.2	2.9 (0)	•	14.2	3.6 (0)	•	13.9	3.6 (0)	•	24.1	5.7 (0)	(0)
vehicle	•	42.8	5.3 (0)	•	23.5	3.4 (0)	•	44.4	5.2 (0)	•	14.8	2.5 (0)	•	47.2	3.3 (0)	•	45.5	2.8 (0)	•	> 99	3.5 (0)	• (0)
vote	•	28.7	3.5 (0)	•	> 99	4.4 (0)	•	15.6	3.0 (0)	•	8.5	2.0 (0)	•	16.7	2.6 (0)	•	60.3	2.8 (0)	•	> 99	2.5 (0)	• (0)
waveform	•	34.9	7.2 (1)	•	16.6	5.6 (1)	•	9.1	3.8 (0)	•	10.5	3.5 (0)	•	14.0	4.0 (0)	•	30.7	5.5 (1)	•	> 99	14.1 (2)	ERR
yeast	•	8.6	4.3 (0)		4.6	4.0 (0)	•	6.0	4.7 (0)		4.3	3.0 (0)	•	7.2	4.7 (0)	•	10.7	4.6 (0)	•	30.0	7.0 (0)	(0)

Table 4: Results for the experiments. Columns represent correct detection (for s=5), difference in standard deviations between p-value in correct interval and average, largest value outside drift window (in standard deviations) and number of false detections (for s=5).