Detecting Change in Data Streams

Shai Ben-David, Johannes Gehrke, Daniel Kifer*

Cornell University

Abstract

Detecting changes in a data stream is an important area of research with many applications. In this paper, we present novel methods for the detection *and* estimation of change. In contrast to previously proposed tools, our techniques provide proven guarantees on the statistical significance of detected change. In addition to providing reliable detection of change, our method allows for a meaningful description and quantification of those changes. Our techniques are nonparametric and so they require no prior assumptions on the nature of the distribution that generates the data, except for assuming that points in the stream are generated independently. Additionally, these techniques work for both continuous and discrete data. In an experimental study we demonstrate the usefulness of our techniques.

1 Introduction

In many applications data is not static but arrives in data streams. Besides the algorithmic difference between processing data streams and static data, there is another significant difference. For static datasets, it is reasonable to assume that the data was generated by a fixed process, for example, the data was sampled from a fixed distribution. But a data stream has necessarily a temporal dimension, and the underlying process that generates the data stream can change over time [17, 1, 22]. The quantification and detection of such change is one of the fundamental challenges in data stream settings.

Change has far-reaching impact on any data processing algorithm. For example, when constructing data stream mining models [17, 1], data that arrived before a change can bias the models towards characteristics that no longer hold. If we process queries over data streams, we may want to give separate answers for each time interval where the underlying data distribution is stable. Most existing work has concentrated on algorithms that adapt to changing distributions either by discarding old data or giving it less weight [17]. However, to the best of our knowledge, previous work does not contain a formal definition of change and thus existing algorithms cannot specify precisely when and how the underlying distribution changes.

In this paper we make a first step towards formally introducing and quantifying change in a data stream. We view the data as being generated by some underlying probability distribution, one data point at a time, in an independent fashion. Our goal is to detect when this data-generating distribution changes, and to quantify and describe this change.

It is unrealistic to allow data stream processing algorithms enough memory capacity to store the full history of the stream. Therefore we base our change-detection algorithm on a two-window paradigm. The algorithm compares the data in some 'reference window' to the data in a current window. Both widows contain a fixed number of successive data points. The current window slides forward with each incoming data point, and the reference window is updated whenever a change is detected.

We analyze this paradigm and seek algorithms (or 'tests') that can be supported by proven guarantees of their sensitivity to change, their robustness against raising false alarms and their running time. Furthermore, we aim to obtain not only reliable change detection, but also a comprehensible description of the nature of the detected change.

1.1 Applications

A change detection test with the above properties has many interesting applications:

Environmental Monitoring. While studying the effects of a nuclear power plant on the local environment, a researcher can measure the strength of the emitted radiation. A statistically significant change in distribution can signal a potential problem in the

Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004

power plant (even if the radiation has not reached critical levels). It also allows for the possibility for more refined analysis of ecological data that is collected at different points in time.

Quality Control. A factory is manufacturing beams made from a metal alloy. The strengths of a sample of the beams can be measured during quality testing. Estimating the amount of defective beams and determining the statistical significance of this number are well-studied problems in the statistics community [5]. However, even if the number of defective beams does not change, the factory can still benefit by analyzing the distribution of beam strengths over time. Changes in this distribution can signal the development of a problem or give evidence that a new manufacturing technique is creating an improvement. Information that describes the change could also help in analysis of the technique.

Theoretical Models. Suppose a data stream mining algorithm is creating a model of the process that is generating the data. Ideally, this model is robust under small changes in the stream distribution - it should still have high accuracy. However, for sufficiently large changes in the distribution, the model will become inaccurate. In this case it is better to completely remove old data (which arrived before the change) from the model rather than to wait for enough new data to come in and outweigh the stale data. Further information about the change could help the user avoid rebuilding the entire model - if the change is localized, it may only be necessary to rebuild part of the model.

1.2 Statistical Requirements

Our basic approach to change detection in data streams uses two sliding windows over the data stream. This reduces the problem of detecting change over a data stream to the problem of testing whether two samples were generated by different distributions. Consequently, we start by considering the simpler case of detecting a difference in distribution between two input samples.

Assume that two datasets S_1 and S_2 where generated by two probability distributions, P_1, P_2. A natural question to ask is: Can we infer from S_1 and S_2 whether they were generated by the same distribution $P_1 = P_2$, or is it the case that $P_1 \neq P_2$? Thus we would like to design a "test" that can tell us whether P_1 and P_2 are different. What are our requirements on such a test? We would like a solution to this problem to have two-way guarantees: we want a guarantee that when a change occurs it is detected, and we also want to limit the amount of false alarms. Furthermore we need to be able to extend those guarantees from the two-sample problem to the datastream.

There are also practical considerations that affect our choice of test. If we know that the distribution of the data has a certain parametric form (for example, it is a normal distribution), then we can draw upon decades of research in the statistics community where powerful tests have been developed (for some parametric distributions, there are tests that are provably the most powerful). Unfortunately, data is not that nice in practice; real data does not follow "nice" parametric distributions. Thus we want a non-parametric test that comes with formal guarantees. Our last requirement is motivated by the users of our test. A user not only wants to know that the underlying distribution has changed, but also wants to know *how* it has changed. Thus we want a test that does not only detects changes, but also *describes* the change in a user-understandable way.

In summary, we want a change detection test with four properties: We can control false positives, we can control false negatives, it is non-parametric, and when it detects a change it also provides a description of the change.

1.3 A Discussion of Common Tests

We have already discarded parametric tests, but some non-parametric tests already exist. One well-known non-parametric statistical test is called the Wilcoxon. With this test we can limit the probability of a false alarm, but we only get guarantees for the detection of limited kinds of changes, such as a change in the mean of a normal distribution. We address the issue of providing guarantees about the changes we can detect by *quantifying* the difference between distributions. Our guarantees will then take the form: "to detect a difference $> \epsilon$ we will need samples of at most n points."

Another shortcoming of the Wilcoxon test is that it is not very descriptive. Instead, one may try to use an information theoretic measure, such as the Jensen-Shannon Divergence (JSD). However, not only are there no known general statistical guarantees for JSD, but it may also be hard to explain the idea of entropy to the average user. Indeed, the description of arbitrary change can be arbitrarily complex and arbitrarily hard to explain to users. We address this vexing problem by noting that, in all likelihood, the user is not interested in *arbitrary* change. We localize the notion of change and show which interval, or more generally, which hyper-rectangle (in the attribute space) is most greatly affected by the distribution change. We formally introduce this notion of change in Section 3.

Now, there exist many other common measures of distance between distributions, but they are either too insensitive or too sensitive. For example, the commonly used L^1 distance between two distributions is too sensitive and can require arbitrarily large samples to determine if two distributions have L^1 distance $> \epsilon$ [4]. At the other extreme, L^p norms (for $p > 1$) are far too insensitive: two distributions D_1 and D_2 can be close in the L^p norm and yet have disjoint support. Thus, in Section 3, we introduce a new distance

metric that is specifically tailored to find distribution changes while providing strong statistical guarantees with small sample sizes.

1.4 Our Contributions

In this paper we give the first formal treatment of change in data streams. Our techniques are non-parametric (they do not make any assumptions about the distribution that generated the data stream), they give provable guarantees that the change that is detected is not only noise but statistically significant, and they allow us to describe the change to a user of our methods. To the best of our knowledge, there is no previous work that addresses any of these requirements – although they are crucial for real data.

The remainder of this paper is organized as follows. After a description of our meta-algorithm in Section 2, we introduce novel metrics over the space of distributions and show that they avoid statistical problems common to previously known distance functions (Section 3). We then show how to apply these metric to detect changes in the data stream setting, and we give strong statistical guarantees on the types of changes that are detected (Section 4). In Section 5 we develop algorithms that efficiently find the areas where change has occurred, and we evaluate our techniques in a thorough experimental analysis in Section 6.

2 A Meta-Algorithm For Change Detection

In this section we describe our meta-algorithm for change detection in streaming data. The meta-algorithm reduces the problem from the streaming data scenario to the problem of comparing two (static) sample sets. We consider a datastream S to be a sequence $< s_1, s_2, \cdots >$ where each item s_i is generated by some distribution P_i and each s_i is independent of the items that came before it. We say that a change has occurred if $P_i \neq P_{i+1}$, and we call time $i+1$ a *change point*[1]. We also assume that only a bounded amount of memory is available, and that in general the size of the data stream is much larger than the amount of available memory.

The meta-algorithm requires a function d, which measures the discrepancy between two samples, and a set of triples $\{(m_{1,1}, m_{2,1}, \alpha_1), \ldots, (m_{1,k}, m_{2,k}, \alpha_k)\}$. The numbers $m_{1,i}$ and $m_{2,i}$ specify the sizes of the i^{th} pair of windows (X_i, Y_i). The window X_i is a 'baseline' window and contains the first $m_{1,i}$ points of the stream that occurred after the last detected change. The window Y_i is a sliding window that contains the latest $m2, k$ items in the data stream. Immediately after a change has been detected, it contains the $m_{2,i}$ points of the stream that follow the window X_i. We slide the window Y_i one step forward whenever a new item appears on the stream. At each such update, we check if $d(X_i, Y_i) > \alpha_i$. Whenever the distance is $> \alpha_i$, we report a change and then repeat the entire procedure with X_i containing the first $m_{1,i}$ points after the change, etc. The meta-algorithm is shown in Figure 1.

[1]It is not hard to realize that no algorithm can be guaranteed to detect any such change point. We shall therefore require the detection of change only when the difference between P_i and P_{i+1} is above some threshold. We elaborate on this issue in section 3.

Algorithm 1 : FIND_CHANGE

```
1: for i = 1...k do
2:    c_0 ← 0
3:    Window_{1,i} ← first m_{1,i} points from time c_0
4:    Window_{2,i} ← next m_{2,i} points in stream
5: end for
6: while not at end of stream do
7:    for i = 1...k do
8:       Slide Window_{2,i} by 1 point
9:       if d(Window_{1,i}, Window_{2,i}) > α_i then
10:         c_0 ← current time
11:         Report change at time c_0
12:         Clear all windows and GOTO step 1
13:      end if
14:   end for
15: end while
```

It is crucial to keep the window X_i fixed while sliding the window Y_i, so that we always maintain a reference to the original distribution. We use several pairs of windows because small windows can detect sudden, large changes while large windows can detect smaller change that lasts over a long period of time.

The key to our scheme is the intelligent choice of distance function d and the constants α_i. The function d must truly quantify an intuitive notion of change so that the change can be explained to a non-technical user. The choice of such a d is discussed in Section 3. The α_i is a parameter that defines our balance between sensitivity and robustness of the detection. The smaller α_i is, the more likely we are to detect small changes in the distribution, but the larger is our risk of false alarm - announcing a change when no the underlying distribution remained unchanged. We wish to provide statistical guarantees about the accuracy of the change report. Providing such guarantees is highly non-trivial because of two reasons: we have no prior knowledge about the distributions and the changes, and the repeated testing of $d(X_i, Y_i)$ necessarily exhibits the multiple testing problem - the more times you run a random experiment, the more likely you are to see something strange. We deal with these issues in section 3 and section 4, respectively.

3 Distance Measures for Distribution Change

In this section we focus on the basic, two-sample, comparison. Our goal is to design algorithms that examine samples drawn from two probability distributions

and decide whether these distributions are different. Furthermore, we wish to have two-sided performance guarantees for our algorithms (or tests). Namely, results showing that if the algorithm accesses sufficiently large samples then, on one hand, if the samples come from the same distributions then the probability that the algorithm will output "CHANGE" is small, and on the other hand, if the samples were generated by different distributions, our algorithm will output 'CHANGE" with high probability. It is not hard to realize that no matter what the algorithm does, for every finite sample size there exist a pair of distinct distributions such that, with high probability, samples of that size will not suffice for the algorithm to detect that they are coming from different distributions. The best type of guarantee that one can conceivably hope to prove is therefore of the type: "If the distributions generating the input samples are sufficiently different, then sample sizes of a certain bounded size will suffice to detect that these distributions are distinct". However, to make such a statement precise, one needs a way to measure the degree of difference between two given probability distributions. Therefore, before we go on with our analysis of distribution change detection, we have to define the type of changes we wish to detect. This section addresses this issue by examining several notions of distance between probability distributions.

The most natural notion of distance (or similarity) between distributions is the *total variation* or the L^1 norm. Given two probability distributions, P_1, P_2 over the same measure space $(X, \mathcal{E})$ (where X is some domain set and $\mathcal{E}$ is a collection of subsets of X - the measurable subsets), the total variation distance between these distributions is defined as $TV(P_1, P_2) = 2\sup_{E\in\mathcal{E}} |P_1(E) - P_2(E)|$ (or, equivalently, when the distributions have density functions, f_1, f_2, respectively, the L^1 distance between the distributions is defined by $\int |f_1(x) - f_2(x)|dx$). Note that the total variation takes values in the interval $[0, 1]$.

However, for practical purposes the total variation is an overly sensitive notion of distance. First, because $TV(P_1, P_2)$ may be quite large for distributions that should be considered as similar for all practical purposes (for example, it is easy to construct two distributions that differ, say, only on real numbers whose 9th decimal point is 5, and yet their total variation distance is 0.2). The second, related, argument against the use of the total variation distance, is that it may be infeasibly difficult to detect the difference between two distributions from the samples they generate. Batu et al [4] prove that, over discrete domains of size n, for every sample-based change detection algorithm, there are pairs of distribution that have total variation distance $\geq 1/3$ and yet, if the sample sizes are below $O(n^{2/3})$, it is highly unlikely that the algorithm will detect a difference between the distributions. In particular, this means that over infinite domains (like the real line) any sample based change detection algorithm is bound to require arbitrarily large samples to detect the change even between distribution whose total variation distance is large.

We wish to employ a notion of distance that, on one hand captures 'practically significant' distribution differences, and yet, on the other hand, allows the existence of finite sample based change detection algorithms with proven detection guarantees.

Our solution is based upon the idea of focusing on a family of significant domain subsets.

Definition 1. Fix a measure space and let $\mathcal{A}$ be a collection of measurable sets. Let P and P' be probability distributions over this space.

- The $\mathcal{A}$-*distance* between P and P' is defined as

$$d_{\mathcal{A}}(P, P') = 2\sup_{A\in\mathcal{A}} |P(A) - P'(A)|$$

  We say that P, P' are ϵ-*close with respect to* $\mathcal{A}$ if $d_{\mathcal{A}}(P, P') \leq \epsilon$.

- For a finite domain subset S and a set $A \in \mathcal{A}$, let the *empirical weight* of A w.r.t. S be

$$S(A) = \frac{|S \cap A|}{|S|}$$

- For finite domain subsets, S_1 and S_2, we define the *empirical distance* to be

$$d_{\mathcal{A}}(S_1, S_2) = 2\sup_{A\in\mathcal{A}} |S_1(A) - S_2(A)|$$

The intuitive meaning of $\mathcal{A}$-distance is that it is the largest change in probability of a set that the user cares about. In particular, if we consider the scenario of monitoring environmental changes spread over some geographical area, one may assume that the changes that are of interest will be noticeable in some local regions and thus be noticeable by monitoring spatial rectangles or circles. This notion of $\mathcal{A}$-distance is a relaxation of the total variation distance, which is defined as $2\sup_E |P(E) - P'(E)|$ (where the sup is taken over all measurable sets). When P and P' both have densities, then the total variation is equal to the L^1 distance. Thus it is not hard to see that $\mathcal{A}$-distance is always $\leq$ the total variation or L^1 norm (if it exists) and therefore is less restrictive. This point helps get around the statistical difficulties associated with the L^1 norm. If $\mathcal{A}$ is not too complex[2], then there exists a test t that can distinguish (with high probability) if any two distributions are ϵ-close (with respect to $\mathcal{A}$)

[2]there is a formal notion of this complexity - the VC-dimension. We discuss it further in Section 3.

using a sample size that is independent of the domain size.

For the case where the domain set is the real line, the Kolmogorov-Smirnov statistics considers $\sup_x |F_1(x) - F_2(x)|$ as the measure of difference between two distributions (where $F_i(x) = P_i(\{y : y \leq x\})$). By setting $\mathcal{A}$ to be the set of all the one-sided intervals $(-\infty, x)$ the $\mathcal{A}$ distance becomes the Kolmogorov-Smirnov statistic. Thus our notion of distance, $d_{\mathcal{A}}$ can be viewed as a generalization of this classical statistics.

The $\mathcal{A}$-distance reflects the relevance of locally centered changes. However, having adopted the concept of determining distance by focusing on a family of relevant subsets, there are different ways of quantifying such a change. The $\mathcal{A}$ measure defined above is additive - the significance of a change is measured by the *difference* of the weights of a subset between the two distributions. Alternatively, one could argue that changing the probability weight of a set from 0.5 to 0.4 is less significant than the change of a set that has probability weight of 0.1 under P_1 and weight 0 under P_2.

Next, we develop a variation of notion of the $\mathcal{A}$ distance, called *relativized discrepancy*, that takes the relative magnitude of a change into account.

As we have clarified above, our aim is to not only define sensitive measures of the discrepancy between distributions, but also to provide statistical guarantees that the differences that these measures evaluate are detectable from bounded size samples. Consequently, in developing variations of the basic $d_{\mathcal{A}}$ measure, we have to take into account the statistical tool kit available for proving convergence of sample based estimates to true probabilities. In the next paragraph we outline the considerations that led us to the choice of our 'relativized discrepancy' measures.

Let P be some probability distribution and choose any $A \in \mathcal{A}$, let p be such that $P(A) = p$. Let S be a sample with generated by P and let n denote its size. Then $nS(A)$ behaves like the sum $S_n = X_1 + \cdots + X_n$ of $|S|$ independent binomial random variables with $P(X_i = 1) = p$ and $P(X_i = 0) = 1 - p$. We can use Chernoff bounds [16] to approximate that tails of the distribution of S_n:

$$P\left[S_n/n \geq (1+\epsilon)p\right] \leq e^{-\epsilon^2 np/3} \quad (1)$$

$$P\left[S_n/n \leq (1-\epsilon)p\right] \leq e^{-\epsilon^2 np/2} \quad (2)$$

Our goal is to find an expression for ϵ as a function $\omega(p)$ so that the rate of convergence is approximately the same for all p. Reasoning informally, $P(p - S_n/n \geq p\omega(p)) \approx e^{-\omega(p)^2 np/2}$ and the right hand side is constant if $\omega(p) = 1/\sqrt{p}$. Thus

$$P\left[(p - S_n/n)/\sqrt{p} > \epsilon\right]$$

should converge at approximately the same rate for all p. If we look at the random variables $X_1^*, \ldots, X_n^*$ (where $X_i^* = 1 - X_i$) we see that $S_n^* = \sum X_i^* = n - S_n$ is a binomial random variable with parameter $1 - p$. Therefore the rate of convergence should be the same for p and $1 - p$. To make the above probability symmetric in p and $1 - p$, we can either change the denominator to $\sqrt{min(p, 1-p)}$ or $\sqrt{p(1-p)}$. The first way is more faithful to the Chernoff bound. The second approach approximates the first approach when p is far from $1/2$. However, the second approach gives more relative weight to the case when p is close to $1/2$.

Substituting $S(A)$ for S_n/n, $P(A)$ for p, we get that $(P(A) - S(A))/\sqrt{\min(P(A), 1 - P(A))}$ converges at approximately the same rate for all A such that $0 < P(A) < 1$ and $(P(A) - S(A))/\sqrt{P(A)(1 - P(A))}$ converges at approximately the same rate for all A (when $0 < P(A) < 1$). We can modify it to the two sample case by approximating $P(A)$ in the numerator by $S`(A)$. In the denominator, for reasons of symmetry, we approximate $P(A)$ by $(S`(A)) + S(A))/2$. Taking the absolute values and the sup over all $A \in \mathcal{A}$, we propose the following measures of distribution distance, and empirical statistics for estimating it:

Definition 2 (Relativized Discrepancy). Let P_1, P_2 be two probability distributions over the same measure space, let $\mathcal{A}$ denote a family of measurable subsets of that space, and A a set in $\mathcal{A}$.

- Define $\phi_{\mathcal{A}}(P_1, P_2)$ as

$$\sup_{A \in \mathcal{A}} \frac{|P_1(A) - P_2(A)|}{\sqrt{\min\{\frac{P_1(A)+P_2(A)}{2}, (1 - \frac{P_1(A)+P_2(A)}{2})\}}}$$

- For finite samples S_1, S_2, we define $\phi_{\mathcal{A}}(S_1, S_2)$ similarly, by replacing $P_i(A)$ in the above definition by the empirical measure $S_i(A) = |S_i \cap A|/|S_i|$.

- Define $\Xi_{\mathcal{A}}(P_1, P_2)$ as

$$\sup_{A \in \mathcal{A}} \frac{|P_1(A) - P_2(A)|}{\sqrt{\frac{P_1(A)+P_2(A)}{2}\left(1 - \frac{P_1(A)+P_2(A)}{2}\right)}}$$

- Similarly, for finite samples S_1, S_2, we define $\Xi_{\mathcal{A}}(S_1, S_2)$ by replacing $P_i(A)$ in the above definition by the empirical measure $S_i(A)$.

Our experiments show that indeed these statistics tend to do better than the $d_{\mathcal{A}}$ statistic because they use the data more efficiently - a smaller change in an area of low probability is more likely to be detected by these statistics than by the $D_{\mathcal{A}}$ (or the KS) statistic.

These statistics have several nice properties. The $d_{\mathcal{A}}$ distance is obviously a metric over the space of

probability distributions. So is the relativized discrepancy $|\phi_{\mathcal{A}}|$ (as long as for each pair of distribution P_1 and P_2 there exists a $A \in \mathcal{A}$ such that F_1 and $P_1(A) \neq P_2(A)$). For the proof, see the appendix. We conjecture that $|\Xi_{\mathcal{A}}|$ is also a metric.

However, the major benefit of the $d_{\mathcal{A}}$, $\phi_{\mathcal{A}}$, and $\Xi_{\mathcal{A}}$ statistics is that in addition to detecting change, they can describe it. All sets A which cause the relevant equations to be $> \epsilon$ are statistically significant. Thus the change can be described to a lay-person: the increase or decrease (from the first sample to the second sample) in the number of points that falls in A is too much to be accounted for by pure chance and therefore it is likely that the probability of A has increased (or decreased).

3.1 Technical preliminaries

Our basic tool for sample based estimation of the $\mathcal{A}$ distance between probaility distributions is based on the Vapnik-Chervonenkis theory.

Let $\mathcal{A}$ denote a family of subsets of some domain set X. We define a function $\Pi_{\mathcal{A}} : \mathbb{N} \mapsto \mathbb{N}$ by

$$\Pi_{\mathcal{A}}(n) = \max\{|\{A \cap B : A \in \mathcal{A}\}| : B \subseteq X \text{and} |B| = n\}$$

Clearly, for all n, $\Pi_{\mathcal{A}} \leq 2^n$. For example, if $\mathcal{A}$ is the family of all intervals over the real line, then $\Pi_{\mathcal{A}}(n) = O(n^2)$, $(0.5n^2 + 1.5n$, to be precise).

Definition 3 (VC-Dimension). The Vapnik-Chervonenkis dimension of a collection $\mathcal{A}$ of sets is

$$\text{VC-dim}(\mathcal{A}) = \sup\{n : \Pi_{\mathcal{A}}(n) = 2^n\}$$

The following combinatorial fact, known as Sauer's Lemma, is a basic useful property of the function $\Pi_{\mathcal{A}}$.

Lemma 3.1 (Sauer, Shelah). If $\mathcal{A}$ has a finite VC-dimension, d, then for all n,

$$\Pi_{\mathcal{A}}(n) \leq \Sigma_{i=0}^{d} \binom{n}{i}$$

It follows that for any such $\mathcal{A}$, $Pi_{\mathcal{A}}(n) < n^d$. In particular, for $\mathcal{A}$ being the family of intervals or rays on the real line, we get $Pi_{\mathcal{A}}(n) < n^2$.

3.2 Statistical Guarantees for our Change Detection Estimators

We consider the following scenario: P_1, P_2 are two probability distributions over the same domain X, and $\mathcal{A}$ is a family of subsets of that domain. Given two finite sets S_1, S_2 that are i.i.d. samples of P_1, P_2 respectively, we wish to estimate the $\mathcal{A}$ distance between the two distributions , $d_{\mathcal{A}}(P_1, P_2)$. Recall that, for any subset A of the domain set, and a finite sample S, we define the S- empirical weight of A by $S(A) = \frac{|S \cap A|}{|S|}$.

The following theorem follows by applying the classic Vapnik-Chervonenkis analysis [21], to our setting.

Theorem 3.1. Let P_1, P_2 be any probability distributions over some domain X and let $\mathcal{A}$ be a family of subsets of X and $\epsilon \in (0, 1)$. If S_1, S_2 are i.i.d m samples drawn by P_1, P_2 respectively, then,

$$P\left[\exists A \in \mathcal{A} \, ||P_1(A) - P_2(A)| - |S_1(A) - S_2(A)|| \geq \epsilon\right] < \Pi_{\mathcal{A}}(2m) 4e^{-m\epsilon^2/4}$$

It follows that

$$P\left[|d_{\mathcal{A}}(P_1, P_2) - d_{\mathcal{A}}(S_1, S_2)| \geq \epsilon\right] < \Pi_{\mathcal{A}}(2m) 4e^{-m\epsilon^2/4}$$

Where P in the above inequalities is the probability over the pairs of samples (S_1, S_2) induced by the sample generating distributions (P_1, P_2).

One should note that if $\mathcal{A}$ has a finite VC-dimension, d, then by Sauer's Lemma, $\Pi_{\mathcal{A}}(n) < n^d$ for all n.

We thus have bounds on the probabilities of both missed detections and false alarms of our change detection tests.

The rate of growth of the needed sample sizes as a function of the sensitivity of the test can be further improved by using the *relativized discrepancy* statistics. We can get results similar to Theorem 3.1 for the distance measures $\phi_{\mathcal{A}}(P_1, P_2)$ and $\Xi_{\mathcal{A}}(P_1, P_2)$. We start with the following consequence of a result of Anthony and Shawe-Taylor [2].

Theorem 3.2. Let $\mathcal{A}$ be a collection of subsets of a finite VC-dimension d. Let S be a sample of size n each, drawn i.i.d. by a probability distribution, P (over X), then

$$P^n(\phi_{\mathcal{A}}(S, P) > \epsilon) \leq (2n)^d e^{-n\epsilon^2/4}$$

(Where P^n is the n'th power of P - the probability that P induces over the choice of samples).

Similarly, we obtain the following bound on the probability of false alarm for the $\phi_{\mathcal{A}}(S_1, S_2)$ test.

Theorem 3.3. Let $\mathcal{A}$ be a collection of subsets of a finite VC-dimension d. If S_1 and S_2 are samples of size n each, drawn i.i.d. by the same distribution, P (over X), then

$$P^{2n}(\phi_{\mathcal{A}}(S_1, S_2) > \epsilon) \leq (2n)^d e^{-n\epsilon^2/4}$$

(Where P^{2n} is the $2n$'th power of P - the probability that P induces over the choice of samples).

To obtain analogous guarantees for the probabilities of missed detection of change, we employ the fact that $\phi_{\mathcal{A}}$ is a metric (see Appendix).

Claim 3.1. For finite samples, S_1, S_2, and a pair of probability distributions P_1, P_2 (all over the same domain set),

$$|\phi_{\mathcal{A}}(P_1, P_2) - \phi_{\mathcal{A}}(S_1, S_2)| \leq \phi_{\mathcal{A}}(P_1, S_1) + \phi_{\mathcal{A}}(P_2, S_2)$$

We can now apply Theorem 3.2 to obtain

Theorem 3.4. Let $\mathcal{A}$ be a collection of subsets of some domain measure space, and assume that the VC-dimension is some finite d. Let P_1 and P_2 be probability distributions over that domain and S_1, S_2 finite samples of sizes m_1, m_2 drawn i.i.d. according to P_1, P_2 respectively.

Then

$$P^{m_1+m_2}\left[|\phi_{\mathcal{A}}(S_1,S_2) - \phi_{\mathcal{A}}(P_1,P_2)| > \epsilon\right]$$

$$\leq (2m_1)^d e^{-m_1\epsilon^2/16} + (2m_2)^d e^{-m_2\epsilon^2/16}$$

(Where $P^{m_1+m_2}$ is the m_1+m_2'th power of P - the probability that P induces over the choice of samples).

Finally, note that, it is always the case that

$$\phi_{\mathcal{A}}(P_1,P_2) \leq \Xi_{\mathcal{A}}(P_1,P_2) \leq 2\phi_{\mathcal{A}}(P_1,P_2)$$

It therefore follows that guarantees against both false-positive and missed-detection errors similar to Theorems 3.3 and 3.4, hold for the $\Xi_{\mathcal{A}}$ statistics as well.

To appreciate the potential benefits of using this relative discrepancy approach, consider the case where $\mathcal{A}$ is the collection of all real intervals. It is easy to verify that the VC-dimension of this family $\mathcal{A}$ is 2. Let us estimate what sample sizes are needed to be 99% sure that an interval I, that changed from having no readings to having η fraction of the detected readings in this interval, indicate a real change in the measured field. Note that for such an interval, $\frac{S_1(I)-S_2(I)}{\sqrt{0.5(S_1(I)+S_2(I))}} = \sqrt{2\eta}$. We can now apply Theorem 3.3 to see that $m = 30/\eta$ should suffice. Note that if we used the $d_{\mathcal{A}}$ measure and Theorem 3.1, the bound we could guarantee would be in the order of $1/\eta^2$.

4 Tight Bounds for Streaming Real Data

Traditional statistical hypothesis testing consists of three parts: the null hypothesis, a test statistic, and a critical region. The null hypothesis is simply a statement about the distributions that generate the data. For example a null hypothesis could be: "samples A and B have the same generating distribution." A statistic is simply a function that is computed over the sampled data. For example, it could be the average, or the Wilcoxon statistic, or the number of heads in a series of coin tossings. A critical region (or rejection region) is a subset of the range of the statistic. If the value of the statistic falls in the critical region, we reject the null hypothsis. Generally, one wants to be careful about rejecting the null hypothesis and wants to limit the probability of this occurring. Therefore critical regions are designed so that if the null hypothesis were true, the probability that the test statistic will take a value in the critical region is less than some user-specified constant.

This framework does not fare very well when dealing with a data stream. For example, suppose a datastream is generated in the following way: an adversary has a collection of biased coins $c_1, c_2, \ldots$ such that the probability that c_i lands with heads side up is p_i and $0 < p_i < 1$ for all i. At each time unit, the adversary flips a coin and reports its results. The adversary can secretly switch coins at any time.

Even if the adversary never switches coins, any pattern of heads and tails will eventually show up in the stream, and thus for any test statistic of bounded memory (that cannot keep track of the length of the sequence) and non-trivial critical region, we will eventually get a value that causes us to falsely reject the null hypothesis (that the coin is fair) and accuse the adversary of switching coins.

Since there is no way to avoid mistakes all together, we direct our efforts to limiting the rate of mistakes. Ideally, we want to construct a critical region such that if the null hypothesis were true, then the expected number of times (per n points) that the test statistic falls in the critical region is small. Generally, it is difficult to compute the expected number of mistakes and so it is difficult to create the desired critical region. Thus we propose the following measure of statistical guarantee against false positive errors, in the spirit of the error rate:

Definition 4 (size). A statistical test over data streams is a $size(n,p)$ test if, on data that satisfies the null hypothesis, the probability of rejecting the null hypothesis after the first n points is at most p.

In the rest of this section we will show how to construct a critical region (given n and p) for the Wilcoxon, Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$, and $\Xi_{\mathcal{A}}$ tests. The critical region will have the form $\{x : x \geq \alpha\}$. In other words, we reject the null hypothesis for inordinately large values of the test statistic.

For the rest of this section, we will assume that the points of a stream $S =< s_1, s_2, \cdots >$ are real-valued and that the collection $\mathcal{A}$ is either a collection of all initial segments $(-\infty, x)$ or the collection of all intervals (a, b).

4.1 Continuous Generating Distribution

In order to construct the critical regions, we must study the distributions of the test statistics under the null hypothesis (all n points have the same generating distribution).

Our change-detection scheme can use the Wilcoxon, Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$ and Ξ_A statistics as well as any other statistic for testing if two samples have the same generating distribution. So let K represent the statistic being used. Pick one window pair and let m_1 be the size of its first window and m_2 be the size

of its second window. Over the first n points of the stream S, our change-detection scheme computes the values: $K(< s_1, \ldots, s_{m_1} >, < s_{i+m_1}, \ldots, s_{i+m_1+m_2} >)$ for $i = 1 \ldots n - m_1 - m_2$. In order for us *not* to reject the null hypothesis, none of these values can be in the critical region $\{x : x \geq \alpha\}$. Let $F_{K,m_1,m_2,n}(S)$ be the maximum of these values. Then we require $F_{K,m_1,m_2,n}(S) < \alpha$. It turns out that when the first n points have the same continuous generating distribution G then $F_{K,m_1,m_2,n}$ is a random variable whose distribution does not depend on G.

Theorem 4.1. If the first n points, $s_1, \ldots, s_n$, of the stream have continuous generating distribution G and the statistic K is either the Wilcoxon, Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$ or $\Xi_{\mathcal{A}}$ then the distribution of $F_{K,m_1,m_2,n}$ does not depend on G.

When $n = m_1 + m_2$ this is the same as testing if two samples have the same continuous generating distribution. In this case, this result for the Wilcoxon and Kolmogorov-Smirnov statistics is well-known.

Theorem 4.2. Under the hypothesis of 4.1, then for any c, $P(F_{K,m_1,m_2,n} > c)$ is $1/n!$ times the number of permutations of the stream $< 1, 2, \ldots, n >$ for which $F_{K,m_1,m_2,n} > c$.

Theorems 4.1 and 4.2 shows us how to construct a size (n, p) test for continuous distributions. Our null hypothesis is "the generating distribution of the data has not changed" and we reject the null hypothesis as soon as $K(< s_1, \ldots, s_{m_1}), < s_{i+m_1}, \ldots, s_{i+m_1+m+2} >) > \alpha$ (that is, as soon as the test statistic comparing the front and rear windows is large enough). Thus the critical region is $\{x : x > \alpha\}$ where α depends on n and p. There are three ways we can compute α:

1. Direction Computation: generate all $n!$ permutations of $< 1, 2, \ldots, n >$ and compute $F_{K,m_1,m_2,n}$. Set α to be the $1 - p$ percentile of the computed values.

2. Simulation: since the distribution $F_{K,m_1,m_2,n}$ does not depend on the generating distribution of the stream, choose some continuous distribution (such as the uniform between 0 and 1), generate ℓ samples of n points each, compute $F_{K,m_1,m_2,n}$ for each sample and take the $1 - p$ quantile. We will show how to choose ℓ in Subsection 4.2.

3. Sampling: since simulation essentially gives us ℓ permutations of $< 1, 2, \ldots, n >$, we can generate ℓ permutations directly, compute $F_{K,m_1,m_2,n}$ and take the $1-p$ quantile. This uses less random bits than the simulation approach since we don't need to generate random variables with many significant digits.

If we are dealing with discrete distributions and are using the Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$ or $\Xi_{\mathcal{A}}$ statistics, then Theorem 4.3 assures us that we can construct the critical region as above and the probability of falsely rejecting the null hypothesis is $\leq p$.

Theorem 4.3. Let G be a distribution function (G could be continuous, discrete, or a mixture of the two) and let H be a continuous distribution function. If K is either the Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$ or $\Xi_{\mathcal{A}}$ statistic, then for any $c \geq 0$, $P_G(F_{K,m_1,m_2,n} > c) \leq P_H(F_{K,m_1,m_2,n} > c)$

4.2 Choosing ℓ

In this section we discuss how to choose ℓ (the number of simulation runs we need to compute the $(1 - p)$ quantile). We have an unknown distribution G from which we sample ℓ points and use the element that falls in the $(1-p)$ quantile as an estimate of the true $1-p$ quantile. If the $1-p$ quantile is unattainable, then we actually compute an estimate of the $1 - p^*$ quantile where $1-p^*$ is the smallest attainable quantile $\geq 1-p$

So given constants L^* and U^* (where $L^* < 1-p < U^*$), and δ, we want to choose ℓ so that our estimate of the of the $1-p$ quantile is between L^* and U^* with probability $1 - \delta$. Let L to be the largest attainable quantile $\leq L^*$ and choose x_L such that $P_G(X \leq x_L) = L$. Similarly, let U be the smallest attainable quantile $\geq U^*$ and choose x_U such that $P_G(X \leq x_U) = U$.

Now let $X_1, \ldots, X_n$ be random variables with distribution G. Define the random variables $Y_1, \ldots, Y_n$ such that $Y_i = 1$ if $X_i \leq x_L$ and 0 otherwise. Define $Z_1, \ldots, Z_n$ so that $Z_i = 1$ if $X_i \leq x_U$ and 0 otherwise. Note that $P(Y_i = 1) = L$ and $P(Z_i = 1) = U$.

Suppose v is the element that falls in the $1 - p$ quantile of the X_i and let $\mu_v = P_G(X \leq v)$ be the true quantile of v. If $\mu_v < L$ then at least $n(1 - p)$ of the Y_i are 1 and if $\mu_v > U$ then at most $n(1 - p)$ of the Z_i are 1. Thus

$$\begin{aligned} P(\mu_v \notin [L, U]) &\leq P\left(\sum_{i=1}^{n} Y_i \geq n(1-p)\right) \\ &+ P\left(\sum_{i=1}^{n} Z_i \leq n(1-p)\right) \quad (3) \end{aligned}$$

Now, if $W_1, \ldots, W_n$ are i.i.d $0-1$ random variables with $P(W_i = 1) = \theta$ and $S_n = W_1 + \cdots + W_n$ then the following holds [10]:

$$P(S_n \leq k) = (n-k)\binom{n}{k} \int_0^{1-\theta} t^{n-k-1}(1-t)^k \, dt$$

The integral is known as the incomplete beta functions $I_x(a, b)$ where $x = 1 - \theta$, $a = n - k$ and $b = k + 1$. [20] shows how to numerically evaluate the incomplete beta function. Then it becomes a simple matter of binary search to find a value of n such that the right hand side of Equation 3 is $\leq \delta$.

5 Algorithms

In this section we will assume that the stream $S =< s_1, s_2, \cdots >$ consists of real-valued points and that $\mathcal{A}$ is either the collection of initial segments or intervals. Algorithms and suitable choices of $\mathcal{A}$ for higher dimensions is an open problem. Our algorithms use the following data structure:

Definition 5 (KS structure). We say that A is a KS structure if

- It is a finite array $< a_1, \ldots, a_m >$ of elements in $\mathbb{R}^2$ where the first coordinate is called the "value" and the second coordinate is called the "weight". The value is referred to as $v(a_i)$. The weight is referred to as $w(a_i)$.
- The array is sorted in increasing order by value.
- The length of the array is referred to as $|A|$.

For each integer k, we can define $G_A(k) = \sum_{i=1}^{k} w(A_i)$

Let (X, Y) be a window pair where X is the rear window and Y is the front window, $|X| = m_1$ and $|Y| = m_2$. We sort all the elements and create a KS-structure $Z =< z_1, z_2, \ldots, z_{m_1+m_2} >$ where $w(z_i) = -1/m_1$ if z_i came from x and $w(z_i) = 1/m_2$ if z_i came from Y. Z can be maintained throughout the life of the stream with incremental cost $O(\log(m_1 + m_2))$ by using a balanced tree.

Using this data structure, the Wilcoxon can be recomputed in time $O(m_1 + m_2)$. The same thing holds for $\phi_{\mathcal{A}}$ and $\Xi_{\mathcal{A}}$ when $\mathcal{A}$ is the set of initial segments. If $\mathcal{A}$ is the set of all intervals then the recomputation time for $\phi_{\mathcal{A}}$ and $\Xi_{\mathcal{A}}$ is $O([m_1 + m_2]^2)$. It is an open question whether it is possible to incrementally recompute those statistics faster.

In the rest of this section, we show how to recompute the Kolmogorov-Smirnov statistic over intervals and initial segments in $O(\log(m_1 + m_2))$ time. For intervals, this is the same as finding the a, b (with $a < b$) that maximize $|G_Z(b) - G_Z(a)|$. For initial segments we need to maximize $|G_Z(a)|$.

Lemma 5.1. Let Z be a KS-structure. Then

$$\max_{a<b} |G_Z(b) - G_Z(a)| = \max_c G_Z(c) - \min_d G_Z(d) \quad (4)$$

Thus it is sufficient to compute $\max_c G_Z(c)$ and $\min_d G_Z(d)$. The quantities of interest are $\max_c G_Z(c) - \min_d G_Z(d)$ (for intervals) and $\max\{\max_c G_Z(c), |\min_d G_Z(d)|\}$ (for initial segments). The next lemma forms the basis of the incremental algorithm.

Lemma 5.2. Let A and B be KS structures. Furthermore, $v(a) \leq v(b)$ for all $a \in A$, $b \in B$. Let M_A maximize G_A and m_A minimize G_A. Similarly let M_B maximize G_B and m_B minimize G_B. Let Z be the KS structure formed from the elements of a and b. Then either M_A or $M_B + |A|$ maximizes G_Z and either m_A or $m_B + |A|$ minimizes G_Z.

Algorithm 2 : START(X,Y)

1: For each $x \in X$ set weight$(x) = 1/|X|$
2: For each $y \in Y$ set weight$(y) = -1/|Y|$
3: Create the KS structure Z from X and Y (Z is sorted by value)
4: Create a binary tree B where the elements of Z are the leaves.
5: DESCEND(B.root)
6: Return B.root.VMax-B.root.vmin

Thus we can create a divide-and-conquer algorithm that maintains KS structures at every level and uses Lemma 5.2 to combine them. The algorithm sorts the elements in the windows X and Y into an array Z and builds a binary tree over it (where the elements of X and Y are contained in the leaves). For every node n, the set of leaves descended from n, referred to as $J(n)$, forms a consecutive subset of Z (we refer to this as a subarray). Thus if n_1 and n_2 are siblings then $J(n_1)$ and $J(n_2)$ are disjoint and the concatenation of $J(n_1)$ and $J(n_2)$ is a subarray of Z. Furthermore, each $J(n)$ is a KS structure. Each node n has the following 5 fields:

1. **sum**= sum of the weights of elements of $J(n)$.
2. **imin** = the integer that minimizes $G_{J(n)}$
3. **IMax** = the integer that maximizes $G_{J(n)}$
4. **vmin** = $G_{J(n)}(imin)$
5. **VMax** = $G_{J(n)}(IMax)$

The algorithm starts at the root. The general step is as follows: if we are examining node n and one of its children c does not have any values for its fields then we recurse down that child. Otherwise if both children have values for those fields, we use Lemma 5.2 to compute these values for n. Algorithms 2 and 3 show how this is done.

The algorithm performs one $O(|X| + |Y|)$ sorting step. Building a blank binary try over these elements can be done in $O(|X| + |Y|)$ time since there are $O(|X| + |Y|)$ nodes and for each node it computes the values of its fields in constant time. Therefore, after the sorting step, the algorithm runs in linear time.

To make this incremental, we note that when a new element arrives in the stream, we remove one element from the front window Y and then add this new element and the weights of the elements in X and Y do not change. Thus we just need to maintain the tree structure of the algorithm in $O(\log(|X| + |Y|))$ time under insertions and deletions. To do this, we replace the binary tree with a balanced tree, such as a B* tree. Now when a new element is inserted or deleted, we can follow the path this element takes from the root to a leaf. Only the nodes along this path are affected and so we can recursively recompute the fields values for those nodes in constant time per node (in a way similar to procedure DESCEND, shown in Algorithm 3). Since the both path length and insert/delete costs are

Algorithm 3 : DESCEND(n)

if n is a leaf **then**
  $a \leftarrow$ the element of Z contained in n
  n.sum$\leftarrow$ weight(a).
  if weight(a) > 0 **then**
    n.imin$\leftarrow 1$, n.IMax$\leftarrow 1$
    n.vmin$\leftarrow 0$, n.VMax$\leftarrow a$
  else
    n.imin$\leftarrow 1$, n.IMax$\leftarrow 1$
    n.vmin$\leftarrow a$, n.VMax$\leftarrow 0$
  end if
  return
end if
$lc \leftarrow$left_child(n), $rc \leftarrow$righ_child(n)
DESCEND(lc); DESCEND(rc)
n.sum$\leftarrow lc$.sum + rc.sum
if lc.VMax $\geq lc$.sum+rc.VMax **then**
  n.VMax$\leftarrow lc$.VMax, n.IMax$\leftarrow lc$.IMax
else
  n.VMax$\leftarrow lc$.sum + rc.VMax
  n.IMax$\leftarrow rc$.IMax + $|J(\text{lc})|$
end if
if lc.vmin $\leq lc$.sum+rc.vmin **then**
  n.vmin$\leftarrow lc$.vmin, n.imin$\leftarrow lc$.imin
else
  n.vmin$\leftarrow lc$.sum + rc.vmin
  n.imin$\leftarrow rc$.imin + $|J(\text{lc})|$
end if
return

Figure 1: Average number of errors in 2,000,000 points

size(n,p)	W	KS	KS (Int)	ϕ	Ξ
$\mathcal{S}$(20k, .05)	8	8	9.8	3.6	7.2
$\mathcal{S}$(50k, .05)	1.4	0.6	1.8	1.6	1.8

$O(\log(|X| + |Y|))$ the incremental algorithm runs in time $O(\log(|X| + |Y|))$.

6 Experimental Results

In order to compare the various statistics for nonparametric change detection, it is necessary to use simulated data so that the changes in generating distributions are known. In each experiment, we generate a stream of $2,000,000$ points and change the distribution every $20,000$ points. Note that the time at which a change is detected is a random variable depending on the old and new distributions. Thus the time between changes is intentionally long so that it would be easier to distinguish between late detections of change and false detections of change. Furthermore, in order to estimate the expected number of false detections, we run the change-detection scheme on 5 control streams with 2 million points each and no distribution change. Figure 1 reports the average number of errors per 2 million points.

In the experiments, our scheme uses 4 window pairs where both windows in a pair have the same size. The sizes are $200, 400, 800, 1600$ points. We evaluate our scheme using the Kolmogorov-Smirnov statistic over intial segments "KS", the Kolmogorov-Smirnov statistic over intervals "KSI", the Wilcoxon statistic "W", and the $\phi_{\mathcal{A}}$ and $\Xi_{\mathcal{A}}$ statistics (where $\mathcal{A}$ is the set of initial segments). We have two version of each experiment, each using a different critical region. The critical regions correspond to size $(50000, .05)$ and $(20000, .05)$. These are referred to as $\mathcal{S}(50k, .05)$ and $\mathcal{S}(20k, .05)$, respectively. The critical regions for each window were constructed by taking the .95 quantile over 500 simulation runs (using the uniform distribution between 0 and 1).

When some window detects a change, it is considered *not* late if the real change point is within the window or if the change point was contained in the window at most M time units ago (where M is the size of the window). Otherwise the change is considered late.

Distribution changes are created as follows: each stream starts with some distribution F with parameters $p_1, \ldots, p_n$ and rate of drift r. When it is time for a change, we choose a (continuous) uniform random variable R_i in $[-r, r]$ and add it to p_i, for all i.

The rest of the experiments deal with streams where the generating distribution changes (there are 99 true changes in each stream and a change occurs every 20,000 points). The numbers are reported as a/b where a is the number of change reports considered to be *not* late and b represents the number of change reports which are late or wrong. Note the average number of false reports should be around the same as in the control files.

The first group of experiments show what happens when changes occur primarily in areas with small probabilities. In Figure 2, the initial distribution is uniform on $[-p, p]$ and p varies at every change point. The changes are symmetric, and as expected, the Wilcoxon statistic performs the worst with almost no change detection. The Kolmogorov-Smirnov test primarily looks at probability changes that are located near the median and doesn't do very well although it clearly outperforms the median. In this case, performing the Kolmogorov-Smirnov test over intervals is clearly superior to initial segments. Clearly the best performance is obtained by the ϕ and Ξ statistics. For example, using the $\mathcal{S}(50k, .05)$ test for ϕ there are 86 on-time detections and 13 late detections. Since its error rate is about 1.6, it is very likely that this test truly detected all changes.

Figure 3 shows a more subtle change. The starting distribution is a mixture of a Standard Normal distribution with some Uniform noise (uniform over $[-7, 7]$). With probability p we sample from the Normal and with proability $1-p$ we sample from the Uniform. A change in generating distribution is obtained

Figure 2: Uniform on $[-p, p]$ ($p = 5$) with drift= 1

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	0/5	0/4
KS	31/30	25/15
KSI	60/34	52/27
ϕ	92/20	86/13
Ξ	86/19	85/9

Figure 3: Mixture of Standard Normal and Uniform[-7,7] ($p = 0.9$) with drift= 0.05

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	0/2	0/0
KS	0/15	0/7
KSI	4/32	2/9
ϕ	16/33	12/27
Ξ	13/36	12/18

Figure 4: Mixture of Standard Normal and Uniform[-7,7] ($p = 0.1$) with drift= 0.06

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	0/11	0/4
KS	0/19	3/14
KSI	13/32	9/24
ϕ	2/9	2/9
Ξ	5/32	5/9

Figure 8: Poisson ($\lambda = 50$) with drift = 1

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	36/35	31/26
KS	23/30	16/27
KSI	14/25	10/18
ϕ	14/21	9/17
Ξ	23/22	17/11

by varying p. Initially $p = .9$, meaning that the distribution is close to Normal. Here we have similar results. The Wilcoxon does not detect any changes and is clearly inferior to the Kolmogorov-Smirnov statistic. Once again, change detection improves when we consider intervals instead of initial segments. The ϕ and Ξ statistics again perform the best (with ϕ being slightly better than Ξ).

By setting $p = 0.1$ we get a distribution the is similar to the uniform, but with a small bulge at the median caused by the Standard Normal part of the mixture. This type of situation should favor the regular Kolmogorov-Smirnov test. However, Figure 4 shows that the best results are obtained by examing intervals and that the Kolmogorov-Smirnov test over intervals is clearly superior.

The next group of experiments investigates the effects of changing parameters of commonly used distributions. Figures 5 and 6 show results for Normal and Exponential distributions. The performance of the tests is similar, given the error rate for $\mathcal{S}(20k, .05)$ tests and so the $\mathcal{S}(50k, 0.5)$ tests are more informative. Overall, the Kolmogorov-Smirnov test does better, suggesting that such parametrized changes primarily affect areas near the median.

Finally, we show results discrete distributions. For all tests but the Wilcoxon, we showed that the error bounds from the continuous case are upper bounds on the error in the discrete case. Thus the results can indicate that some tests perform better in the discrete setting or that for some tests, bounds for discrete distributions are closer to the bounds for continuous distributions. However, it is not possible to distinguish between these two cases without more theoretical analysis. In the case of the Wilcoxon test, we do not know if the bounds for continuous distributions are upper bounds for discrete distributions. However, is we assume the same error rate as in Figure 1 we could compare the results. Figures 7 and 8 show our results for Binomial and Poisson distributions. The Wilcoxon appears to perform the best, both in early detection and total detection of change. However, it is difficult to judge the significance of this result. Among the other tests, the Kolmogorov-Smirnov test appears to be best.

7 Related Work

There is much related work on this topic. Some of the standard background includes statistical hypothesis testing and the multiple testing problem [5]. There has been much work on change point analysis in the statistics literature [6], However, most of the tests are parametric in nature (except the tests discussed in Section 1), and thus their assumptions are rarely satisfied for real data. Furthermore, the tests are run only once - after all of the data has been collected. The most related work from the statistics literature is the area of scan statistics [14, 15]. However, work on scan statistics does not work in the data stream model: the algorithms require that all the data can be stored in-memory, and that the tests are preformed only once after all the data is gathered. Neill and Moore improve the efficiency of Kulldorff's spatial scan statistics using a hierarchical tree structure [19].

In the database and data mining literature there is a plethora of work on processing data streams (see [3] for a recent survey). However, none of this work addresses the problem of change in a data stream. There is some work on evolving data [11, 12, 13, 7], mining evolving data streams [7, 17], and change detection in semistructured data [9, 8]. The focus[3] of that work, however, is detection of specialized types of change and not general definitions of detection of change in the underlying distribution. There has been recent work on frameworks for diagnosing changes in evolving data streams based on velocity density estimation [1] with the emphasis on heuristics to find trends, rather than formal statistical definitions of change and when change is statistically meaningful, the approach taken in this paper.

The work closest to ours is work by Kleinberg on

[3]No pun intended [13].

Figure 5: Normal ($\mu = 50$, $\sigma = 5$) with drift= 0.6

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	10/27	6/16
KS	17/30	9/27
KSI	16/47	10/26
ϕ	16/38	11/31
Ξ	17/43	16/22

Figure 6: Exponential ($\lambda = 1$) with drift= 0.1

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	12/38	6/34
KS	11/38	9/26
KSI	7/22	4/14
ϕ	7/29	5/18
Ξ	11/46	4/20

Figure 7: Binomial ($p = 0.1$, $n = 2000$) with drift= 0.001

St.	$\mathcal{S}$(20k,.05)	$\mathcal{S}$(50k,.05)
W	36/42	25/30
KS	24/38	20/26
KSI	17/22	13/15
ϕ	12/32	11/18
Ξ	23/33	15/23

the detection of word bursts in data stream, but his work is tightly coupled with the assumption of discrete distributions (such as the existence of words), and does not apply to continuous distributions [18].

8 Conclusions and Future Work

We believe that this is a promising first step towards non-parametric change detection that is suitable for data mining. The experiments imply something well known in the statistics community: there is no test that is "best" in all situations. However, the $\phi_{\mathcal{A}}$ and $\Xi_{\mathcal{A}}$ statistics do not perform much worse than the other statistics we tested, and in some cases they were vastly superior. So one direction for future research is to characterize the relative strengths and weaknesses of various non-parametric tests and to study the types of changes that occur in "real-life data." It would also be interesting to see how much can we relax the assumption that the points in the stream are generated independently. Other interesting directions for future work are improving bounds for discrete distributions, designing fast algorithms (especially for statistics computed over intervals), determining which classes of sets $\mathcal{A}$ are useful in higher dimensions, and better estimation of the point in time at which the change occurred.

References

[1] C. Aggarwal, J. Han, J. Wang, and P. S. Yu. A framework for clustering evolving data streams. In *VLDB 2003*.

[2] M. Anthony and J. Shawe-Taylor. A result of vapnik with applications. *Discrete Applied Mathematics*, 47(2):207–217, 1993.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS 2002*.

[4] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *FOCS 2000*.

[5] P. J. Bickel and K. Doksum. *Mathematical Statistics: Basic Ideas and Selected Topics*. Holden-Day, Inc., 1977.

[6] E. Carlstein, H.-G. Müller, and D. Siegmund, editors. *Change-point problems*. Institute of Mathematical Statistics, Hayward, California, 1994.

[7] S. Chakrabarti, S. Sarawagi, and B. Dom. Mining surprising patterns using temporal description length. In *VLDB 1998*.

[8] S. S. Chawathe, S. Abiteboul, and J. Widom. Representing and querying changes in semistructured data. In *ICDE 1998*.

[9] S. S. Chawathe and H. Garcia-Molina. Meaningful change detection in structured data. In *SIGMOD 1997*.

[10] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, inc., 3rd edition, 1970.

[11] V. Ganti, J. Gehrke, and R. Ramakrishnan. Demon: Mining and monitoring evolving data. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 13(1):50–63, 2001.

[12] V. Ganti, J. Gehrke, and R. Ramakrishnan. Mining data streams under block evolution. *SIGKDD Explorations*, 3(2):1–10, 2002.

[13] V. Ganti, J. Gehrke, R. Ramakrishnan, and W.-Y. Loh. A framework for measuring differences in data characteristics. *Journal of Computer and System Sciences (JCSS)*, 64(3):542–578, 2002.

[14] J. Glaz and N. Balakrishnan, editors. *Scan statistics and applications*. Birkhäuser Boston, 1999.

[15] J. Glaz, J. Naus, and S. Wallenstein. *Scan statistics*. Springer New York, 2001.

[16] T. Hagerup and C. Rub. A guided tour of chernoff bounds. *Information Processing Letters*, 33:305–308, 1990.

[17] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *KDD 2001*.

[18] J. M. Kleinberg. Bursty and hierarchical structure in streams. In *KDD 2002*.

[19] D. Neill and A. Moore. A fast multi-resolution method for detection of significant spatial overdenisities. Carnegie Mellon CSD Technical Report, June 2003.

[20] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C : The Art of Scientific Computing*. Cambridge University Press, 1992.

[21] V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, 1998.

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

A Proof of Theorems 4.1, 4.2 and 4.3

Proof of Theorems 4.1 and 4.2. Let G be the generating distribution. Since the distribution is continuous, all the s_i are distinct with probability 1. We can treat the first n points of S, $< s_1, \ldots, s_n >$, as an n-dimensional point $\overrightarrow{T}$ in $\mathbb{R}^n$ (where $s_i \in \mathbb{R}$ for all i). Let $\pi(\overrightarrow{T}) = \{\sigma_1(\overrightarrow{T}), \ldots, \sigma_{2^n}(\overrightarrow{T})\}$ be the set of all permutation of $\overrightarrow{T}$ (for example $< 3, 1, 2 >$ is a permutation of $< 1, 2, 3 >$). Since the first n points have the same generating distribution, each permutation of $\overrightarrow{T}$ is as equally likely as $\overrightarrow{T}$.

Now fix $\overrightarrow{T^*} \in \mathbb{R}^n$ and consider

$$P(F_{K,m_1,m_2,n}\left(\overrightarrow{T}\right) > c \;\middle|\; \overrightarrow{T} \in \pi(\overrightarrow{T^*})\Big) \tag{5}$$

Note this probability is independent of G since each permutation of $\overrightarrow{T^*}$ is equally likely. Since none of the tests (Wilcoxon, Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$,$\Xi_{\mathcal{A}}$) depend on the values of the points (they only depend on their relative order), this conditional probability is the same as

$$P\left(F_{K,m_1,m_2,n}(\overrightarrow{T}) > c \;\middle|\; \overrightarrow{T} \in \pi(< 1, 2, 3, \ldots, n >)\right)$$

This, in turn equals $(n!)^{-1}$ times the number of permutations of $< 1, 2, 3, \ldots, n >$ for which $F_{K,m_1,m_2,n} > c$. Thus Equation 5 is independent of T^*, therefore it equals $P_G(F_{K,m_1,m_2,n}(\overrightarrow{T}) > c)$, and so is independent of the generating distribution of the first n points. □

Proof of Theorem 4.3. Fix a $c \geq 0$. As usual we can represent the first n elements of a stream as an n-dimensional vector $\overrightarrow{T} =< s_1, s_2, \ldots, s_n >$ in $\mathbb{R}^n$. If G is the generating distribution, then not all s_i have to be distinct. Once again we will take conditional expectations and the theorem follows once we prove (for all $\overrightarrow{T^*} \in \mathbb{R}^n$:

$$P_G(F_{K,m_1,m_2,n}(\overrightarrow{T_1}) > c \mid T_1 \in \pi(\overrightarrow{T^*})) \leq P_H(F_{K,m_1,m_2,n}(\overrightarrow{T_2}) > c) \tag{6}$$

Let $t_1, \ldots, t_k$ be the distinct values the appear as coordinates of $\overrightarrow{T^*} =< s_1, \ldots, s_n >$ and let d_i be the number of times that t_i appears as a coordinate of $\overrightarrow{T^*}$. Each permutation of $\overrightarrow{T^*}$ is still equally likely, but now there are only $n!/\prod_{i=1}^{k} d_i$ such permutations instead of $n!$ permutations. To show that Equation 6 holds, it is enough to show that for every permutation $\sigma_i(\overrightarrow{T^*})$ of $\overrightarrow{T^*}$ such that $F_{K,m_1,m_2,n}(\sigma_i(\overrightarrow{T^*})) > c$ there are $D = \prod_{i=1}^{k} d_i$ permutations $\sigma^*_{i,1}, \sigma^*_{i,2}, \ldots, \sigma^*_{i,D}$ of $< 1, 2, 3, \ldots, n >$ such that $F_{K,m_1,m_2,n}(\sigma^*_{i,j}) > c$ (for $j = 1, \ldots, D$) and all the $\sigma^*_{i,j}$ are distinct for all i, j.

First we partition the set $\{1, 2, \ldots, n\}$ into k partitions (where k is the number of distinct elements in $\overrightarrow{T^*}$) such that the first d_1 numbers go into partition t^*_1, then next d_2 numbers go into t^*_2, etc, so that partition t^*_i has d_i elements. Now pick a permutation $\sigma_i(\overrightarrow{T*})$ such that $F_{K,m_1,m_2,n}(\sigma_i(\overrightarrow{T^*})) > c$. We form the permutations $\sigma^*_{i,1}, \ldots, \sigma^*_{1,D}$ by examining $\sigma_i(\overrightarrow{T^*})$. For each r ($r = 1, \ldots, n$) we look at the r^{th} coordinate of $\sigma_i(\overrightarrow{T^*})$. Say this value is t_j. Then we choose some value from the partition t^*_j put put this value in the r^{th} coordinate of our new permutation (making sure we don't use the same once per new permutation). It is clear that this will really give us $D = \prod_{i=1}^{k} d_i$ distinct permutations for each $\sigma_i(\overrightarrow{T^*})$. Also, if $\sigma_i(\overrightarrow{T^*})$ and $\sigma_j(\overrightarrow{T^*})$ are distinct permutatons, then it is also clear that $\sigma^*_{i,1}, \ldots, \sigma^*_{i,D}, \sigma_{j,1}, \ldots, \sigma^*_{j,D}$ are all distinct. Thus we just have to show that if $F_{K,m_1,m_2,n}(\sigma_i(\overrightarrow{T^*})) > c$ then $F_{K,m_1,m_2,n}(\sigma^*_{i,j}) > c$ for all i, j.

Note that the Kolmogorov-Smirnov, $\phi_{\mathcal{A}}$ and $\Xi_{\mathcal{A}}$ statistics all have the form

$$\sup_{C \in \mathcal{A}} f\left(\frac{|X \cap C|}{|X|}, \frac{|Y \cap C|}{|Y|}\right)$$

. So if $F_{K,m_1,m_2,n}(\sigma_i(\overrightarrow{T^*})) > c$ then there is some set $C \in \mathcal{A}$ and some integer r such that $f(|X \cap C|/m_1, |Y \cap C|/m_2) > c$ where $X = \{s_1, \ldots, s_{m_1}\}$ and $Y = \{s_{r+m_1}, \ldots, s_{r+m_1+m_2}\}$. As a result of our construction of the permutations $\sigma^*_{i,j}$, and by conditions on $\mathcal{A}$, there exists a C' such that for each $\sigma^*_{i,j} =< s^{(i,j)}_1, \ldots, s^{(i,j)}_n >$, $f(|X' \cap C'|/m_1, |Y' \cap C'|/m_2) > c$ where $X' = \{s^{(i,j)}_1, \ldots, s^{(i,j)}_{m_1}\}$ and $Y' = \{s^{(i,j)}_{r+m_1}, \ldots, s^{(i,j)}_{r+m_1+m_2}\}$. This completes the proof. □

B Relativized Convergence as a metric

It turns out that $|\phi|$ is also a metric on the space of distribution functions. Let I be the unit interval $[0, 1]$. We define the following functions from $I \times I \to \mathbb{R}$:

$$\begin{aligned} f(x,y) &= \begin{cases} \frac{x-y}{\sqrt{\frac{x+y}{2}}} & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \\ g(x,y) &= \begin{cases} \frac{x-y}{\sqrt{1-\frac{x+y}{2}}} & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \\ \psi(x,y) &= \max\{f(x,y), g(x,y)\} \end{aligned}$$

The main goal of this section is to show that $|f|, |g|, |\psi|$ are metrics over the unit interval. Simple analysis shows that f, g and ψ are continuous.

Lemma B.1. For any $x, y \in I$, the $f(x, y)/\sqrt{2}$ is increasing in x and decreasing in y.

Proof. It is increasing in x only if the partial derivative with respect to x is positive on $(0,1)$.

$$\begin{aligned}\frac{\partial f(x,y)/\sqrt{2}}{\partial x} &= \frac{1}{\sqrt{x+y}} - \frac{x-y}{2(x+y)^{3/2}} \\ &= \frac{x+3y}{2(x+y)^{3/2}} > 0\end{aligned}$$

for $0 < x < 1$ since $y \geq 0$. By symmetry, it is decreasing in y. □

Corollary 1. For $x, y \in I$, $g(x,y)$ and $\psi(x,y)$ are increasing in x and decreasing in y.

Proof. Note $g(x,y) = f(1-y, 1-x) = -f(1-x, 1-y)$. The latter is decreasing in $1-x$ (by Lemma B.1) and so is increasing in x. Thus $g(x,y)$ is increasing in x and by symmetry it is decreasing in y. The claim for ψ follows immediately from the definition of ψ. □

Theorem B.1. $|f(x,y)|$ is a metric on $I = [0,1]$.

Proof. Clearly $|f|$ is symmetric in x and y and $|f(x,y)| = 0$ if and only if $x = y$. All that remains is to show that the triangle inequality holds. Choose x, y, z from I. We will show that $|f(x,z)| \leq |f(x,y)| + |f(y,z)|$. Without loss of generality, we may assume that x, y and z are all distinct. Furthermore, since $|f|$ is symmetric in its arguments, we may assume that $x > z$. We have the following cases:

Case 1: $y > x > z$. In this case $|f(x,z)| = f(x,z) \leq f(y,z) \leq |f(x,y)| + |f(y,z)|$. The first inequality follows from Lemma B.1.

Case 2: $x > z > y$. In this case $|f(x,z)| = f(x,z) \leq f(x,y) \leq |f(x,y)| + |f(y,z)|$. The inequality follows from Lemma B.1.

Case 3: $x > y > z$. We must show that $f(x,z) \leq f(x,y) + f(y,z)$. So let $q(x) = 2^{-1/2}(f(x,y) + f(y,z) - f(x,z))$. Our goal is to show that $q(x) \geq 0$ when $x > y > z$. Note that if $x = y > z$ then $q(x) = 0$. Thus we can show that $q(x) \geq 0$ (when $x > y > z$) by showing that $q'(x) > 0$ when $y < x < 1$.

$$\begin{aligned}q'(x) &= \frac{\partial f(x,y)/\sqrt{2}}{\partial x} - \frac{\partial f(x,z)/\sqrt{2}}{\partial x} \\ &= \frac{x+3y}{2(x+y)^{3/2}} - \frac{x+3z}{2(x+z)^{3/2}}\end{aligned}$$

Since $y > z$, we show that $q'(x) > 0$ by showing that $\frac{x+3y}{2(x+y)^{3/2}}$ is increasing in y on the interval $(0,x)$ and we do that by showing that the partial derivative with respect to y is positive on the interval $(0,x)$

$$\begin{aligned}\frac{\partial \frac{x+3y}{2(x+y)^{3/2}}}{\partial y} &= \frac{3}{2(x+y)^{3/2}} - \frac{3(x+3y)}{4(x+y)^{5/2}} \\ &= \frac{3x-3y}{4(x+y)^{5/2}} > 0\end{aligned}$$

since $x > y$. □

Corollary 2. $|g(x,y)|$ and $|\psi(x,y)|$ are metrics on $[0,1]$.

Proof.

$$\begin{aligned}|g(x,z)| &= |f(1-z, 1-x)| \\ &\leq |f(1-z, 1-y)| + |f(1-y, 1-x)| \\ &= |g(y,z)| + |g(x,y)|\end{aligned}$$

Since $|\psi(x,y)| = \max(|f(x,y)|, |g(x,y)|)$, the fact that $|\psi|$ is a metric then follows from Theorem B.1. □

From this it clearly follows that

Theorem B.2. The functional $\phi(F_1(x), F_2(x)) = \sup_x |\psi(F_1(x), F_2(x))|$ is a metric on the space of distribution functions.

C Proof of Lemmas 5.1 and 5.2

Proof of Lemma 5.1. Let a, b maximize the left hand side of Equation 4. First assume $G_Z(b) - G_Z(a) > 0$ and b does not maximize $G_{(X,Y)}$ (the case when a does not minimize it is symmetric). Let c be a point that maximizes G_Z. Clearly $c \neq a$. If $c > a$ then we have a contradiction since $G_Z(c) - G_Z(a) > G_Z(b) - G_Z(a) > 0$. If $c < a$ then we also have a contradiction since $|G_Z(a) - G_Z(c)| = G_Z(c) - G_Z(a) > G_Z(b) - G_Z(a) > 0$. Therefore b must maximize G_Z and by symmetry a must minimize it.

If $G_Z(b) - G_Z(a) < 0$ then replace "maximize" with "minimize" and vice-versa in the previous paragraph. □

Proof of Lemma 5.2. Suppose neither M_A nor M_B maximize G_Z. Then choose a j that maximizes G_Z. If $j \leq |A|$ then clearly j maximizes G_A and $G_A(j) > G_A(M_A)$. This is a contradiction. So suppose $j > |A|$. Then $G_Z(j) = G_A(|A|) + G_B(j - |A|)$ and $G_Z(M_B + |A|) = G_A(|A|) + G_B(M_B)$. Thus $G_Z(j - |A|) > G_Z(M_B + |A|)$ implies $G_B(j - |A|) > G_B(M_B)$ which is another contradiction. The case for m_A and m_B is symmetric. □