Proceedings of the 2017 Winter Simulation Conference
W. K. V. Chan, A. D'Ambrogio, G. Zacharewicz, N. Mustafee, G. Wainer, and E. Page, eds.

DATA DRIVEN STOCHASTIC APPROXIMATION FOR CHANGE DETECTION

Thomas Flynn

Department of Computer Science
The Graduate Center, CUNY
365 5th Ave.
New York, NY 10016, USA

Olympia Hadjiliadis

Department of Mathematics and Statistics
Hunter College, CUNY
695 Park Ave.
New York, NY 10065, USA

Ioannis Stamos
Felisa J. Vázquez-Abad

Department of Computer Science
Hunter College, CUNY
695 Park Ave.
New York, NY 10065, USA

ABSTRACT

Online change detection has many applications, ranging from finance and manufacturing, to security and computer vision. Designing a change detector for use in a given domain can be very time consuming, and model-based algorithms often require knowledge of the underlying stochastic model. To address these issues, in this work we explore a supervised learning approach to a change detector. We implement a gradient based procedure to find the optimal parameters for a change detector. We demonstrate the methodology on both synthetic and real world data for classifying 3D laser range image data in real-time.

1 INTRODUCTION

In this work we investigate a supervised learning approach to the problem of statistical change detection. A change detector is any algorithm that monitors a stream of observations and raises an alarm when it believes a change has occurred in the process generating the data. There is a wide literature on the theory of change detection, including various results on what are the best change detection algorithms to use, given different assumptions on the data generating processes and different notions of optimality (Poor and Hadjiliadis 2009). These algorithms use statistical hypothesis testing and therefore require knowledge of the underlying distribution of the processes. In many applications of interest, it may be extremely difficult to characterize the various distributions involved. This leads us to consider a data driven approach, wherein we use a body of ground truth data to train the algorithm in order to find the best change detector algorithm in a family of parameterized change detectors.

As a motivating example, consider a problem from 3D computer vision. A laser scanner delivers a stream $X_1, X_2, \ldots$ of 3D points as it sweeps across the scene and the goal is to detect when the surface being scanned changes from one class to another. For example, in an urban setting one encounters surfaces such as roads, vehicles, and building facades. Not only does one want to classify the points into each class but in real-time applications, such as robotics or self-driving cars, this classification must happen on the fly as points are acquired. This problem motivated our work in (Flynn, Hadjiliadis, and Stamos 2015), (Stamos

978-1-5386-3428-8/17/$31.00 ©2017 IEEE

et al. 2012), (Hadjiliadis and Stamos 2010). In (Flynn, Hadjiliadis, and Stamos 2015) we used a change detection framework based on detailed stochastic models for the different categories, such as vegetation, building facade, road, vehicle. The method requires manually building the statistics for hypothesis testing assuming knowledge of the distributions of the points for each class. Going beyond, we envision now a self-tuned change detector algorithm that can be used ubiquitously. Based on the observation that change detection algorithms are defined by a specific hypothesis test, we propose to write the test statistics in general form using a polynomial function of the observed values. Its coefficients are the parameters of our model, to be optimized using gradient based stochastic approximation. Ground truth data is used to train the algorithm, which will then automatically produce the test that works best for the data, and we do not assume knowledge of the underlying distributions.

We begin in Section 2 with some background on change detection and introduce our supervised learning approach in Section 3. Our system has three components: (1) The model, which comprises a change detection algorithm depending on some parameters, (2) the objective, which gives a performance measure to the change detector, and (3) the optimization algorithm, that considers an approximation to the problem that makes gradient estimation efficient. The algorithm takes examples of marked sequences and uses gradient descent to approximate the optimal parameters. In Section 4 we show the results of experiments using synthetic and real world data. In the synthetic case we test our approach on the classical problem of identifying a change of mean. Then we present an example from 3D computer vision, where a laser scanner delivers a stream of 3D points and the change detector must determine when the surface being scanned changes from non-vegetation to vegetation.

2 CHANGE DETECTION

Let $t \in \mathbb{N}$ denote the (true, unknown) time of change in regime. In the simplest setting, the observed process $\{X_t\}$ is assumed to satisfy: $\{X_i, i < t\}$ are independent and identically distributed (iid) with distribution Q_0, and $\{X_i, i \geq t\}$ are also iid but with a different distribution Q_1. In more general models Q_0 and Q_1 are measures governing the stochastic processes before and after the regime change.

The change detector is an algorithm that makes a judgment about whether the change has occurred at time k based on the observations $X_1, \ldots, X_k$. The result of the test is a stopping-time T, and there are various performance criteria one may be interested in, such as the probability of early detection events $\{T < t\}$, expectation of the delay $(T-t)^+$, or a two-sided error like $|T-t|$. A well-studied change detector is the CUSUM algorithm (Page 1954). It is assumed that the measures are mutually absolutely continuous, so that the likelihood ratio is well defined. Then T is defined using the likelihood ratios, a threshold h, and an auxiliary real-valued process $\{Z_k\}$ as follows.

$$Z_0 = 0, \quad Z_k = \max\left(0, Z_{k-1} + \log \tfrac{dQ_1}{dQ_0}(X_k)\right), \tag{1a}$$

$$T = \min\{k \geq 0 : \, Z_k \geq h\}. \tag{1b}$$

The sequence Z_k is known as the *CUSUM statistic process*. For a certain value of the threshold h, the CUSUM rule just described is the optimal stopping rule for minimizing worst-case detection delay $(T-y)^+$ subject to a lower-bound constraint on the mean time to the first false alarm $\mathbb{E}[T \,|\, t > T]$ (Poor and Hadjiliadis 2009). Roughly speaking, the CUSUM has the property that, for any desired value of γ, there is a corresponding value of the threshold h^* such that the CUSUM with threshold h^* has optimal detection delay while satisfying $\mathbb{E}[T \,|\, t > T] \geq \gamma$.

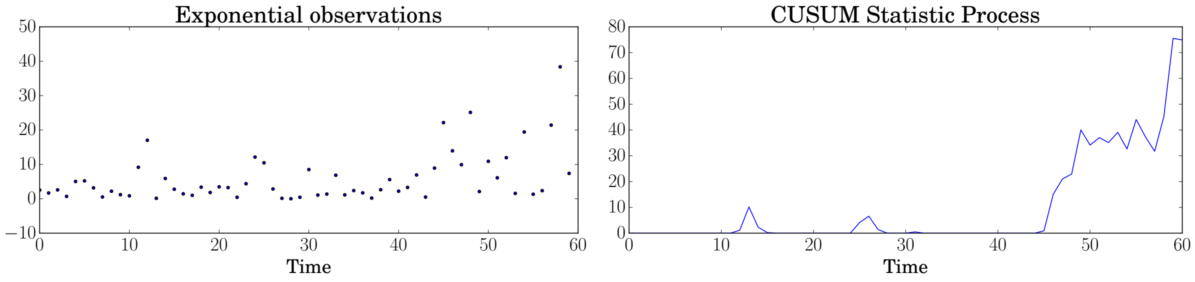
Example 1 Let $Q_0 \sim \mathcal{N}(0,1)$ be a normal distribution and $Q_1 \sim \mathcal{N}(\mu,1)$. According to (1a)

$$Z_0 = 0, \quad Z_k = \max\left(0, Z_{k-1} + \mu X_k - \tfrac{1}{2}\mu^2\right). \tag{2}$$

Example 2 Say that Q_0 and Q_1 are exponential distributions, with means $\mu_0 < \mu_1$. Here

$$Z_0 = 0, \quad Z_k = \max\left(0, Z_{k-1} + \left[\tfrac{1}{\mu_0} - \tfrac{1}{\mu_1}\right] X_k + \log \tfrac{\mu_0}{\mu_1}\right). \tag{3}$$

Fig. 1a shows a sample sequence of observations $\{X_t\}$ where $t = 40$, $\mu_0 = 3$ and $\mu_1 = 13$. Next to that Fig. 1b shows the process Z_k used in the CUSUM algorithm, as defined in (3). Small values of h detect very quickly, but they may give rise to false detection. Large values will increase the mean time to the first false alarm, but also the average delay.



(a)

(b)

Figure 1: (a) A typical sequence of exponential random variables that have a mean $\mu_0 = 3$ until $t = 40$ when they switch to $\mu_1 = 13$. (b) The corresponding sequence Z_k used in the CUSUM procedure.

Example 3 Let Q_0 and Q_1 be Bernoulli distributions, with means p_0 and p_1 respectively. Then

$$Z_k = \max\left(0, Z_{k-1} + \left[\log \tfrac{p_1(1-p_0)}{p_0(1-p_1)}\right] X_k + \log \tfrac{1-p_1}{1-p_0}\right). \tag{4}$$

For the choice of $p_0 = 0.2$ and $p_1 = 0.8$ this leads to $Z_k = \max(0, Z_{k-1} + (\log 16)X_k + \log \frac{1}{4})$. In words, an observation of a "success" increases the CUSUM statistic by $\log 4$, while an observation of a failure decreases the CUSUM statistic the same quantity.

To gain insight on the behavior of the test, an important observation is that the process $\{Z_k\}$ is a non-negative super-martingale under Q_0 and a sub-martingale under Q_1. That is, the expected drift is strictly negative (positive) before (after) t, so that

$$\mathbb{E}^{Q_0}[Z_{k+1} - Z_k \,|\, Z_k] \leq 0, \text{ and } \mathbb{E}^{Q_1}[Z_{k+1} - Z_k \,|\, Z_k] > 0. \tag{5}$$

Because of this, the CUSUM statistic process will hit zero with high probability before the regime change. However, after the change the drift is positive, enabling detection.

Note however that implementing the CUSUM requires evaluation of the likelihood ratio. Furthermore, while a very compelling feature of the CUSUM is that the rule depends only on the difference between the two probability measures, it remains the case that in many real world scenarios, this cannot be formally specified. Even if the likelihood ratio can be evaluated there is still the matter of calibrating the threshold h, which appears to be a non-trivial matter. These considerations lead us to consider a data driven approach, where we optimize the parameters of the change detection procedure in a supervised-learning framework.

3 PROPOSED FRAMEWORK

We will define a family of change detectors that depend on a parameter $\theta \in \mathbb{R}^d$ and then use a data set of ground truth together with gradient descent to optimize that parameter.

3.1 Model

Let $(\Omega, \mathfrak{F}, \mathbb{P})$ be a probability space and let t be a random variable on $(\Omega, \mathfrak{F})$. Consider a n-dimensional process $\{X_k; k = 1, 2, \ldots\}$ on $(\Omega, \mathfrak{F})$ such that $X_k \sim Q^0$ for $k < t$ and $X_k \sim Q^1$ for $k \geq t$. Denote by $\mathfrak{F}_k = \sigma(X_1, \ldots, X_k)$ the σ-field generated by the "history" of the process up to time k. The computer vision situations that motivate this work justify the assumption that the *change point t* is independent of $\mathfrak{F}_k, k < t$ and it has finite support $\{1,\dots,N+1\}$. A scanner has only N points in every direction that it observes. Detecting a change means detecting a change, say, from street to car, from car to trees, etc. Naturally there are situations where a scan line does not present any changes. This is represented by the event $\{t = N+1\}$.

Definition. For $\theta \in \mathbb{R}^d$, let $Z_0(\theta) = 0$ be the *generalized CUSUM process* is

$$Z_k(\theta) = \max(0, Z_{k-1}(\theta) + g(X_k,\theta)), \tag{6}$$

where $g\colon \mathbb{R}^n \times \mathbb{R}^d \to \mathbb{R}$ satisfies:

(a). For any $x \in \mathbb{R}^n$ $g(x,\theta)$ is a linear function of $\theta \in \mathbb{R}^d$. That is, g has the general form $g(x,\theta) = \theta_0 + \theta_1\gamma_1(x) + \theta_2\gamma_2(x) + \cdots + \theta_d\gamma_d(x)$, for some functions $\gamma_j(x); j = 1,\dots,d$.
(b). The random variable $Y(X) = \nabla_\theta g(X,\cdot)$ has uniformly bounded variance under both Q^0 and Q^1.
(c). There is a non-empty compact set $\Theta \subset \mathbb{R}^d$ such that for any $\theta \in \Theta$

$$\mathbb{E}^{Q_0}[g(X,\theta)] < 0, \quad \mathbb{E}^{Q_1}[g(X,\theta)] > 0. \tag{7}$$

Because of linearity in θ, the vector $Y(X)$ is independent of θ and Assumption (b) holds if its variance is bounded under both measures. From Assumption (c) it follows that the real-valued process $\{Z_k(\theta)\}$ satisfies (5). For this reason, when $\theta \in \Theta$, stopping rules of the form (1b) have high probability of detecting the change and low probability for false detection, depending on the value of h.

The examples above are all particular cases of our model, obtained by choosing $\theta \in \mathbb{R}^2$ in terms of the (unknown) parameters such as mean and variance of Q_0 and Q_1. For instance, Example 1 can be recovered in our model, using $g(x,(\theta_1,\theta_2)) = \theta_1 x + \theta_2$, with $\theta_1 = \mu$ and $\theta_2 = -\frac{1}{2}\mu^2$.

Using (6) yields a parameterized change detector. Our idea is to tune the parameter θ using gradient-search. Before stating the optimization problem we provide a result that justifies the use of stochastic gradients in this case.

Lemma 1 Let τ be a finite stopping time (w.p..1) with $\sup_\theta \mathbb{E}[\tau] < \infty$ adapted to the generalized CUSUM process. For any continuously differentiable function $\phi\colon \mathbb{R} \to \mathbb{R}$ with $\mathbb{E}[\phi(Z_\tau(\theta))] < \infty$, $\nabla_\theta \mathbb{E}[\phi(Z_\tau(\theta))] = \mathbb{E}\left[\phi'(Z_\tau(\theta))\nabla_\theta Z_\tau(\theta)\right]$, provided that the above expectations are finite.

Proof. Let $\theta \in \mathbb{R}^d$ be given. Because ϕ is continuously differentiable, it suffices to show that $Z_\tau(\theta)$ is Lipschitz continuous w.p.1, because $\phi(Z_\tau(\theta))$ would then be Lipschitz continuous w.p.1 which ensures that we can interchange derivative and expectation. It can be shown that if $|\nabla_\theta Z_\tau(\theta)|$ has finite expectation independent of θ then $Z_\tau(\theta)$ is Lipschitz continuous w.p.1.

By definition, $\nabla_\theta Z_\tau(\theta)$ follows from the recursion

$$\nabla_\theta Z_k(\theta) = \begin{cases} 0 & \text{if}\, Z_k(\theta) = 0 \\ \nabla_\theta Z_{k-1}(\theta) + \nabla_\theta g(X_k,\theta) & \text{otherwise,} \end{cases}$$

so that, for any index K, $|\nabla_\theta Z_K(\theta)| \le \sum_{k=1}^K |\nabla_\theta g(X_k,\theta)| = \sum_{k=1}^K |Y(X_k)|$. By Assumption (b) it follows that there is a constant C such that for each k, $\mathbb{E}[|Y(X_k)|] \le C$. Using Wald's equation for random stopping times, $\mathbb{E}|\nabla_\theta Z_\tau(\theta)| \le \mathbb{E}\left[\sum_{k=1}^\tau |Y(X_k)|\right] \le \mathbb{E}[\tau]C < \infty$. The quantity $\phi'(Z_\tau)\nabla_\theta Z_\tau(\theta)$ is the *stochastic gradient.* □

As will be stated in the following section, the optimization seeks to minimize an error function that penalizes the absolute error $|T(\theta) - t|$. Unfortunately, the stopping time $T(\theta)$ as defined in (1b) is not differentiable. It can be expressed in terms of the indicator functions

$$\mathbf{1}_{\{Z_k(\theta)\ge h\}} = \begin{cases} 1 & \text{if } Z_k(\theta) \ge h \\ 0 & \text{otherwise.} \end{cases}$$

To address this problem we use a mollifier technique. Some theoretical properties of approximation by mollifiers were investigated by Andrieu, Cohen, and Vázquez-Abad (2011). Instead of the indicator function we use the sigmoid $\sigma(x) = 1/(1+e^{-x})$, following the approach taken in neural networks such as the Boltzmann machine (Hinton and Sejnowski 1983). This results in a randomized decision: for each $\omega \in \Omega$, the sequence of observations $X_1, X_2 \dots$ specifies a distribution of the change detector $\tilde{T}(\theta)$:

$$Z_0(\theta) = 0, \quad Z_k(\theta) = \max(0, Z_{k-1}(\theta) + g(X_k, \theta)), \tag{8a}$$

$$\mathbb{P}[\tilde{T}(\theta) = k \,|\, \mathfrak{F}_k, \tilde{T}(\theta) \geq k] = \begin{cases} \sigma_h(Z_k(\theta)) & k \leq N, \\ 1 & k = N+1, \end{cases} \tag{8b}$$

where and $\sigma_h(z) = \sigma(z-h)$. Comparing this with (1b), the difference is that instead of using a hard threshold h to determine the rejection region of the test, a randomized procedure is used to either accept or reject a regime change. It follows that for all $k \leq N$

$$p_k(Z, \theta) \stackrel{\text{def}}{=} \mathbb{P}[\tilde{T}(\theta) = k \,|\, \mathfrak{F}_k] = \sigma_h(Z_k(\theta)) \prod_{\ell=1}^{k-1} (1 - \sigma_h(Z_\ell(\theta))) \tag{9}$$

and $p_{N+1}(Z, \theta) = \mathbb{P}[\tilde{T}(\theta) = N+1 \,|\, \mathfrak{F}_N] = 1 - \sum_{k=1}^{N} p_k(Z, \theta)$. Piecewise differentiability of $Z_k(\theta)$ and randomness in $\tilde{T}(\theta)$, make it straightforward to apply gradient based optimization, as we will see below.

3.2 Optimization Problem

Let $L\colon \mathbb{R} \to \mathbb{R}^+$ be a convex loss function such that $L(0) = 0$ and $L(x) \to \infty$ when $|x| \to \infty$ that represents a penalty for a detection error. Loss functions are usually symmetric. One reason to weigh equally early and late detection is that in our application, consecutive observations refer to adjacent spatial points rather than time. In our case L is the absolute value function $L(x) = |x|$ (used by Karatzas (2003)).

Proposition 3.1 The change detector with $Z_0(\theta^*) = 0$ and

$$Z_k(\theta^*) = \max(0, Z_{k-1}(\theta^*) + g(X_k, \theta^*)), \tag{10a}$$

$$T(\theta^*) = \min(k\colon Z_k(\theta^*) \geq h) \tag{10b}$$

where θ^* solves the optimization problem $\min_\theta \mathbb{E}[L(T(\theta) - t)]$ is independent of the threshold value h.

Proof. The proof follows from a scaling argument, using the form of the function g in Assumption (a). Multiplying the vector θ by a constant c will yield the scaled process $Z_k(c\theta) = cZ_k(\theta)$ for every trajectory ω. Therefore, such scaling is equivalent to a scaled threshold ch. Because we are optimizing the loss function, the optimal parameter θ^* will also adjust the value of h that will yield equivalent (albeit scaled) processes. At the end, one may always use the optimal solution θ^* divided by h and a unit threshold. □

Ideally we would directly optimize the change detector based on hard thresholds. That means solving the the problem

$$\min_{\theta \in \mathbb{R}^d} \mathbb{E}[L(T(\theta) - t)]. \tag{11}$$

For purposes of computation this is not feasible. In order to use a gradient-based method we consider the approximate optimization problem

$$\min_{\theta \in \mathbb{R}^d} \left(J(\theta) \stackrel{\text{def}}{=} \mathbb{E}[L(\tilde{T}(\theta) - t)] \right). \tag{12}$$

Proposition 3.2 Let $\tilde{T}(\theta)$ be defined as in (8b). If the functions $\gamma_j(\cdot); j = 1, \dots, d$ are non-negative, then $J(\theta)$ is a continuously differentiable quasiconvex function.

Proof. It is straightforward to show that $Z_k(\theta)$ is a.s-monotone non-decreasing in each component of the parameter θ. By Assumption (a), $g(\theta)$ is linear. Given $\omega \in \Omega$ the corresponding trajectory of $\{\gamma_i(X_k(\omega))\}$ for each i is assumed non-negative, so an increase in θ_j will increase the value of $g(X(\omega),\theta)$. Using mathematical induction, we conclude that $Z_k(\theta)$ in (8a) is also monotone non-decreasing in each θ_j, implying that $\tilde{T}(\theta)$ is monotone non-increasing. This implies that $L(\tilde{T}(\theta)-t)$ is quasiconvex in θ and so is its expectation $J(\theta)$. To show that it is continuously differentiable, notice that

$$\mathbb{E}[L(\tilde{T}(\theta)-t)\,|\,t] = \sum_{k=1}^{N+1} L(k-t)\,\mathbb{P}[\tilde{T}(\theta)=k\,|\,t] = \mathbb{E}\left[\sum_{k=1}^{N+1} L(k-t)\,p_k(Z,\theta)\,|\,t\right]$$

where we have used conditioning first on the value of $\tilde{T}(\theta)$ and then in $\mathbb{P}[\tilde{T}(\theta)=k\,|\,X_1,X_2,\ldots]$. The claim now follows from differentiability of p_k w.r.t. θ, given in (9) and the fact that t is a random variable independent of θ. □

Using this result, the optimization problem to solve is given by:

$$\min_{\theta\in\mathbb{R}^d}\left(\mathbb{E}\Big[\sum_{k=1}^{N} L(k-t)\,\sigma_h(Z_k(\theta))\prod_{\ell=1}^{k-1}(1-\sigma_h(Z_\ell(\theta)))\Big] + \mathbb{E}[L(N+1-t)\,p_{N+1}(Z,\theta)]\right). \tag{13}$$

It follows from Lemma 1 that the gradient of the function in (12) can be estimated using the stochastic gradient, because the $\sigma(\cdot)$ is continuously differentiable, so we can interchange derivative and expectation.

The difference between the *control problem* and the *optimization problem* is subtle, but important. When a change detector is in use, the algorithm should be designed to solve the original control problem that detects the change in regime. In contrast, the optimization problem (12) is the problem to be solved when the detecting machine is being trained to learn the optimal parameters θ^* that it will use to run the algorithm. The next subsection focuses on solving the problem (12) using the approximation (8a)-(8b) that make it possible to use gradient-search methods and a training sequence of observations to find θ^*.

3.3 Supervised Learning via Stochastic Approximation

The stochastic approximation procedure that we use to solve (12) is given by:

$$\theta(j+1) = \theta(j) - \varepsilon Y(\theta(j)), \tag{14}$$

where

$$Y(\theta(j)) = \nabla_\theta\left(\sum_{k=1}^{N+1} L(k-t^{(j)})\,p_k(Z^{(j)},\theta)\right) \tag{15}$$

and $\{(t^j,X_1^{(j)},X_2^{(j)},\ldots,X_N^{(j)}), j=1,2,\ldots\}$ are iid sequences with the same distribution as the model in Section 3.1. We use the notation $Z^{(j)}$ to denote the statistic process (6) with parameter $\theta(j)$. From Lemma 1, for any $\theta\in\mathbb{R}^d$ the stochastic gradient is unbiased and $\mathbb{E}[Y(\theta)] = \nabla_\theta J(\theta)$. The procedure corresponds to a stochastic approximation with exogenous noise, as the change point and the observations are independent of θ. Let $\vartheta^\varepsilon(t) = \theta(\lfloor t/\varepsilon\rfloor)$ be the piecewise constant interpolation of the consecutive iterations. For each ε this process is a time-continuous stochastic process. The theory of stochastic approximation (Kushner and Yin 2003, Vázquez-Abad 1999) implies that as $\varepsilon\to 0$, the process $\vartheta^\varepsilon(\cdot)$ converges in distribution weakly to the solution of the ODE $x'(s) = -\nabla_\theta J(x(s))$. Proposition 3.2 implies that any ODE limit is a stable point of the quasiconvex function $J(\cdot)$. If J has a unique minimum it is also the unique limit point of the ODE. Polyak's averaging method (Kushner and Yin 2003) can be shown to decrease the variance in the estimation of the optimal value. This method uses $\hat{\theta}(j) = \frac{1}{j}\sum_{\ell=1}^{j}\theta(\ell)$.

Direct evaluation of the gradient of Y in (14) requires k computations for the k-th term, so it is of the order $O(N^2)$. The following result shows that the gradient can be calculated more efficiently.

Proposition 3.3 Given θ, a change point t and a generalized CUSUM process $\{Z_i(\theta)\}$, define

$$P_n = p_n(Z,\theta), \quad d_n = L(n-t)P_n, \quad b_n = -\sigma_h(Z_n), \quad \text{and} \quad a_i = d_i + b_i \sum_{n=i}^{N} d_n ; n = 1,\ldots,N+1,$$

with $Z_{N+1} \equiv 0$. Then $Y_\theta = \sum_{i=1}^{N} a_i \nabla_\theta Z_i$. Moreover, P_n, d_n and b_n can calculated sequentially in $O(N)$ operations, after which the a_n coefficients can be calculated backwards from $n = N$ to $n = 1$, also requiring $O(N)$ operations. Thus the gradient calculation is of order $O(N)$.

Proof. Fix the index j of the subsequence and omit the use of (j) for ease of notation. From (15) it follows that

$$Y(\theta) = \nabla_\theta \left(\sum_{k=1}^{N+1} L(k-t)\, p_k(Z,\theta) \right) = \sum_{n=1}^{N+1} L(k-t)\, p_k(Z,\theta)\, \nabla_\theta \log p_k(Z,\theta).$$

For $1 \le k \le N$, using the definition of the sigmoid function, we obtain

$$\begin{aligned}
\nabla_\theta \log p_k(Z,\theta) &= \nabla_\theta \log \prod_{i=1}^{n-1} [1-\sigma_h(Z_i(\theta))]\sigma_h(Z_n(\theta)) = \nabla_\theta \left[\sum_{i=1}^{n-1} \log(1-\sigma_h(Z_i(\theta))) + \log \sigma_h(Z_n(\theta)) \right] \\
&= \sum_{i=1}^{n-1} -\sigma_h(Z_i(\theta)) \frac{d}{d\theta} Z_i(\theta) + \Big(1-\sigma_h(Z_n(\theta))\Big) \frac{d}{d\theta} Z_n(\theta) = \sum_{i=1}^{n} b_i \frac{d}{d\theta} Z_i(\theta) + \frac{d}{d\theta} Z_n(\theta).
\end{aligned}$$

and similarly $\nabla_\theta \log p_{N+1}(Z,\theta) = \sum_{i=1}^{N+1} b_i \nabla_\theta z_i(\theta) + \nabla_\theta z_{N+1}(\theta)$. We use now these expressions for $Y(\theta)$, factoring the terms in $\nabla_\theta Z_k(\theta)$:

$$\begin{aligned}
Y(\theta) &= \sum_{n=1}^{N+1} L(n-t)\, p_n(Z,\theta) \left[\sum_{i=1}^{n} b_i \frac{d}{d\theta} Z_i(\theta) + \frac{d}{d\theta} Z_n(\theta) \right] \\
&= \left(\sum_{i=1}^{N+1} b_i \left(\sum_{n=i}^{N+1} d_n \right) \frac{d}{d\theta} Z_i(\theta) \right) + \sum_{n=1}^{N+1} d_n \frac{d}{d\theta} Z_n(\theta) = \sum_{i=1}^{N+1} a_i \frac{d}{d\theta} Z_i(\theta) = \sum_{i=1}^{N} a_i \frac{d}{d\theta} Z_i(\theta)
\end{aligned}$$

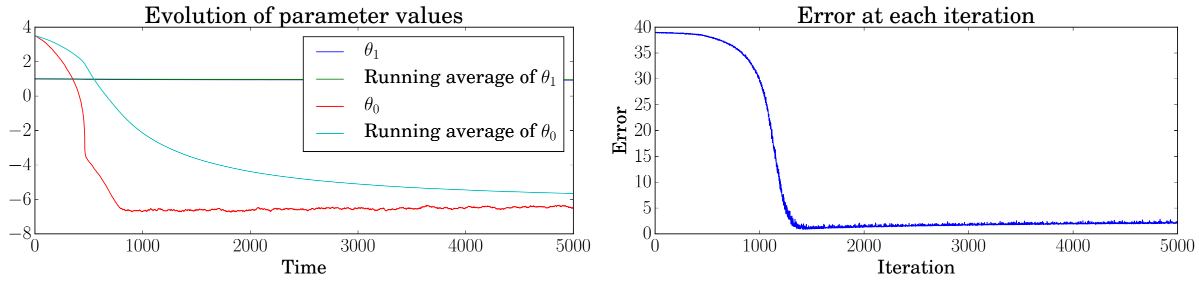
where the last equality follows since $Z_{N+1} = 0$. The variables P_n, b_n and d_n can be computed at the same time as the Z_n. Using these variables, the a_n can be calculated backwards: First a_N, then a_{N-1}, etc, keeping a register with the partial sums $\sum_{n \ge i} d_i$. Therefore the complexity for one gradient calculation is $O(N)$. □

4 NUMERICAL RESULTS FOR SYNTHETIC DATA

Type of sequence	Q_0	Q_1	t	N
Normal	$\mu_0 = 3, \sigma^2 = 1$	$\mu_1 = 10, \sigma^2 = 1$	40	60
Exponential	$\lambda_0 = 1/3$	$\lambda_1 = 1/10$	40	60
Bernoulli	$p_0 = 0.2$	$p_1 = 0.8$	40	60

Table 1: Parameters of each experiment.

We performed simulation experiments for training a change detector for various models of iid sequences with change in mean. We consider Normal, Exponential, and Bernoulli distributions, as shown in Table 1. We used $g(x,\theta) = \theta_1 x + \theta_0$. Optimization was initialized at $\theta_1 = 1, \theta_2 = 0.5$, and used $N = 20{,}000$ random sequences, with constant step-size $\varepsilon = 0.0001$, and the threshold was $h = 10$.



(a) (b)

Figure 2: Normal experiment results (a) Stochastic approximation (14), (b) Approximation of the objective (11) at each stage.

For the Normal distribution experiment, the sequence of parameters with their moving averages are shown in Fig. 2a. To get a sense of how well optimization is progressing, in Fig. 2b we show the approximate value of the objective 11. This is the error when using the detector which uses the hard-threshold stopping time T. The progress during optimization for the Bernoulli and Exponential sequences is shown in Fig. 3. After training, we validated the detector using new sequences with a different change point t than that used for training. The detector is of the form (10), using the value $\hat{\theta}(j)$ at the final value $j = 20,000$ of the stochastic approximation that used $t = 40$. Figs 4 show histograms resulting from replicating each scenario 1000 times for the Normal experiment. The Bernoulli and Exponential experiments yielded similar results.

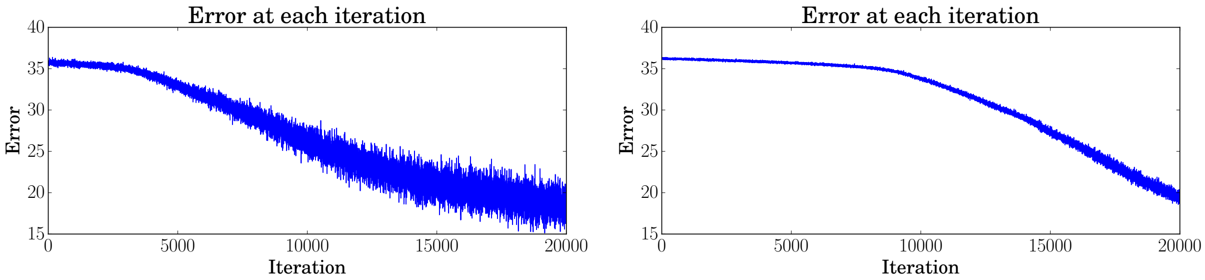
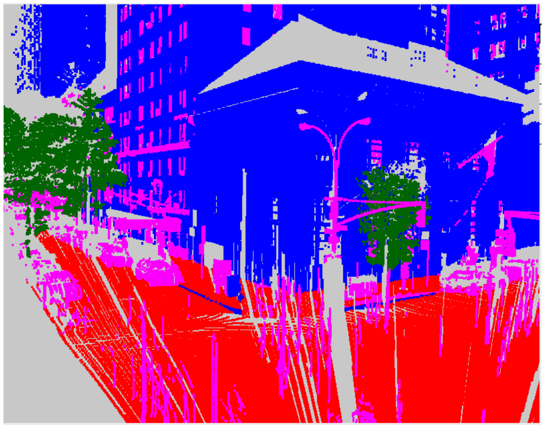
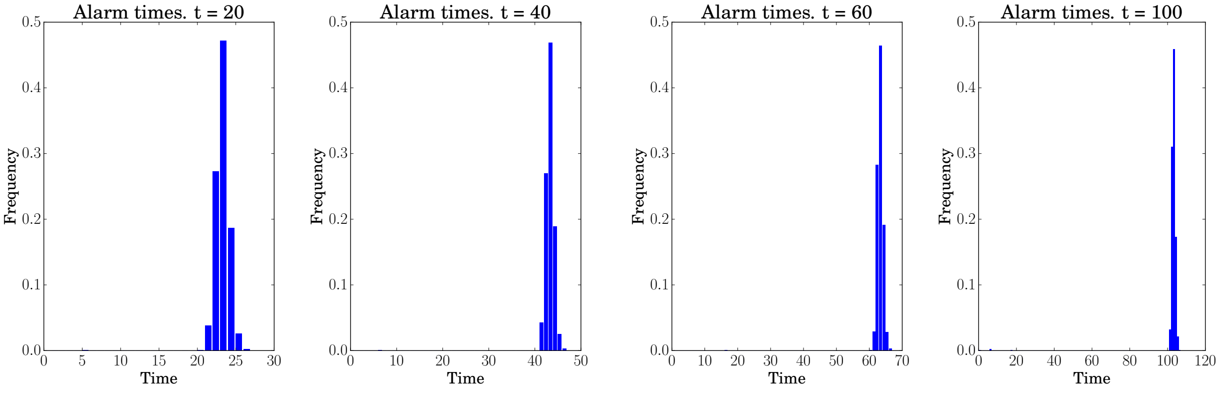


Figure 3: Errors at the running averages for the Bernoulli (left) and the Exponential (right) experiments.

5 3D POINT CLOUD CLASSIFICATION



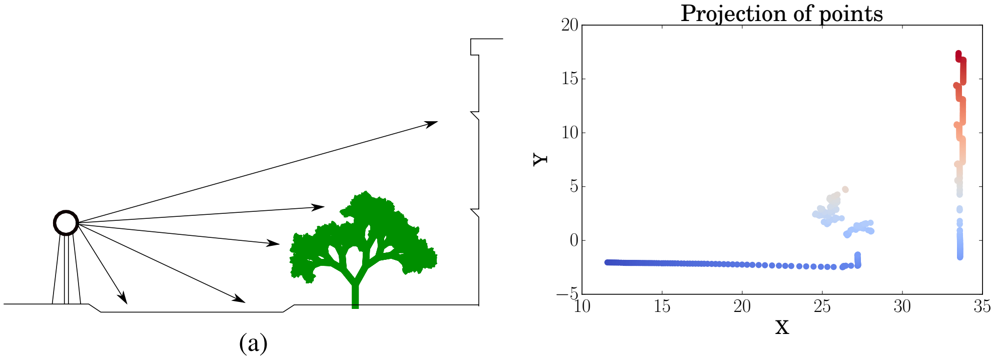
To the left we see a 2D perspective projection of a typical scene from our data set. In this scene, one can see ground, tree, and facade points. The aim of the optimization is to train a change detector that can find the transition from non-vegetation to vegetation. One of our primary motivations is classification in 3D computer vision, where the goal is to classify a stream of 3D points that are imaged by a laser range scanner. A Schematic operation of the 3D range scanner is shown in Fig. 5a. The scanner (shown on the left) sends laser beams into the scene in a sequential fashion, using the time of flight to determine the distance of the points. In this example, the scanner first encounter road points, then tree points, and finally building points.



(a) (b) (c) (d)

Figure 4: Histograms of $T(\hat{\theta})$: (a) $t = 20$, Fig. (b) $t = 40$, (c) $t = 60$, and (d) $t = 100$.

In Fig. 5b we see a projection on to the horizontal plane of the points making up one scan line in our data set. In this scan line one can make out ground points, followed by vegetation and then building facade points. Change detection algorithms should detect transitions such as ground to building, ground to vegetation, etc.



(a)

(b)

Figure 5: A Schematic operation of the 3D range scanner showing one scanline.

Flynn, Hadjiliadis, and Stamos (2015) develop a system to classify 3D points using change detectors. Novel summary statistics are computed online as the input to CUSUM-based change detection algorithms. In this section we suggest how this can be combined with the supervised learning approach of Section 3. We focus on a small instance of this difficult problem: detecting a change from non-vegetation to vegetation.

5.1 Related Work

A variety of approaches for classifying the points obtained by 3D range scanners have been explored. Munoz et al. (2009) and Zhao et al. (2010) use methods based on Markov Random Fields (MRF) and they require all the data before an inference can be made. This is in contrast to the online approach of the present approach. The 'training' phase of the MRF methodology typically consists in optimizing the weights applied to each feature vector, which is similar to our approach. Other methods include those based on convolutional neural networks, such as described by Prokhorov (2010) and by Zelener and Stamos (2016). Flynn, Hadjiliadis, and Stamos (2015) apply change detection methods to this problem. however our work there is based on hand constructed statistical models. The present work combines machine learning and change detection-based methods, by using optimization to tune the parameters of the detector.

5.2 Summary Statistics

The classification algorithm processes each scanline independently. Let $P_1, P_2, \ldots$, where $P_i \in \mathbb{R}^3$ denote the sequence of 3D coordinates of the points in the scanline. Fig. 5b shows the 2D projection of a sequence of points making up one scanline. The angle between the vector $P_i - P_{i-1}$ and the z-axis (the up direction) is denoted by A_i. Setting $e_3 = (0,0,1)$ to be the unit vector on the z-axis, the algorithm computes

$$A_i = \frac{1}{\|P_i - P_{i-1}\|}(P_i - P_{i-1})^\top e_3.$$

On all ground points, $A_i \approx 90$ degrees, while on vertical surfaces (such as buildings) $A_i \approx 0$. In vegetation, we expect the angles to be seem random. We refer the reader to (Flynn, Hadjiliadis, and Stamos 2015) for more details. The calibration of the laser scanner allows us to extract the height, relative to the scanner, at each point. This is the z-component of our 3D points, $H_i = P_i^\top e_3$. We also use a low level feature called voxel-occupancy vectors V_i. These features encode the local neighborhood around a point as follows. To calculate a voxel occupancy vector at a point, the program takes a cube in 3D space centered at the point, then discretizes this cube, creating a 3D grid of cubes, and we count how many points land in each cell. This grid is then vectorized to obtain the feature. Our cube had a side length of 0.2 meters, and this was discretized into a 3D grid of side length 3. Therefore, the voxel feature V_i is 27 dimensional.

In summary, the input statistic to our algorithm for each point in the scanline is a 29 dimensional vector $X_i = (A_i, H_i, V_i)$ consisting of the angle, height, and voxel-occupancy information.

5.3 Ground Truth Data

The dataset contains urban scenes, with buildings, cars, sidewalks, and vegetation. For training the change detector we used a collection of manually labeled 3D scanlines, indicating when transition occurred from ground to vegetation (if such a transition did occur). Specifically, our data set was an aggregate of 4 scans, for a total of 3328 scan lines. Some scanlines had a transition into vegetation points and some did not.

5.4 Experimental Results

The algorithm of Section 3.3 was used to solve the optimization problem (12). In the context of supervised learning a database with examples of the observation sequences $(X^{(s)}; s = 1, \ldots, S)$ is available for training the machine. A subset of these samples are used for the training in the stochastic approximation as follows: at iteration j a sequence is chosen from the available ones at random (with replacement) to calculate $Y(\theta^{(j)})$. We partitioned our data into training and testing sets as follows. A group of 250 scan lines from each of the 4 scans was chosen for testing. This results in a total of 1000 scan lines for testing. The remaining $S = 2328$ were used for optimization. From here, we proceed as in the synthetic experiments. Our algorithm started from a randomized choice of θ and used a step-size of $\varepsilon = 10^{-7}$. At each step of optimization our algorithm calculated gradients using a random scan line. Optimization ran for 1 million iterations, and took just over 20 minutes on a 2.2 Ghz. Linux machine with 8 Gb. of memory.

The running error during optimization is shown in Fig. 6a. Each point in this plot is the approximate value of the objective function, using parameters from the corresponding iteration of optimization. In the beginning of optimization, the change detector seems to simply count the points - the alarm goes off within the first few dozen observations, independently of whether vegetation is present. As optimization progresses, the parameters adapt to yield a more discriminating detector. After one million steps of optimization, the change detector exhibits an average error of around 63, meaning the alarm goes off on average within 63 time steps of when the change actually occurred.

This preliminary experiment on 3D data suggests that the methodology is practical. A full implementation could have a number of extensions. For instance, a backtracking step could be useful. This means to refine the estimate of the time $T(\theta)$, which likely has some of delay, by running another change detector in the opposite direction after a detection. Additionally, there is extra structure in the stream of points that

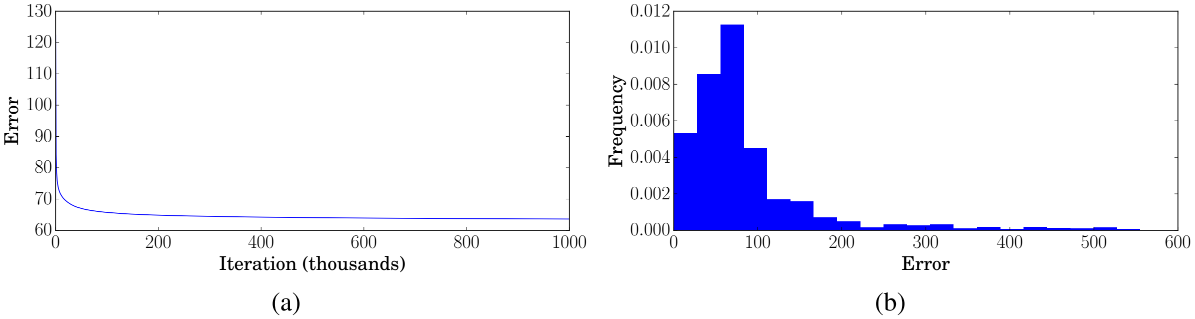


Figure 6: In Fig. 6a, the sequence of errors when using the stopping rule T evaluated during optimization for the 3D experiment. The error function is defined as in Equation (11). A histogram of the errors observed using the final parameters is shown in Fig. 6b

can be taken advantage of. This could involve running the detector not only within a scanline, but also simultaneously across a scanline. Combining the results of the two detectors could improve the results.

6 CONCLUSION

We proposed a methodology for applying supervised learning to change detection. Motivated by the CUSUM algorithm, we consider a change detector depending on a parameter and try to optimize the parameter. The detector is designed so the overall error is differentiable and gradient based methods can be applied. Experiments on synthetic and real-world data suggest that it may be useful in practice. There are several extensions to our work that are possible. We considered functions $g(x,\theta)$ that are linear in θ, and it may be useful to extend this to more complex functions, for example neural networks. We focused on monitoring for a change that happens once. In applications the behavior may be more complicated, such as switching back and forth, and dealing with more than two regimes.

ACKNOWLEDGMENTS

This work was partially supported by grants CUNY IRG-21-2153 grant and by the CUNY Institute for Computer Simulation, Stochastic Modeling and Optimization.

REFERENCES

Andrieu, L., G. Cohen, and F. J. Vázquez-Abad. 2011. “Gradient-Based Simulation Optimization Under Probability Constraints”. *European Journal of Operational Research* 212 (2): 345–351.

Flynn, T., O. Hadjiliadis, and I. Stamos. 2015, Oct. “Online Classification in 3D Urban Datasets Based on Hierarchical Detection”. In *3D Vision (3DV), 2015 International Conference on*, 380–388.

Hadjiliadis, O., and I. Stamos. 2010, May. “Sequential Classification in Point Clouds of Urban Scenes”. In *5th International Symposium on 3D Data Processing, Visualization and Transmission*. Paris, France.

Hinton, G. E., and J. Sejnowski. 1983. “Optimal Perceptual Inference”. In *Computer Vision and Pattern Recognition*.

Karatzas, I. 2003. “A Note on Bayesian Detection of Change-points With an Expected Miss Criterion”. *Statistics & Decisions/International mathematical Journal for stochastic methods and models* 21 (1/2003): 3–14.

Kushner, H., and G. G. Yin. 2003. *Stochastic Approximation and Recursive Algorithms and Applications*, Volume 35. Springer Science & Business Media.

Munoz, D., J. A. Bagnell, N. Vandapel, and M. Hebert. 2009. “Contextual Classification With Functional Max-Margin Markov Networks”. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 975–982. IEEE.

Page, E. S. 1954. “Continuous Inspection Schemes”. *Biometrika* 41 (1/2): 100–115.

Poor, V., and O. Hadjiliadis. 2009. *Quickest detection*. 1st ed. New York, New York: Cambridge University Press.

Prokhorov, D. 2010, May. “A Convolutional Learning System for Object Classification in 3-D Lidar Data”. *IEEE Transactions on Neural Networks* 21 (5): 858–863.

Stamos, I., O. Hadjiliadis, H. Zhang, and T. Flynn. 2012, Oct. “Online Algorithms for Classification of Urban Objects in 3D Point Clouds”. In *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, 332–339.

Vázquez-Abad, F. J. 1999. “Strong Points of Weak Convergence: A Study Using RPA Gradient Estimation for Automatic Learning”. *Automatica* 35 (7): 1255–1274.

Zelener, A., and I. Stamos. 2016. “CNN-based Object Segmentation in Urban LIDAR With Missing Points”. In *3D Vision (3DV), 2016 Fourth International Conference on*, 417–425. IEEE.

Zhao, H., Y. Liu, X. Zhu, Y. Zhao, and H. Zha. 2010. “Scene Understanding in a Large Dynamic Environment through a Laser-Based Sensing”. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, 127–133. IEEE.

AUTHOR BIOGRAPHIES

THOMAS FLYNN is a doctoral student in the Department of Computer Science at the Graduate Center, CUNY. His research interests are optimization, machine learning, and computer vision. His email address is tflynn@gradcenter.cuny.edu.

OLYMPIA HADJILIADIS is a Professor of Mathematics and Statistics at Hunter College and a member of the graduate faculty in the Departments of Mathematics and Computer Science of the Graduate Center. She received her PhD from Columbia University’s Statistics Department. She then went on to spend two years at Princeton University’s department of electrical engineering where she co-authored a book on Quickest detection. Professor Hadjiliadis research lies in the area of Applied probability and, more specifically, on quickest detection and sequential identification. Her email address is olympia.hadjiliadis@gmail.com.

IOANNIS STAMOS is a Professor of Computer Science at Hunter College and at the Graduate Center of the City University of New York since 2001. He received his Ph.D. degree from the Computer Science Department of Columbia University. Professor Stamos founded the Computer Vision Laboratory at Hunter College consisting of state-of-the-art laser sensors and cameras, as well as mobile robots. His current research interests are in the broad area of 3D processing, classification and identification from range and image data. His email address is istamos@hunter.cuny.edu.

FELISA VÁZQUEZ-ABAD is Professor of Computer Science at Hunter College of the City University New York. She is Executive Director of the CUNY Institute for Computer Simulation, Stochastic Modeling and Optimization that she helped to create in 2013. She has a Ph.D. in Applied Mathematics from Brown University. She was a professor at the University of Montreal, Canada in 1993 until 2004 when she became a professor at the University of Melbourne, Australia, until 2009. Her interests focus on the optimization of complex systems under uncertainty, primarily to build efficient self-regulated learning systems. Her email address is felisav@hunter.cuny.edu.