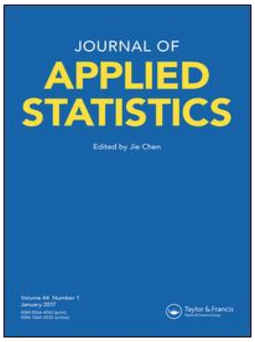


JOURNAL OF
APPLIED
STATISTICS
Edited by Jie Chen

Journal of Applied Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/cjas20

Two multivariate online change detection models

Lingzhe Guo & Reza Modarres

JOURNAL OF APPLIED STATISTICS
https://doi.org/10.1080/02664763.2020.1815674

Taylor & Francis
Taylor & Francis Group

Two multivariate online change detection models

Lingzhe Guo and Reza Modarres

Department of Statistics, The George Washington University, Washington, DC, USA

ABSTRACT

Online change point detection methods monitor changes in the distribution of a data stream. This article discusses two non-parametric online change detection methods based on the energy statistics and Mahalanobis depth. To apply the energy statistic, we use sliding-window algorithm with efficient training and updating procedures. For Mahalanobis depth, we propose an algorithm to train the threshold with desired protective ability against false alarms and discuss factors that have an influence on the threshold. Numerical studies evaluate and compare the performance of the proposed models with three existing methods to detect changes in the mean and variability of a data stream. The methods are applied to detecting changes in the flowing volume of the Mississippi River.

ARTICLE HISTORY
Received 3 January 2020
Accepted 22 August 2020

KEYWORDS
Online change detection; nonparametric; sliding-window algorithm; energy statistics; depth model

MSC 2010
62G10; 62G20; 62H15

1. Introduction

Recent technological advances have paved the way to collect immense amounts of data that stream at rapid speed. Examples include medical, environmental, video surveillance, sensor networks, and traffic monitoring data where an infinite sequence of observations is streamed. Sensors provide an incredible amount of information that may change over time. In environment monitoring, a set of sensors keep track of environmental variables such as presence and amount of particular elements in air, water, or soil [4,16]. The sensors take measurements at regular time intervals and transmit them for further processing and analysis. The same schema holds for medical monitoring where sensors that are attached to a patient continuously measure vital signs such as blood pressure, heart rate, temperature, and oxygen intake [12,22]. Other applications include fraud detection in banking systems [24] where millions of transactions take place in a short period of time, as well as anomaly detection in surveillance, early-warning systems for tsunamis and earthquakes [18]. In these applications, it is crucial to find and report the changes in the data stream as fast as possible.

The aim of change point detection methods is to find changes in a series of time-ordered multivariate observations when a property of the time series changes. Detection procedures are traditionally classified as *offline* and *online*. For offline detection procedures, the entire

CONTACT Reza Modarres reza@gwu.edu Department of Statistics, The George Washington University, Washington, DC 20052, USA

data set is available at once. In contrast, the online change detection process runs continuously as more observations become available. Suppose $\{\mathbf{Z}_i\}_{i=1}^{T}$ is a sequence of independent random vectors in $\Re^d$ with probability distribution functions F_i. An offline change point detection considers the null and alternative hypotheses,

$$H_0 : F_1 = F_2 = \cdots = F_T$$
$$H_a : F_1 = \cdots = F_{\tau_1 - 1} \neq F_{\tau_1} = \cdots = F_{\tau_2 - 1} \neq F_{\tau_2} = \cdots = F_{\tau_s - 1} \neq F_{\tau_s} \cdots = F_T,$$

where $1 < \tau_1 < \tau_2 < \cdots < \tau_s < T$ are the respective unknown locations of the change points and s is the unknown number of change points.

In online detection, the entire dataset is not available at once since the data are collected continuously and has the structure of a data stream. A data stream is a potentially infinite sequence of observations obtained from transient data that arrive continuously. We use $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ to denote a data stream, where $\mathbf{Z}_i$ is a data vector arriving at time stamp i. For a sequence with the structure of data stream, the offline approaches and criteria are not suitable. The goal of an offline change detection problem is to locate all the change points in a given sequence whereas in an online setting, the goal is to detect the next change point as soon as possible. Given a data stream $\{\mathbf{Z}_i\}_{i=1}^{\infty}$, an online change detection method considers the null and alternative hypotheses

$$H_0 : \mathbf{Z}_i \sim F_1 \text{for } i = 1, 2, \ldots,$$
$$H_a : \mathbf{Z}_i \sim F_1 \text{for } i < \tau, \text{and } \mathbf{Z}_i \sim F_2 \text{for } i \geq \tau,$$

where F_1, F_2 are different distribution functions, and τ is the location of the change point.

An online change detection model can be parametric or non-parametric. Parametric models need to assume a particular distribution in the detection process. Wald [23] proposed the sequential probability ratio test by monitoring a cumulative variable defined as the difference between the current value and the cumulative sum of the log likelihood ratio. Page [19] used the sequential test on time series and proposed cumulative sum control chart (CUSUM) to detect change points. When the cumulative variable exceeds a predefined threshold, a change in the process is reported. Wu [25] proposed a streaming algorithm that uses *sketches* as basic components to summarize the input and perform fast and space-efficient sequential tests. Yildirim et al. [26] presented a sequential Monte Carlo expectation-maximization algorithm to detect the changes in a Markov chain. Crosier [2], Mei [15] and Zou et al. [30] extended CUSUM to high-dimensional data streams. Zou et al. [28] combined Bayesian information criterion (BIC) with LASSO variable selection method to build a unified diagnosis framework. Kawaharal, and Sugiyama [6] present a fast change–point detection algorithm based on direct density-ratio estimation. Miyaguchi and Yamanishi [17] discussed the online change detection procedure when the changes are continuous instead of abrupt.

When the distributional assumptions are violated, parametric models may not perform as expected. In contrast, non-parametric models do not need to specify the distribution in advance and are more robust when dealing with different underlying distributions. Bakir and Reynolds [31] considered the cumulative sum of the Wilcoxon signed-rank statistic for group observations. Chakraborti and Van de Wiel [32] proposed a Shewhart chart using the Mann–Whitney test statistic. An exponentially weighted moving-average (EWMA) control chart with a non-parametric goodness-of-fit test is investigated by Zou and Tsung [29].

Such online change detection method aims to detect different types of changes in location, scale, or shape. However, the EWMA control chart requires pre-specifying the weight parameter. Regarding this issue, Li [33] developed an adaptive CUSUM Chart without depending on any tuning parameter. In multivariate settings, Messaoud et al. [13] proposed an EWMA control chart based on data depth. Li et al. [8] proposed a self-starting EWMA chart based on forward variable selection. Li et al. [9] discussed the impact of high dimension on the online detection procedure and provided a diagnostic procedure to control the weighted missed discovery rate. Zhang et al. [27] proposed a rank-based approach through random projections for high-dimensional monitoring. Chen et al. [1] developed a distribution-free multivariate control chart using Wilcoxon rank-sum test for each component and applied it on a semiconductor manufacturing process. In this paper, we will include the method from Chen et al. [1] in the simulation Section for comparison.

This paper contributes to the literature by proposing two approaches for online change detection without specifying the underlying distribution or the type of changes. We build multivariate non-parametric online change detection procedure with the energy statistics, which is widely used in offline change point detection. We also discuss the detection method using depth function and make a comparison between them. The contribution of this paper includes reducing the computing complexity of using energy statistics in online settings, proposing the algorithm of determining the thresholds for desire ability to prevent false alarms, and comparing different multivariate online detection methods under different distributions.

This paper is organized as follows. In the next Section, we discuss the criteria for evaluating an online change detection method. Section 3 presents an online detection method with the sliding-window model based on the energy statistic. We discuss the online change detection method based on data depth in Section 4.

Section 5 discusses the influence of the hyper-parameters on the proposed methods. The proposed methods are compared with three existing methods to detect changes in the means and variances in Section 6. We apply the methods to detect changes in the flowing volume of the Mississippi River in Section 7. The last section is devoted to summary and concluding remarks.

2. Online evaluation criterion

In the online change detection problem, we need to make a decision on every newly received observation as observations become available in time order. This represents a repeated sequence of hypothesis tests and raises the issue of multiple testing. In offline settings, we use the α level (Type-I error) to measure the misdetection rate of the testing method. The criterion is usually the $(1-\alpha)$-th quantile of the distribution of the test statistic under the null hypothesis of no change. This criterion ensures that the probability of a false change point is bounded by α. Such criterion is useful in offline scenarios because we only need to make one decision for the entire dataset and the null distribution of the statistic can be derived or approximated. One of the challenges of any online change detection method is to find the null distribution of the statistic when the sample size is not fixed. Because the total number of tests is unknown and the test statistics are usually correlated, a Bonferroni correction is difficult to apply. Thus, the α level (Type-I error) is not a feasible criterion in online settings.

In the online settings, the system is in-control (IC) when the distribution of the observations does not change. Correspondingly, any change of the observations' distribution leads to out-of-control (OC) status. The purpose of an online change detection procedure is to make reaction when OC performance occurs. Qiu [20] discussed new criteria for the evaluation of online change detection procedures.

Definition 2.1 (In-control run length (IC-RL)): Suppose L_t is the tth test statistic of an in-control data stream $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ and a change is declared when L_t exceeds the threshold h. The in-control run Length(IC-RL) is defined as

$$\mathrm{IC} - \mathrm{RL} = \min\{t : L_t > h\} - 1 \tag{1}$$

Definition 2.2 (Out-of-control run length (OC-RL)): Consider a data stream $\mathbf{Z}_1, \ldots, \mathbf{Z}_\tau$, $\mathbf{Z}_{\tau+1}, \ldots$, where a change occurs at $\mathbf{Z}_\tau$. Suppose L_t is the test statistic of an online detection method based on $\{\mathbf{Z}_i\}_{i=1}^{t}$ and h is the threshold of declaring a change. The out-of-control run length (OC-RL) of an online change detection method is defined by

$$\mathrm{OC} - \mathrm{RL} = \min\{t : L_t > h,\ t > \tau\} - \tau \tag{2}$$

By the definitions, we can see that the IC-RL of an online detection procedure counts the number of observations before a change point is declared under an in-control data stream. Larger IC-RL implies less chance of false alarms. The OC-RL shows how fast an online method detects a change. Smaller OC-RL implies a more sensitive detection process. Note that both IC-RL and OC-RL are random variables whose distributions are determined by the distribution of the data stream, the change detection method, and the value of the threshold h. To compare the performance of different online change detection methods, Qiu [20] considers the mean of IC-RL and OC-RL. We denote the means as IC-ARL and OC-ARL, respectively. Kifer et al. [7] proposed another criterion as the protective ability against false alarms. It is definition is shown as follows.

Definition 2.3 (Protective ability against false alarms (RL_α)): The ability of an online change detection method to prevent false alarms is measure by RL_α, which is the αth quantile of IC-RL defined by Equation (1). Large value of RL_α implies good protection against false alarms.

For an integer n, the inequality $RL_\alpha \geq n$ implies that the probability of reporting any false alarms given n observations is at most α. In real applications, RL_α might be more useful than IC-ARL because it bounds the probability of showing false alarms. If a change is declared within RL_α, the possibility of it being a false alarm is very small. In this paper, we use RL_α and OC-ARL to compare different online change detection methods.

3. Sliding-window algorithm with energy statistics

A time window is a consecutive segment of the data stream. The sliding-window model for online change detection performs homogeneity tests over two different time windows that are defined below.

Definition 3.1 (Baseline window): The time window that contains the observations from the unchanged distribution is the baseline window.

Definition 3.2 (Current window): The time window that contains the latest observations in the data stream is the current window.

The analysis of the window-based model mainly concerns the choice of test statistics, initialization, and update of the *baseline* and *current* windows.

Algorithm 1 describes the sliding-window method. Suppose $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ is a data stream of independent random vectors in $\Re^d$ and $F_1, F_2, F_3, \ldots$ are the corresponding distributions. Our aim is to find the time point τ such that $F_\tau \neq F_{\tau+1}$ and we call time τ a change point. Let $\mathscr{B}$ be the *baseline window*, which contains the first n_1 observations of the data steam, i.e $\mathscr{B} = \{\mathbf{Z}_i\}_{i=1}^{n_1}$. We assume that $\tau > n_1$, which ensure that the change point is not in the *baseline* window. The *current window* $\mathscr{C}$ contains the latest n_2 observations from the data stream. To start with, the *current window* contains the n_2 observations immediately after the *baseline window*. When a new observation arrives, we slide the *current window* one data point to the right. This absorbs the new observation and discards the oldest observation. Therefore, the *current window* is $\mathscr{C}_t = \{\mathbf{Z}_i\}_{i=t}^{t+n_2-1}$, for $t = n_1 + 1, n_1 + 2, \ldots$. The parameter n_1 and n_2 are known as the width of the *baseline* and the *current windows*. At each update of the *current window*, we test for the homogeneity of the *baseline* and *current* windows and compare the statistic $L(\mathscr{B}, \mathscr{C}_t)$ with a predefined threshold h. We declare a change when $L(\mathscr{B}, \mathscr{C}_t) > h$. After a change is found, we refresh the *baseline* and the *current windows* by the $n_1 + n_2$ data points after the change point and repeat the entire process.

Algorithm 1: Sliding-Window Algorithm

```
Input  : Data stream $\{\mathbf{Z}_i\}_{i=1}^{\infty}$, using the baseline window width $n_1$, the current
         window width $n_2$.
Output: The locations of the change points in the original data stream $\tau_1, \tau_2, \tau_3, \ldots$
$s = 1, \tau_0 = 0$;
while not at end of the data stream do
    t=0;
    $\mathscr{B} = \{\mathbf{Z}_1, \ldots, \mathbf{Z}_{n_1}\}$;
    do
        $t = t + 1$;
        $\mathscr{C}_t = \{\mathbf{Z}_{n_1+t}, \ldots, \mathbf{Z}_{n_1+t+n_2-1}\}$;
        Calculate the statistic $L(\mathscr{B}, \mathscr{C}_t)$ based on $\mathscr{B}$ and $\mathscr{C}_t$;
    until $L(\mathscr{B}, \mathscr{C}_t) > h$;
    A change point is detected and recorded as $\tau_s = \tau_{s-1} + n_1 + t$.;
    Discard the observations before $\mathbf{Z}_{n_1+t+n_2}$ and re-index the observations
      $\mathbf{Z}_{n_1+t+n_2}, \mathbf{Z}_{n_1+t+n_2+1}, \ldots$ by $1, 2, 3, \ldots$;
end
return $\tau_1, \tau_2, \tau_3, \ldots$
```

Kifer et al. [7] consider the univariate online detection and use the Wilcoxon and the Kolmogorov–Smirnov test statistics. However, these statistics only work for the univariate two-sample tests. To deal with multivariate observations, interpoint distance methods are valuable. Methods based on interpoint distances can process high-dimensional data, with a known update mechanism (See Section 3.1) as new data points become available. Székely and Rizzo [21] propose the energy statistic based on interpoint distances as a non-parametric two-sample test. Matteson and James [14] apply such statistics in the offline change detection scenario. We use the energy test statistic in the sliding-window model for online change detection of multivariate data and discuss it in Section 3.1.

The value of the threshold h is important in a change detection procedure. If the threshold h is too small, it is more likely to reject the null hypothesis, which leads to more false alarms. However, if h is too large, the change detection method will have little power. We determine the threshold h to ensure that the RL_α is larger than some user-specified constant. Thus, we control the risk of reporting false alarms for the change detection models. To calculate h for the corresponding RL_α, we propose a training approach described in Algorithm 2. This training process is conducted offline and will not contribute to additional computational overhead in the online detection procedure.

Suppose R is the number of the training samples and each sample contains $n_1 + RL_\alpha$ observations. If the training samples are not given, we can generate the training samples with $n_1 + RL_\alpha$ observations from the data stream by taking R random permutations. Suppose $\{\mathbf{Z}_i^{(r)}\}_{i=1}^{n_1+RL_\alpha}$ is the rth training sample, for $r = 1, \dots, R$. We compute the values $L(\mathscr{B}^{(r)}, \mathscr{C}_t^{(r)})$, where $\mathscr{B}^{(r)} = \{\mathbf{Z}_i^{(r)}\}_{i=1}^{n_1}$ and $\mathscr{C}_t^{(r)} = \{\mathbf{Z}_i^{(r)}\}_{i=n_1+t}^{n_1+t+n_2-1}$, for $t = 1, \dots, RL_\alpha - n_2 + 1$. Let $h^{(r)}$ be the maximum value of the $L(\mathscr{B}^{(r)}, \mathscr{C}_t^{(r)})$, i.e. $h^{(r)} = \max_t L(\mathscr{B}^{(r)}, \mathscr{C}_t^{(r)})$, for $r = 1, \dots, R$. Let Q_α be the α percentile of $\{h^{(1)}, \dots, h^{(R)}\}$. The threshold h is defined by $Q_{1-\alpha}$.

```
Algorithm 2: Training for the threshold $h$.
  Input  : protective ability against false alarms $RL_\alpha$; the number of training
           samples $R$; training samples $\mathbf{Z}_1^{(r)}, \dots, \mathbf{Z}_{n_1+RL_\alpha}^{(r)}$, for $r = 1, \dots, R$; the
           baseline window width $n_1$, the current window width $n_2$.
  Output: The threshold $h$.
  for $r = 1, \dots, R$ do
      The $r$-th training sample is $\mathbf{Z}_1^{(r)}, \mathbf{Z}_2^{(r)}, \dots, \mathbf{Z}_{n_1+RL_\alpha}^{(r)}$;
      $\mathscr{B} = \{\mathbf{Z}_1^{(r)}, \dots, \mathbf{Z}_{n_1}^{(r)}\}$;
      for $t = 1, \dots, RL_\alpha - n_2 + 1$ do
          $\mathscr{C}_t^{(r)} = \{\mathbf{Z}_{n_1+t}^{(r)}, \dots, \mathbf{Z}_{n_1+t+n_2-1}^{(r)}\}$;
          Calculate $L(\mathscr{B}^{(r)}, \mathscr{C}_t^{(r)})$;
      end
      $h^{(r)} = \max_t L(\mathscr{B}^{(r)}, \mathscr{C}_t^{(r)})$;
  end
  $h = Q_{1-\alpha}$ where $Q_\alpha$ is the $\alpha$ percentile of $\{h^{(1)}, \dots, h^{(R)}\}$;
  return $h$
```

We will next discuss the energy statistics for the change detection model and derive a computing shortcut for updating the windows. A numerical study is conducted under bivariate normal distribution and the influence of the window length is discussed.

3.1. Updating the energy statistic

Székely and Rizzo [21] propose the energy statistic based on the interpoint distances within and between two groups for testing the equality of two distributions. We apply the test to detect changes in the sliding-window model as follows.

Let L_t be the tth statistic for finding a new change point. The baseline and current windows are $\mathscr{B} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_{n_1}\}$ and $\mathscr{C}_t = \{\mathbf{Z}_{n_1+t}, \dots, \mathbf{Z}_{n_1+t+n_2-1}\}$, respectively. We assume that the distribution of the observations has finite first absolute moment. To test the homogeneity of $\mathscr{B}$ and $\mathscr{C}_t$, the energy statistic is

$$L_t = L(\mathscr{B}, \mathscr{C}_t) = 2\hat{\mu}_{BC} - \hat{\mu}_{BB} - \hat{\mu}_{CC}, \tag{3}$$

where

$$\hat{\mu}_{BC} = (n_1 n_2)^{-1} \sum_{i=1}^{n_1} \sum_{j=n_1+t}^{n_1+t+n_2-1} \|\mathbf{Z}_i - \mathbf{Z}_j\|,$$

$$\hat{\mu}_{BB} = \binom{n_1}{2}^{-1} \sum_{i=1}^{n_1-1} \sum_{j=i+1}^{n_1} \|\mathbf{Z}_i - \mathbf{Z}_j\|,$$

$$\hat{\mu}_{CC} = \binom{n_2}{2}^{-1} \sum_{i=n_1+t}^{n_1+t+n_2-2} \sum_{j=i+1}^{n_1+t+n_2-1} \|\mathbf{Z}_i - \mathbf{Z}_j\|,$$

and $\|\cdot\|$ is the Euclidean norm for a vector.

Calculating the statistic L_t directly from the $\mathscr{B}$ and $\mathscr{C}_t$ is computationally inefficient. One can show that the updating procedure for the sliding-window model only modifies L_t partially to produce L_{t+1}. The *baseline window* is unchanged and the *current windows* overlap before and after an update. A computing form for L_{t+1} based on L_t is proposed in the following lemma.

Lemma 3.1: *Suppose L_t is the energy statistic based on $\mathscr{B} = \{\mathbf{Z}_1, \dots, \mathbf{Z}_{n_1}\}$, and $\mathscr{C}_t = \{\mathbf{Z}_{n_1+t}, \dots, \mathbf{Z}_{n_1+t+n_2-1}\}$. Suppose L_{t+1} is the energy statistic based on $\mathscr{B}$, and $\mathscr{C}_{t+1} = \{\mathbf{Z}_{n_1+t+1}, \dots, \mathbf{Z}_{n_1+t+n_2}\}$. One can compute L_{t+1} using*

$$L_{t+1} = L_t + (n_1 + n_2)^{-1} \sum_{i=1}^{n_1} \left\{ \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t+n_2}\| - \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t}\| \right\}$$
$$+ \binom{n_2}{2}^{-1} \sum_{i=n_1+t+1}^{n_1+t+n_2-1} \left\{ \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t+n_2}\| - \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t}\| \right\}.$$

If we calculate L_{t+1} directly by Equation (3), we need to calculate $\binom{n_1+n_2}{2}$ different norms. The computing complexity is $O((n_1 + n_2)^2)$. If we calculate L_{t+1} by Lemma 3.1, we only need to calculate $2(n_1 + n_2 - 1)$ different norms. The computing complexity reduces to $O(n_1 + n_2)$.

4. Depth model

A depth function measures the closeness of a data point with respect to the center of a probability distribution. Let F be a distribution with finite mean vector $\boldsymbol{\mu}_F$ and non-sparse covariance matrix Σ_F. Liu and Singh [11] propose the Mahalanobis depth (MD) of a data point $\mathbf{z}$ with respect to the distribution F as

$$D_{Mah}(\mathbf{z}|F) = [1 + (\mathbf{z} - \boldsymbol{\mu}_F)'\Sigma_F^{-1}(\mathbf{z} - \boldsymbol{\mu}_F)]^{-1}.$$

Suppose $\mathscr{X} = \{\mathbf{X}_1, \ldots, \mathbf{X}_n\}$ is a random sample obtained from F. The sample version of the Mahalanobis depth is

$$D_{Mah}(\mathbf{z}|F_n) = [1 + (\mathbf{z} - \bar{\mathbf{X}})'S_X^{-1}(\mathbf{z} - \bar{\mathbf{X}})]^{-1},$$

where $\bar{\mathbf{X}}$ and S_X are the mean and covariance matrix of $\mathscr{X}$ and F_n is the empirical distribution of $\mathscr{X}$. In this section, we propose an online change detection method based on data depth. The depth function we use is the Mahalanobis depth. Note that the Mahalanobis depth relies on the assumptions for the mean vector and covariance matrix. The mean should be finite and the covariance matrix should be non-sparse. If the assumptions are violated, other depth functions such as lens depth [10] that do not rely on the covariance or its inverse should be considered. When the dimension is high, the covariance matrix may be sparse. To address this situation, Holgersson and Karlsson [5] discussed three estimators for the inverse of covariance matrix.

Let $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ be a data stream where the observations are independent. Suppose F is the distribution of the observations before the change. The online change detection problem considers the null hypothesis $H_0 : \mathbf{Z}_i \sim F \text{for } i = 1, 2, 3, \ldots$, against the alternatives

$$H_a : \mathbf{Z}_i \sim F \text{ for } i = 1, \ldots, \tau - 1$$
$$\mathbf{Z}_i \nsim F \text{ for } i = \tau, \tau + 1 \cdots$$

where τ is the location of the change point. Qiu [20] discuss several online change detection methods based on Mahalanobis depth. The method we use is shown by Algorithm 1. Let $\mathscr{B}$ be the *baseline window* that contains the first n observations of the data stream. We assume that the observations in $\mathscr{B} = \{\mathbf{Z}_i\}_{i=1}^{n}$ are independent and identically distributed (i.i.d.), i.e. there is no change point in the *baseline window*. If not, we refill the *baseline window* with new observations until the i.i.d assumption is satisfied. For a new observation $\mathbf{Z}_t$ $(t > n)$, the closeness of $\mathbf{Z}_t$ to the center of $\mathscr{B}$ is measured by

$$D_{Mah}(\mathbf{Z}_t|F_n) = (1 + (\mathbf{Z}_t - \bar{\mathbf{Z}})S_Z^{-1}(\mathbf{Z}_t - \bar{\mathbf{Z}}))^{-1}, \tag{4}$$

where $\bar{\mathbf{Z}}$ and S_z are the mean and covariance matrix of $\mathscr{B}$ and F_n is the empirical distribution of $\mathscr{B}$. Suppose $\mathscr{C}_t$ is the *current window* which contains k observations $\{\mathbf{Z}_i\}_{i=t}^{t+k}$. We declare the change point $\mathbf{Z}_t$ if the depth values for the observations in $\mathscr{C}_t$ are all less than a threshold h. That is, $\max\limits_{i=t,\ldots,t+k} \{D_{Mah}(\mathbf{Z}_i|F_n)\} < h$. To reduce computing time, we update the current window $\mathscr{C}_t$ by sliding k observations to the right. Therefore, the potential current windows are adjacent instead of overlapping. For example, consider the data stream $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ and let $n = 50$ and $k = 2$. The *baseline window* is $\{\mathbf{Z}_i\}_{i=1}^{50}$ and the potential *current windows* are $\{\mathbf{Z}_{51}, \mathbf{Z}_{52}\}, \{\mathbf{Z}_{53}, \mathbf{Z}_{54}\}, \ldots$

Algorithm 3: Online change point detection with Mahalanobis depth

```
Input  : The baseline window width n, the data stream $\mathbf{Z}_1, \mathbf{Z}_2, \ldots$, the number of
         consecutive observations k, the threshold h.
Output: The next change point.
$\mathscr{B} = \{\mathbf{Z}_1, \ldots \mathbf{Z}_n\}$;
$\bar{\mathbf{Z}}$ and $S_Z$ are the mean and covariance matrix of $\mathscr{B}$;
t=0;
do
    t=t+1;
    for $i = 1, \ldots, k$ do
        $D_{Mah}(\mathbf{Z}_{n+(t-1)k+i}|F_n) = (1 + (\mathbf{Z}_{n+(t-1)k+i} - \bar{\mathbf{Z}})S_Z^{-1}(\mathbf{Z}_{n+(t-1)k+i} - \bar{\mathbf{Z}}))^{-1}$;
    end
    $MAX_t = \max(D_{Mah}(\mathbf{Z}_{n+(t-1)k+1}|F_n), \ldots, D_{Mah}(\mathbf{Z}_{n+tk}|F_n))$;
until $MAX_t < h$;
Declare a change point at $n + (t-1)k + 1$;
```

4.1. Training the threshold

In this section, we train the threshold h to ensure that the RL_α is large enough so that the probability of reporting false alarms is in control. A search algorithm is introduced, followed by the theoretical expression derived under normal distribution. The observations in the training samples should follow the same distribution as the observations in testing sample before a change. If we know the distribution, we can simulate the training samples. Otherwise, we need to apply an offline change point detection method on a few segments of the data stream and use the segments with no change point as the training samples.

Let R be the number of the training samples. Each sample contains $n + RL_\alpha$ observations from the same distribution. Algorithm 1 shows the training approach for h. Suppose $\{\mathbf{Z}_1^{(r)}, \mathbf{Z}_2^{(r)}, \ldots, \mathbf{Z}_{n+RL_\alpha}^{(r)}\}$ is the rth training sample. In the rth training process, the *baseline window* is $\mathscr{B}^{(r)} = \{\mathbf{Z}_1^{(r)}, \ldots, \mathbf{Z}_n^{(r)}\}$. The *current window* is chosen from $\mathscr{C}_s^{(r)} = \{\mathbf{Z}_i^{(r)}\}_{i=n+(s-1)k+1}^{n+sk}$, for $s = 1, \ldots, RL_\alpha/k$. We calculate the depths $D_{Mah}(\mathbf{Z}_t^{(r)}|F_n^{(r)})$, for $t = n+1, \ldots, n + RL_\alpha$. A change is declared at $\mathbf{Z}_{n+(s-1)k+1}^{(r)}$ when the depth values of all the observations in the current window $\mathscr{C}_s^{(r)}$ are less than the threshold. That is, $\max\{D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}^{(r)}|F_n^{(r)}), \ldots, D_{Mah}(\mathbf{Z}_{n+sk}^{(r)}|F_n^{(r)})\} < h$, for $s = 1, \ldots, RL_\alpha/k$. Thus, to ensure no change is detected within the running length under unchanged data stream $\{\mathbf{Z}_i\}_{i=1}^{n+RL_\alpha}$, we select the threshold $h^{(r)}$ such that $h^{(r)} \leq \max\{D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}^{(r)}|F_n^{(r)}), \ldots, D_{Mah}(\mathbf{Z}_{n+sk}^{(r)}|F_n^{(r)})\}$, for $s = 1, \ldots, RL_\alpha/k$. Therefore, the threshold $h^{(r)}$ for the rth sample is

$$h^{(r)} = \min_{s=1,\dots,RL_\alpha/k} \max(D_{Mah}(\mathbf{Z}^{(r)}_{n+(s-1)k+1}|F^{(r)}_n),\dots,D_{Mah}(\mathbf{Z}^{(r)}_{n+sk}|F^{(r)}_n)). \tag{5}$$

The trained threshold is calculated as the α percentile of $\{h^{(1)},\dots,h^{(R)}\}$.

Algorithm 4: Training the threshold for the depth model.

```
Input  : The protective ability against false alarms $RL_\alpha$; the number of training
         samples $R$; training samples $\mathbf{Z}^{(r)}_1,\dots,\mathbf{Z}^{(r)}_{n+RL_\alpha}$, for $r = 1,\dots,R$; the
         baseline window width n, the number of consecutive observation k.
Output: The threshold $h$.
for $r = 1,\dots,R$ do
    The $r$-th training sample is $\mathbf{Z}^{(r)}_1, \mathbf{Z}^{(r)}_2,\dots,\mathbf{Z}^{(r)}_{n+RL_\alpha}$;
    $\mathscr{B} = \{\mathbf{Z}^{(r)}_1,\dots,\mathbf{Z}^{(r)}_n\}$;
    for $t = 1,\dots,RL_\alpha/k$ do
        for $i = 1,\dots,k$ do
            Calculate $D_{Mah}(\mathbf{Z}^{(r)}_{n+(t-1)k+i}|F^{(r)}_n)$
        end
        $MAX_t = \max(D_{Mah}(\mathbf{Z}^{(r)}_{n+(t-1)k+i}|F^{(r)}_n),\dots,D_{Mah}(\mathbf{Z}^{(r)}_{n+tk+i}|F^{(r)}_n))$
    end
    $h^{(r)} = \min_{t=1,\dots,RL_\alpha/k} MAX_t$;
end
$h = \alpha$-th percentile of $\{h^{(1)},\dots,h^{(R)}\}$;
return $h$
```

If the data stream follows a multivariate normal distribution, we can approximate h using Chi-square distribution based on the following findings.

Theorem 4.1: *Suppose $\{\mathbf{Z}_i\}_{i=1}^n$ is a sequence of independent observations follow d-dimensional multivariate normal distribution F with mean $\boldsymbol{\mu}_F$ and covariance $\boldsymbol{\Sigma}_F$. Suppose $\mathbf{X} \sim \chi^2_d$, and $\mathbf{Y} = (1+\mathbf{X})^{-1}$. Let $\mathscr{B} = \{\mathbf{Z}_i\}_{i=1}^n$ and*

$$D_{Mah}(\mathbf{Z}|F_n) = (1 + (\mathbf{Z} - \bar{\mathbf{Z}})S_Z^{-1}(\mathbf{Z} - \bar{\mathbf{Z}}))^{-1}, \tag{6}$$

where $\bar{\mathbf{Z}}$ and S_Z are the mean and covariance matrix of $\mathscr{B} = \{\mathbf{Z}_i\}_{i=1}^n$. We show that

$$D_{Mah}(\mathbf{Z}|F_n) \xrightarrow{p} \mathbf{Y} \mathrm{as} n \to \infty. \tag{7}$$

Theorem 4.2: *Suppose $\{\mathbf{Z}_i\}_{i=1}^\infty$ is an independent data stream from a d-variate normal distribution F and we use Algorithm 3 to detect a change point on the data stream. Given a pre-set RL_α, when the size of baseline window n goes to infinity, the corresponding threshold h converge to the cth quantile of $Y = (1+X)^{-1}$, where $c = [1-(1-\alpha)^{k/RL_\alpha}]^{1/k}$ and $X \sim \chi^2_d$.*

Table 1. The thresholds trained by Algorithm 1 or computed by Theorem 4.2. ($RL_\alpha = 50000, \alpha = 0.05$).

k		1	3	5	10
By Theorem 4.2		0.035	0.106	0.170	0.303
By Algorithm 1	$n = 50$	0.026	0.088	0.145	0.272
	$n = 80$	0.029	0.094	0.151	0.285
	$n = 100$	0.030	0.096	0.159	0.286

Corollary 4.1: *Under the conditions stated in Theorem* 4.2, *when the size baseline window* n *is large, the threshold* h *can be approximated by* $(1 + \chi^2_d(1-c))^{-1}$, *where* $\chi^2_d(1-c)$ *is the* $(1-c)$*-th quantile of chi-square distribution.*

Table 1 shows a numerical study to compare the threshold given by Theorem 4.2 and the threshold trained by Algorithm 4 under bivariate normal distribution. We control the false alarms by setting $RL_\alpha = 50,000$ with $\alpha = 0.05$. The width of the baseline window and the number of consecutive observations vary in $n \in \{50, 80, 100\}$ and $k \in \{1, 3, 5, 10\}$, respectively. To train the threshold, we generate $R = 1000$ samples from standard bivariate normal distribution. Each sample contains $n + RL_\alpha$ observations. From the table, we can see that when we increase the width of the baseline window, the trained thresholds increase and get closer to the value that is computed by Theorem 4.2. However, the threshold does not change much when we change n. When the value of k increases, the value of the threshold also increases.

5. Selection of parameters

In this section, we conduct simulation studies on the performance of the proposed methods with different choices of the parameters. The ability to prevent false alarms is controlled by the RL_α. Thus, we only compare the OC-ARL when we select different value of the parameters.

5.1. Window width in the sliding-window model

The sliding-window model with energy statistic is considered under bivariate normal distributions to determine the influence of the window width on the performance of the detection procedure. The window widths are chosen from $n_1 = n_2 \in \{5, 50, 80, 100\}$. For each window width, we first use Algorithm 2 to train the threshold h with the simulated data from a standard bivariate normal distribution. The number of the training sample is $R = 1000$, and we set $RL_\alpha = 50,000$ with $\alpha = 0.05$. Table 2 shows the trained thresholds when the window width varies. From the table, we can see that the sliding-window model with larger window width needs a smaller threshold to maintain the same RL_α. We use these thresholds to detect the changes from simulated data.

With the trained h, we consider the ability of the sliding-window model to detect a single change in the data stream. Let $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{\tau-1}, \mathbf{Z}_\tau, \mathbf{Z}_{\tau+1}, \ldots$ be a data stream with independent observations and assume the distribution of the observations changes at $\mathbf{Z}_\tau$. The first $\tau - 1$ observations $\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_{\tau-1}$ follow bivariate normal distribution with mean

Table 2. The trained threshold by Algorithm 1 using bivariate standard normal distributions. ($R = 1000, \alpha = 0.05, RL_\alpha = 50,000$).

Window width (n_1)	5	50	80	100
Threshold (h)	3.450	0.363	0.218	0.171

Table 3. The OC-ARL and the probability of successful detection for sliding-window model with 1000 replications.

	n_1			
μ	5	50	80	100
0.5	12410 (0.236)	1021 (0.993)	177.2 (1)	116.9 (1)
1	6943 (0.725)	34.21 (1)	42.07 (1)	47.21 (1)
1.5	1105 (0.969)	23.81 (1)	29.69 (1)	32.47 (1)
2	42.37 (1)	18.86 (1)	23.18 (1)	25.69 (1)
2.5	10.89 (1)	15.84 (1)	19.31 (1)	21.47 (1)
3	4.801 (1)	13.85 (1)	17.01 (1)	18.66 (1)

$\boldsymbol{\mu}_0 = (0, 0)'$, and rest of the observations $\mathbf{Z}_\tau, \mathbf{Z}_{\tau+1}, \ldots$ follow bivariate normal distribution with mean $\boldsymbol{\mu}_1 = (\mu, \mu)'$. The covariances are identity matrices. Suppose $\tau = n_1 + n_2$, we generate τ observations first and we keep generating the observations until the online detecting procedure detect a change. Thus, the size of the data stream is not fixed. If the detection procedure does not detect any change in RL_α observations, we claim that the procedure fail to detect such change. Each simulation is conducted under 1000 replications and we report the OC-ARL and the probability of a successful detection for each scenario. The degree of the change is defined by $\mu \in \{0.5, 1, 1.5, 2, 2.5, 3\}$.

Table 3 shows the OC-ARL and the probability of successful detection for different window widths. From the table, we can see that there is not a value of the window width that leads to the best performance in all situations. When the change is small, the sliding-window model with large windows detects the changes more often than the model with small windows. The OC-ARL for the model with larger windows is also smaller on the average. However, when the change is severe, the model with small windows can detect the changes quicker than the model with large windows. In practice, when we need to detect sudden, large changes, a smaller window is preferred. If one wants to detect small changes that last over long periods of time, the model with large windows performs better.

5.2. The parameter n and k in the depth model

We conduct simulations to see the performance of the depth methods under different settings of k and n. Consider an independent data stream $\mathbf{Z}_1, \ldots, \mathbf{Z}_{\tau-1}, \mathbf{Z}_\tau, \mathbf{Z}_{\tau+1}, \ldots$ where the distribution of the observations changes at τ. We generate the first $\tau - 1$ observations $\{\mathbf{Z}_i\}_{i=1}^{\tau-1}$ from a bivariate normal distribution with mean $\boldsymbol{\mu}_0 = (0, 0)'$, and the rest of the observations $\{\mathbf{Z}_i\}_{i=\tau}^{\infty}$ from a bivariate normal distribution with mean $\boldsymbol{\mu}_1 = (\mu, \mu)'$. The degree of change is defined by $\mu \in \{1, 1.5, 2, 2.5, 3\}$. The covariance matrix of the normal distribution is the identity matrix. Let $\tau = n + k$, after the τ observations, we continue data generation until the online detecting procedure detect a change or the number of the

Table 4. The OC-ARL and the probability of successful detection (shown in parenthesis) for depth method.

		k			
n	μ	1	3	5	10
50	1	13020 (0.432)	8859 (0.829)	6686 (0.930)	4953 (0.980)
	1.5	7910 (0.822)	932.5 (0.998)	360.6 (1)	157.2 (1)
	2	2607 (0.975)	63.8 (1)	29.2 (1)	23.3 (1)
	2.5	611.9 (0.999)	11.6 (1)	9.3 (1)	11.9 (1)
	3	58.6 (1)	4.7 (1)	5.7 (1)	10.2 (1)
100	1	13784 (0.677)	6239 (0.996)	4254 (0.989)	2903 (0.999)
	1.5	4284 (0.977)	321.9 (1)	155.4 (1)	104.4 (1)
	2	614.5 (0.999)	34.1 (1)	21.2 (1)	21.0 (1)
	2.5	91.4 (1)	8.1 (1)	7.6 (1)	11.7 (1)
	3	14.8 (1)	4.1 (1)	5.4 (1)	10.3 (1)

observations reaches RL_α. Since we only control the false alarms within the RL_α observations, a detection procedure fails if it cannot detect the change within RL_α observations. If we detect a change beyond RL_α points, we cannot distinguish it from a false alarm.

Table 4 displays the OC-ARL and the probability of a successful detection with 1000 replications of the experiment. We keep four significant figures for OC-ARL and the probability of a successful detection is shown in the parenthesis. Larger values of the probability of a successful detection and smaller values of the OC-ARL indicate better performance. From Table 4, we can see that increasing the width of the baseline window can increase the rate of successful detection and decreases the OC-ARL. In practice, when computing power allows, we should choose the baseline window as wide as possible for this model. We also need to verify that the observations in the baseline window are all from the same distribution. The influence of k on the detection performance is more complex. When the degree of the change (μ) is small, increasing the value of k leads to better performance of the detection procedure. However, for large changes, the depth method with larger k has larger OC-ARL on the average. For example, when $\mu = 1$, the depth model with $k = 10$ performs better than the model with $k = 3$, but when $\mu = 3$, the model with $k = 3$ does a better job.

6. Comparison

In this Section, we compare the sliding-window and depth models with two recently proposed multivariate online change detection methods under simulated data. Since the data are simulated, we know the actual change point and can use it to check the OC-ARL of different methods. All the OC-ARLs in the results are shown with 4 significant figures. The first method is the sum of local CUSUM method proposed by Mei [15]. Suppose $\{\mathbf{Z}_i\}_{i=1}^{\infty}$ is a d-dimensional data stream. We use $Z_i^{(j)}$ to represent the j-th variable of the observation $\mathbf{Z}_i$, for $i = 1, 2, \dots$ and $j = 1, \dots, d$. A local CUSUM method considers a single variable. For the j-th variable, suppose the marginal density is initially f_j. At some unknown time point, the density change to g_j. At time i, the local CUSUM statistic is calculated recursively by $S_j(i) = \max\{0, S_j(i-1) + \log \frac{g_j(Z_i^{(j)})}{f_j(Z_i^{(j)})}\}$ for $i \geq 1$, and $S_j(0) = 0$. By taking the summation, the statistic proposed by Mei [15] at time point i is defined as

$$\mathrm{CUSUM}_{\mathrm{SUM}}(i) = \sum_{j=1}^{d} S_j(i).$$

When $\mathrm{CUSUM}_{\mathrm{SUM}}(i)$ exceeds a predefined threshold h, an alarm is raised. The assumption of underlying distribution should be made to specify the test statistics. In this paper, the assumption of normality is made to derive the CUSUM statistics. Note that CUSUM can also be defined for any other distribution.

The second method is proposed by Zou et al. [30]. Instead of taking the sum of local CUSUM statistics directly, they develop goodness-of-fit tests of the local cumulative sum statistics. Let $U_j(i) = J(S_j(i))$ and $J(\cdot)$ denotes the cdf of $S_j(i)$ under the null hypothesis of no change. The statistic proposed by Zou et al. [30] is

$$\mathrm{CUSUM}_{\mathrm{GOF}}(i) = \sum_{j=1}^{d} \left\{ \log \left[\frac{U_{(j)}^{-1}(i) - 1}{(d-1/2)/(j-3/4) - 1} \right] \times I_{\{U_{(j)}(i) > (j-3/4)/d\}} \right\}^2,$$

where $U_{(1)}(i) \leq \cdots \leq U_{(d)}(i)$ are the order statistics of $\{U_j(i)\}_{j=1}^{d}$ and $I(\cdot)$ is the indicator function. The cdf function $J(\cdot)$ can be obtained by permutations or the approximation discussed by Grigg and Spiegelhalter [3]. A change is claimed when the $\mathrm{CUSUM}_{\mathrm{GOF}}(i)$ exceeds a predefined threshold h at time point i.

The third method for comparison is proposed by Chen et al. [1]. They developed a control chart based on the Wilcoxon rank-sum test. Consider the d-dimension data stream $\{\mathbf{Z}_i\}_{i=1}^{\infty}$. Suppose $Z_i^{(j)}$ is the j-th variable of the observation $\mathbf{Z}_i$, for $i = 1, 2, \ldots$ and $j = 1, \ldots, d$. The value of the multivariate exponentially weighted moving-average (MEWMA) control chart proposed by Chen et al. [1] at time point i is

$$\mathrm{MEWMA}(i) = \sum_{j=1}^{d} T_j^2(i),$$

where

$$T_j(i) = \sum_{t=i-\omega+1}^{i} (1-\lambda)^{i-t} \frac{R_{jit} - \omega(i+1)/2}{\sqrt{\omega(i+1)(i-\omega)/12}},$$

and ω is the window size, λ is the smoothing parameter, R_{jit} is the rank of $Z_t^{(j)}$ among $\{Z_1^{(j)}, \ldots, Z_i^{(j)}\}$. A change is claimed when $\mathrm{MEWMA}(i)$ exceeds a predefined threshold.

6.1. Mean change under multivariate normal distribution

To evaluate the power of the online methods for detecting changes in the mean of a data stream, we simulate the data stream under multivariate normal distribution with different mean vectors. We set RL_α to control false alarms and let n_1 be the length of the baseline window in sliding-window and depth models. We generate $n_1 + RL_\alpha + 500$ observations from multivariate normal distribution with mean vector $\boldsymbol{\mu}_0 = \mathbf{0}_d$ and covariance $\Sigma_0 = I_d$, where $\mathbf{0}_d$ is the zero vector and I_d is the $d \times d$ identity matrix. The first $n_1 + RL_\alpha$ observations are used for training the thresholds and the detection starts from observation number $n_1 + RL_\alpha + 1$. After the online detection procedure processes 500 observations, we change the mean vector to $\boldsymbol{\mu}_1 = \mu \mathbf{1}_d$. The degree of the changes is measured by the value of μ. We keep generating observations from the normal distribution with the new mean until the detection procedure claims a change. If the detection procedure does not claim any change within RL_α observations after the change occurs, we say that it fails to detect the change.

The settings for the sliding-window model and the depth model are $n_1 = n_2 = 50$, $n = 50$, $k = 5$ and $RL_\alpha = 5000$. The dimension varies in $d \in \{2, 10\}$. Each simulation is conducted over 1000 replications and we present the OC-ARL and the probability of successful detection in Table 5. When $\mu = 0$, the means of the distributions do not change. Hence, the probability of detection represents the false alarm rate. The IC-ARL is not applicable in the table because most of the times no change is claimed within the RL_α, so that we cannot know the run length for those situations.

From Table 5, we can see that all methods have acceptable abilities to prevent false alarms. In terms of the detection power, the multivariate CUSUM methods perform the best. This is expected because we build the CUSUM statistics based on the assumption of normality. Between the two proposed methods, the depth model is not competitive in detecting a change in small mean, but it catches up when the change is more pronounced.

Table 6 displays the performance of the methods when the variables are correlated. Suppose $d = 3$ and the correlation is constant so that Σ has 1s on the main diagonal and ρ on the off-diagonal where ρ is the correlation between the variables. We compare the performance of different model in detecting mean change with the influence of constant correlation matrix instead of the identity matrix. The correlation coefficient varies in $\rho \in \{0, 0.5, 0.9\}$. The other settings are the same as before. From the table, we can see that the correlations do not have much influence on the ability to prevent false alarms to the models. In terms of the detection power, all models are impacted, but the changes can still be captured to some degree. This is because the values from different dimensions tend to be similar due to high correlations. The competitiveness among the methods are similar to when no correlation occurs.

6.2. Variance change under multivariate normal distribution

To evaluate the above methods for detecting changes in variability, we simulate the data stream from multivariate normal distributions with different variances. Similar to the

Table 5. The ARL and the probability of successful detection (shown in parenthesis) for different methods under change in the mean.

d	μ	Sliding-window	Depth	CUSUM$_{\text{SUM}}$	CUSUM$_{\text{GOF}}$	MEWMA
2	0	NA (0.055)	NA (0.049)	NA (0.051)	NA (0.057)	NA (0.058)
	1	30.87 (1)	962.5 (0.894)	11.28 (1)	10.79 (0.998)	32.48 (1)
	2	17.96 (1)	10.40 (1)	4.24 (1)	4.586 (1)	11.17 (1)
	3	13.91 (1)	5.13(1)	2.82(1)	3.061 (1)	9.521 (1)
10	0	NA (0.055)	NA (0.053)	NA (0.046)	NA (0.043)	NA (0.056)
	1	18.70 (1)	29.47 (1)	3.668 (1)	4.506 (0.999)	10.56 (1)
	2	11.13 (1)	5 (1)	1.724 (1)	1.746 (0.999)	6.756 (1)
	3	8.714 (1)	5 (1)	1.055 (1)	1.020 (1)	5.873 (1)

Table 6. The ARL and the probability of successful detection (shown in parenthesis) for different method under change in the mean when the variable correlation is constant as ρ ($d = 3$).

ρ	μ	Sliding-window	Depth	CUSUM$_{\text{SUM}}$	CUSUM$_{\text{GOF}}$	MEWMA
0	0	NA (0.055)	NA (0.047)	NA (0.044)	NA (0.047)	NA (0.059)
	1	27.93 (1)	443.0 (0.983)	8.826 (1)	9.521 (0.999)	19.44 (1)
	2	15.73 (1)	7.040 (1)	3.297 (1)	3.502 (1)	9.642 (1)
0.5	0	NA (0.052)	NA (0.051)	NA (0.069)	NA (0.071)	NA (0.055)
	1	33.83 (1)	667.4 (0.941)	12.22 (1)	12.32 (0.999)	38.48 (1)
	2	18.45 (1)	13.69 (1)	4.673 (1)	4.791 (0.998)	11.87 (1)
0.9	0	NA (0.062)	NA (0.049)	NA (0.067)	NA (0.064)	NA (0.063)
	1	42.51 (1)	1226 (0.755)	16.38 (1)	16.51 (0.997)	107.8 (0.998)
	2	21.33 (1)	42.08 (1)	6.206 (1)	6.294 (0.999)	14.94 (1)

Table 7. The ARL and the probability of successful detection (shown in parenthesis) for different method under change in the variance.

d	σ	Sliding-window	Depth	CUSUM$_{\text{sum}}$	CUSUM$_{\text{GOF}}$	MEWMA
2	1	NA (0.053)	NA (0.052)	NA (0.043)	NA (0.061)	NA (0.059)
	2	1629 (0.593)	901.7 (0.945)	2563 (0.081)	2727 (0.103)	2386 (0.118)
	3	352.8 (1)	132.6 (1)	2281 (0.065)	2285 (0.081)	2389 (0.210)
	5	48.16 (1)	23.94 (1)	2232 (0.075)	2512 (0.116)	2290 (0.428)
	10	29.13 (1)	8.090 (1)	2396 (0.071)	2610 (0.091)	2012 (0.756)
10	1	NA (0.051)	NA (0.046)	NA (0.054)	NA (0.045)	NA (0.057)
	2	250.2 (1)	30.10 (1)	3049 (0.061)	3267 (0.070)	2098 (0.294)
	3	38.39 (1)	7.613 (1)	2382(0.095)	3044 (0.071)	2040 (0.470)
	5	25.44 (1)	5.025 (1)	2697 (0.055)	1866 (0.060)	1975 (0.597)
	10	17.25 (1)	5 (1)	2498 (0.065)	2328 (0.156)	1113 (0.958)

change in the mean scenario, we first generate $n_1 + RL_\alpha + 500$ observations from a multivariate normal distribution with mean $\boldsymbol{\mu}_0 = \mathbf{0}_d$ and covariance $\Sigma_0 = I_d$, where the first $n_1 + RL_\alpha$ observations are used for training the thresholds. After providing 500 observations from the standard normal distribution, we change the covariance $\Sigma_0 = I_d$ to $\Sigma_1 = \sigma I_d$ and where σ measures the degree of the change in the variance. The generation of the observations ends when a change is detected. A detection procedure fails when it does not detect any change within RL_α observations.

The settings in the change in the variance scenario are $n_1 = n_2 = n = 50,\ k = 5$ and $RL_\alpha = 5000$. Table 7 shows the OC-ARL and the probability of successful detection for different methods. Four significant figures are kept for OC-ARL. When $\sigma = 1$, no change occurs and the results represent the false alarm rate and the average running length to claim a false alarm. From the table, we can see that the ability to prevent false alarms are acceptable for all the methods. The results are expected because, under the null hypothesis of no changes, the distribution of the observations is the same as the distribution in Section 6.1 when $\mu = 0$. Consider the detection power of the methods. One can see that the CUSUM methods and MEWMA method can hardly detect any changes. The depth model performs the best in detecting the changes in the variance. It has the highest detecting rate and the least OC-ARL.

Table 8. The ARL and the probability of successful detection (shown in parenthesis) for different method under multivariate t-distribution.

d	ν	μ	Sliding-window	Depth	CUSUM$_{\text{sum}}$	CUSUM$_{\text{GOF}}$	MEWMA
2	3	0	NA (0.037)	NA (0.043)	NA (0.342)	NA (0.242)	NA (0.054)
		1	39.28 (1)	2086 (0.178)	53.37 (1)	58.78 (0.991)	108.3 (0.999)
		3	16.06 (1)	327.4 (0.891)	9.729 (1)	10.31 (0.992)	10.93 (1)
	5	0	NA (0.034)	NA (0.062)	NA (0.028)	NA (0.025)	NA (0.040)
		1	36.08 (1)	1874 (0.310)	25.26 (1)	26.89 (0.998)	58.38 (1)
		3	14.54 (1)	22.86 (1)	3.708 (1)	4.377 (0.994)	10.23 (1)
10	3	0	NA (0.044)	NA (0.043)	NA (0.459)	NA (0.554)	NA (0.038)
		1	23.23 (1)	2455 (0.118)	59.05 (1)	63.75 (0.997)	14.99 (1)
		3	10.38 (1)	177.8 (0.926)	4.460 (1)	5.806 (0.979)	6.689 (1)
	5	0	NA (0.039)	NA (0.053)	NA (0.247)	NA (0.204)	NA (0.047)
		1	20.92 (1)	1859 (0.337)	10.83 (1)	12.18 (0.998)	12.84 (1)
		3	9.552 (1)	5.051 (1)	1.744 (1)	2.411 (0.992)	6.318 (1)

6.3. Parameter change under multivariate t-distribution.

To assess the performance of the models under non-normal distributions, we first consider the parameter change under multivariate t-distribution. Suppose $\mathbf{x}$ and u are independent and distributed as $N(\mathbf{0}, \Sigma)$ and χ^2_ν, respectively. The random vector $\mathbf{y} = \frac{\mathbf{x}-\boldsymbol{\mu}}{\sqrt{u/\nu}}$ follows a multivariate t-distribution with ν degrees of freedom where $\boldsymbol{\mu}$ and Σ are the location vector and shape matrix of the t-distribution, respectively.

We consider the change of the location vector while keeping the degree of freedom and the shape matrix constant. Specifically, the location vectors varies in $\{\mathbf{0}_d, \mathbf{1}_d, \mathbf{3}_d\}$. The mean vector and the covariance matrix are defined for the multivariate t-distribution when $\nu > 2$. Thus, we consider the degree of freedom from $\{3, 5\}$. The dimension varies in $d \in \{2, 10\}$. The shape matrix is the identity matrix in this simulation.

Table 8 shows the ARLs and the probability of successful detection under multivariate t-distribution. From the table, we can see that the CUSUM methods cannot maintain their abilities to prevent false alarms while the other three methods perform well when there is no actual change. In terms of the power, there is not a single method that dominates others in all situations. Depth model is not competitive when d and μ are small, but catches up in high dimension and severe change cases. In real applications, a detection system that combines multiple methods is recommended in order to obtain better performance.

6.4. Parameter change under multinomial distribution

We also consider the cases where the observations follow multinomial distribution. The dimension varies in $d \in \{5, 10\}$. For IC status, the observations follow multinomial distribution with parameter $N = 100, \boldsymbol{p}_x = \frac{1}{d}\mathbf{1}_d$, denoted as $MT(N = 100, \boldsymbol{p}_x = \frac{1}{d}\mathbf{1}_d)$. The OC distribution is $MT(N = 100, \boldsymbol{p}_y)$, where $\boldsymbol{p}_y$ is defined as follows. Let p_{yi} be the i-th component of the vector $\boldsymbol{p}_y$. We define $p_{y1} = \xi$, and for $i = 2, \ldots, d-1$, $p_{yi} = \frac{1-\xi}{d-1}$. The value of ξ measures the degree of change. We choose ξ in $\{0.3, 0.5, 0.7, 0.9\}$. The other settings are as before so $n_1 = n_2 = 50$, $n = 50$, $k = 5$ and $RL_\alpha = 5000$.

Table 9 shows the ARLs and the probability of successful detection for different methods. When $d = 5, \xi = 0.2$ represent the IC status while for $d = 10, \xi = 0.1$ represent the IC status. From the table, we can see that the CUSUM methods cannot detect the changes

Table 9. The ARL and the probability of successful detection (shown in parenthesis) for different method under multinormial distribution.

d	ξ	Sliding-window	Depth	CUSUM$_{\text{sum}}$	CUSUM$_{\text{GOF}}$	MEWMA
5	0.2	NA (0.042)	NA (0.051)	NA (0)	NA (0)	NA (0.058)
	0.3	21.08 (1)	73.48 (1)	NA (0)	NA (0)	11.74 (1)
	0.5	9.954 (1)	5 (1)	NA (0)	NA (0)	6.174 (1)
	0.7	7.579 (1)	5 (1)	NA (0)	NA (0)	5.490 (1)
	0.9	6.581 (1)	5 (1)	NA (0)	NA (0)	5.512 (1)
10	0.1	NA (0.043)	NA (0.054)	NA (0)	NA (0)	NA (0.049)
	0.3	10.87 (1)	5.02 (1)	NA (0)	NA (0)	6.778 (1)
	0.5	7.329 (1)	5 (1)	NA (0)	NA (0)	4.460 (1)
	0.7	6.048 (1)	5 (1)	NA (0)	NA (0)	3.855 (1)
	0.9	5.507 (1)	5 (1)	NA (0)	NA (0)	3.905 (1)

and show NA on the table, while the other three methods are working. The reason of the NA is that the CUSUM methods will lose their power when the distributional assumption no longer holds. The other three methods do not rely much on the specification of a distribution. Among the three workable methods, the depth model is more powerful under some situations. However, its OC-ARL is limited by the parameter k, which is the number of the consecutive excesses. The MEWMA method beats the proposed methods in some cases because it is sensitive to the change in the mean of multinomial distribution.

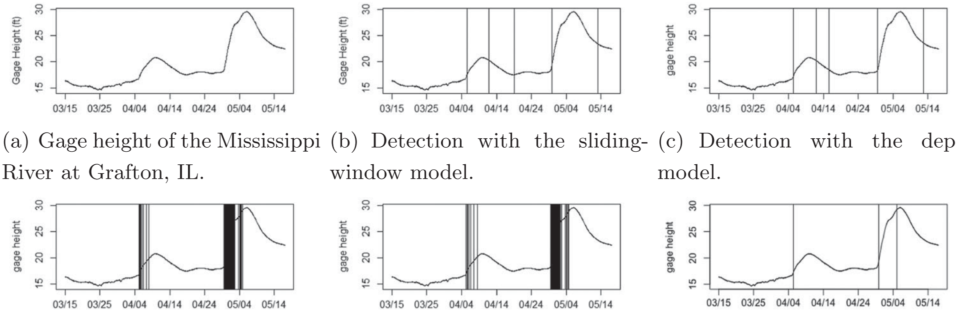
7. Application

We apply the proposed and existing methods of online change detection to data collected by the US Geological Survey (USGS) on the flowing volume of the Mississippi River. The data are collected for flood preparedness. Because it is hard to measure the flowing volume of a river, USGS records the gage height of a river on different stations. We consider the gage heights of the Mississippi River at Grafton, IL. Figure 1(a) shows the gage heights of the Mississippi River at Grafton from March 15, 2017 to May 15, 2017. The data is collected every half hour for 24 h a day. Missing values are replaced by their neighbors' average. There are 3024 data points available in total for this application study.

We apply the sliding-window and the depth model on the movement of gage heights. A change is claimed when the gage heights differ rapidly. The detection results are shown in Figure 1. For the sliding-window model, the width of the baseline and current windows are set as $n_1 = n_2 = 100$. The running length is $AL_\alpha = 500$, where $\alpha = 0.05$. Before the detection procedure, we train the threshold with 100 permutations of the first 600 observations. Thus, the actual change detection starts from March 27, 2017.

Figure 1(b) shows the detection results of the sliding the window model. Each vertical line represents an alarm for change. The procedure detects both changes when the gage height increases rapidly (first and fourth detection). We consider the second, third and fifth detections as false alarms because there is no need to alarm when the gage height decrease or stay stable after a change. However, in terms of the data itself, these detections are reasonable.

For the depth model, we set $n = 100$ and $k = 5$. The running length and the training sample size are the same as the sliding-window model. The detection result is shown in Figure 1(c) where one sees that the depth model also detects the two major increases of the



(a) Gage height of the Mississippi River at Grafton, IL.

(b) Detection with the sliding-window model.

(c) Detection with the depth model.

(d) Detection with the CUSUM statistic.

(e) Detection with the GOF test of CUSUM.

(f) Detection with MEWMA.

Figure 1. Online change detections over the gage heights of the Mississippi River. (a) Gage height of the Mississippi River at Grafton, IL. (b) Detection with the sliding-window model. (c) Detection with the depth model. (d) Detection with the CUSUM statistic. (e) Detection with the GOF test of CUSUM and (f) Detection with MEWMA.

gage heights. The OC-ARL for the second increase is less than the sliding-window model. The third detection of the depth model is a false alarm because the movement of the gage height does not change much. While the depth model is more accurate in terms of OC-ARL it is not as robust as the sliding-window model.

Figure 1(d,e) shows the detection results using CUSUM and the GOF test of CUSUM. When the gage height increases continuously, these two methods will keep claiming changes until the increase slows down. That is the reason for the black areas in Figure 1(d,e). Figure 1(f) shows the detection result with MEWMA. The method captures the two major increases as the other methods.

8. Discussion and summary

The online change point problem is different from offline change point problem because the total number of the observations is unknown and could be infinite. Issues like computing and memory limitations and multiple testing will occur if we use offline methods to do online change detection. In this paper, we discuss two online change detection models: the sliding-window model with energy statistic and the depth model with Mahalanobis distance. For each model, we propose detection algorithms and training methods to compute the threshold. We conduct numerical studies under bivariate normal distributions and discuss the influence of the window width on both models. Changing the window width will affect the performance of the sliding-window models, but barely influences the depth model. We also compare the proposed models with two multivariate CUSUM models and a multivariate distribution-free model under different type of changes. Both methods show their competitiveness under some scenarios. In practice, we suggest to have a combination of different methods when facing an online detection problem. For example, a majority ensemble rule claims a change if the majority of the detection models claim a change.

In this work, we assumed that the observations of the data stream are independent and identically distributed under the hypothesis of no change. In practice, the vector observations may not be independent. It is important to account for possible correlations among the observations under some time series models. A method that is currently under investigation is to embed the observation vectors within matrices and detect changes in the matrix observations.

Acknowledgments

We thank two anonymous referees and the associate editor for constructive comments.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Reza Modarres http://orcid.org/0000-0003-1240-6027

References

[1] N. Chen, X. Zi, and C. Zou, *A distribution-free multivariate control chart*, Technometrics 58 (2016), pp. 448–459.

[2] R.B. Crosier, *Multivariate generalizations of cumulative sum quality-Control schemes*, Technometrics 30 (1988), pp. 291–303.

[3] O.A. Grigg and D.J. Spiegelhalter, *An empirical approximation to the null unbounded steady-state distribution of the cumulative sum statistic*, Technometrics 50 (2008), pp. 501–511.

[4] C. Hartland, N. Baskiotis, S. Gelly, M. Sebag, and O. Teytaud, *Change point detection and meta-bandits for online learning in dynamic environments*, CAp, Grenoble, France, 2007, pp. 237–250.

[5] H.E.T. Holgersson and P.S. Karlsson, *Three estimators of the Mahalanobis distance in high-dimensional data*, J. Appl. Stat. 39 (2012), pp. 2713–2720.

[6] Y. Kawaharal and M. Sugiyama, *Fast and accurate detection of changes in data streams*, Stat Anal Data Mining 7 (2014), pp. 125–139.

[7] D. Kifer, S. Ben-David, and J. Gehrke, *Detecting change in data streams*, in Proceedings of the 30th VLDB Conference, Vol. 30, 2004, pp. 180–191,

[8] W. Li, X. Pu, F. Tsung, and D. Xiang, *A robust self-starting spatial rank multivariate EWMA chart based on forward variable selection*, Comput. Industr. Engin. 103 (2017), pp. 116–130.

[9] W. Li, D. Xiang, F. Tsung, and X. Pu, *A diagnostic procedure for high-dimensional data streams via missed discovery rate control*, Technometrics 62 (2020), pp. 84–100.

[10] Z. Liu and R. Modarres, *Lens data depth and median*, J. Nonparametric Stat. 23 (2011), pp. 1063–1074.

[11] R.Y. Liu and K. Singh, *A quality index based on data depth and multivariate rank tests*, J. Am. Stat. Assoc. 88 (1993), pp. 252–260.

[12] R. Malladi, G.P. Kalamangalam and B. Aazhang, *Online Bayesian change point detection algorithms for segmentation of epileptic activity*, In 2013 Asilomar Conference on Signals, Systems and Computers, 2013, pp. 1833–1837.

[13] A. Messaoud, C. Weihs, and F. Hering, *A Nonparametric Multivariate Control Chart Based on Data Depth*, Technical Report, No. 2004,61, Universität Dortmund, 2004.

[14] D.S. Matteson and N.A. James, *A nonparametric approach for multiple change point analysis of multivariate data*, J. Am. Stat. Assoc. 109 (2014), pp. 334–345.

[15] Y. Mei, *Efficient scalable schemes for monitoring a large number of data streams*, Biometrika 97 (2010), pp. 419–433.

[16] J. Mellor and J. Shapiro, *Thompson sampling in switching environments with Bayesian online change detection*. In 2013 Artificial Intelligence and Statistics, 2013, pp. 442–450,

[17] K. Miyaguchi and K. Yamanishi, *Online detection of continuous changes in stochastic processes*, Inter. J. Data Sci. Anal. 3 (2017), pp. 213–229.

[18] R. Modarres and G.P. Patil, *Hotspot detection with bivariate data*, Planning and Inference J. Stat. Plan. Inference. 137 (2007), pp. 3643–3654.

[19] E.S. Page, *Continuous inspection schemes*, Biometrika 41 (1954), pp. 100–115.

[20] P. Qiu, *Introduction to Statistical Process Control*, Chapman and Hall/CRC, Florida, 2013.

[21] G.J. Székely and M.L. Rizzo, *Energy statistics: A class of statistics based on distances*, Statistical Planning and Inference¡/DIFdel¿J. Stat. Plan. Inference. 143 (2013), pp. 1249–1272.

[22] P. Yang, G. Dumont, and J.M. Ansermino, *An adaptive Cusum test based on a hidden semi-Markov model for change detection in non-invasive mean blood pressure trend*. In 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, 2006, pp. 3395–3398,

[23] A. Wald, *Sequential Analysis*, 1st ed., John Wiley and Sons, New York, 1947.

[24] W. Wei, J. Li, L. Cao, Y. Ou, and J. Chen, *Effective detection of sophisticated online banking fraud on extremely imbalanced data*, World Wide Web 16 (2013), pp. 449–475.

[25] Y. Wu, *Streaming techniques for statistical modeling*, Doctoral dissertation, Rutgers University-Graduate School-New Brunswick, 2007.

[26] S. Yildirim, S.S. Singh, and A. Doucet, *An online expectation–maximization algorithm for changepoint models*, J. Comput. Graph. Stat. 22 (2013), pp. 906–926.

[27] C. Zhang, N. Chen, and J. Wu, *Spatial rank-based high-dimensional monitoring through random projection*, J. Qual. Technol. 52 (2020), pp. 111–127.

[28] C. Zou, W. Jiang, and F. Tsung, *A lasso-based diagnostic framework for multivariate statistical process control*, Technometrics 53 (2011), pp. 297–309.

[29] C. Zou and F. Tsung, *A multivariate sign EWMA control chart*, Technometrics 53 (2011), pp. 84–97.

[30] C. Zou, Z. Wang, X. Zi, and W. Jiang, *An efficient online monitoring method for high-dimensional data streams*, Technometrics 57 (2015), pp. 374–387.

[31] S.T. Bakir, and M.R. Reynolds, *A nonparametric procedure for process control based on within-group ranking*, Technometrics 21 (1979), pp. 175–183.

[32] S. Chakraborti, and M.A. Van de Wiel, *A Nonparametric Control Chart Based on the Mann-Whitney Statistic*, Institute of Mathematical Statistics, Ohio, 2008.

[33] J. Li, *Efficient global monitoring statistics for high?dimensional data*, Quality and Reliability Engineering International 36 (2020), pp. 18–32.

Appendix

Proof of Lemma 3.1.:

$$
\begin{aligned}
L_{t+1} &= 2(n_1n_2)^{-1}\sum_{i=1}^{n_1}\sum_{j=n_1+t+1}^{n_1+t+n_2}\|\mathbf{Z}_i-\mathbf{Z}_j\| - \binom{n_1}{2}^{-1}\sum_{i=1}^{n_1-1}\sum_{j=i+1}^{n_1}\|\mathbf{Z}_i-\mathbf{Z}_j\| \\
&\quad - \binom{n_2}{2}^{-1}\sum_{i=n_1+t+1}^{n_1+t+n_2-1}\sum_{j=i+1}^{n_1+t+n_2}\|\mathbf{Z}_i-\mathbf{Z}_j\| \\
&= 2(n_1n_2)^{-1}\sum_{i=1}^{n_1}\left\{\sum_{j=n_1+t}^{n_1+t+n_2-1}\|\mathbf{Z}_i-\mathbf{Z}_j\| + \|\mathbf{Z}_i-\mathbf{Z}_{n_1+t+n_2}\| - \|\mathbf{Z}_i-\mathbf{Z}_{n_1+t}\|\right\} \\
&\quad - \binom{n_2}{2}^{-1}\left\{\sum_{i=n_1+t}^{n_1+t+n_2-2}\sum_{j=i+1}^{n_1+t+n_2-1}\|\mathbf{Z}_i-\mathbf{Z}_j\| - \sum_{i=n_1+t+1}^{n_1+t+n_2-1}\|\mathbf{Z}_i-\mathbf{Z}_{n_1+t+n_2}\|\right. \\
&\quad \left. + \sum_{i=n_1+t+1}^{n_1+t+n_2-1}\|\mathbf{Z}_i-\mathbf{Z}_{n_1+t}\|\right\} - \hat{\mu}_{BB}
\end{aligned}
$$

$$= 2\hat{\mu}_{BC} - \hat{\mu}_{BB} - \hat{\mu}_{CC} + (n_1 + n_2)^{-1} \sum_{i=1}^{n_1} \left\{ \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t+n_2}\| - \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t}\| \right\}$$

$$+ \binom{n_2}{2}^{-1} \sum_{i=n_1+t+1}^{n_1+t+n_2-1} \left\{ \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t+n_2}\| - \|\mathbf{Z}_i - \mathbf{Z}_{n_1+t}\| \right\}. \qquad \blacksquare$$

Proof of Theorem 4.1.: Suppose $\boldsymbol{\mu}_F$ and $\boldsymbol{\Sigma}_F$ are the mean vector and the dispersion matrix of distribution F. One can show that $\bar{\mathbf{Z}} \xrightarrow{p} \boldsymbol{\mu}_F$, and $S_Z \xrightarrow{p} \boldsymbol{\Sigma}_F$, as $n \to \infty$. Thus,

$$(\mathbf{Z} - \bar{\mathbf{Z}})S_Z^{-1}(\mathbf{Z} - \bar{\mathbf{Z}}) \xrightarrow{p} (\mathbf{Z} - \boldsymbol{\mu}_F)\boldsymbol{\Sigma}_F^{-1}(\mathbf{Z} - \boldsymbol{\mu}_F), \text{as} n \to \infty. \tag{A1}$$

Since F is a multivariate normal distribution, we have

$$(\mathbf{Z} - \boldsymbol{\mu}_F)\boldsymbol{\Sigma}_F^{-1}(\mathbf{Z} - \boldsymbol{\mu}_F) \sim \chi_d^2. \tag{A2}$$

By Equations (A1) and (A2), we have

$$(\mathbf{Z} - \bar{\mathbf{Z}})S_Z^{-1}(\mathbf{Z} - \bar{\mathbf{Z}}) \xrightarrow{p} \mathbf{X}, \text{as} n \to \infty. \tag{A3}$$

where $\mathbf{X} \sim \chi_d^2$. By equations (6) and (A3), we have $D_{Mah}(\mathbf{Z}|F_n) \xrightarrow{p} \mathbf{Y}, \text{as} n \to \infty$ where $\mathbf{Y} = (1 + \mathbf{X})^{-1}$. ■

Proof of Theorem 4.2.: By Algorithm 3, the *baseline window* is $\mathscr{B} = \{\mathbf{Z}_i\}_{i=1}^n$. Consider the sequence $\{D_{Mah}(\mathbf{Z}_{n+1}|F_n), D_{Mah}(\mathbf{Z}_{n+2}|F_n), \ldots\}$. Let

$$MAX_s = \max(D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}|F_n), \ldots, D_{Mah}(\mathbf{Z}_{n+sk}|F_n)),$$

for $s = 1, 2, \ldots$. Since the observations in the data stream are i.i.d, the probability of not declaring a change at the s-th consecutive observations is

$$\begin{aligned}
P(MAX_s > h) &= P\{\max(D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}|F_n), \ldots, D_{Mah}(\mathbf{Z}_{n+sk}|F_n)) > h\} \\
&= 1 - P\{\max(D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}|F_n), \ldots, D_{Mah}(\mathbf{Z}_{n+sk}|F_n)) \le h\} \\
&= 1 - P\{D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}|F_n) \le h, \ldots, D_{Mah}(\mathbf{Z}_{n+sk}|F_n) \le h\} \\
&= 1 - (P\{D_{Mah}(\mathbf{Z}_{n+(s-1)k+1}|F_n) \le h\})^k \\
&= 1 - (P\{D_{Mah}(\mathbf{Z}_{n+1}|F_n) \le h\})^k
\end{aligned}$$

Thus, if the running length (RL) is larger than t, it follows that $MAX_s > h$ for at least t/k terms. Because RL_α is the α-th quantile of the running length (RL), we have

$$\begin{aligned}
1 - \alpha &= P(RL > RL_\alpha) = P(MAX_s > h, \text{for} s = 1, \ldots, RL_\alpha/k) \\
&= [P(MAX_s > h)]^{RL_\alpha/k} = [1 - (P\{D_{Mah}(\mathbf{Z}_{n+1}|F_n) \le h\})^k]^{RL_\alpha/k}
\end{aligned}$$

With some algebra, we have $P\{D_{Mah}(\mathbf{Z}_t|F_n) \le h\} = c$. Therefore, h is the c-th quantile of $D_{Mah}(\mathbf{Z}_t|F_n)$. Since, $D_{Mah}(\mathbf{Z}_{n+1}|F_n) \xrightarrow{p} \mathbf{Y}$, h converge to the c-th quantile of Y in probability when $n \to \infty$. ■

Proof of Corollary 4.1.: Note that the value of c is determined by RL_α, and RL_α is influenced by h. Thus, the value of c is not constant and is related to h. Let $Y = (1 + X)^{-1}$ and $X \sim \chi_d^2$. By Theorem 4.2, we have $P(Y < h) \approx c$. Therefore, one can show that

$$\begin{aligned}
c &\approx P(Y < h) = P((1 + X)^{-1} < h) \\
&= P(1 + X > 1/h) = P(X > 1/h - 1) = 1 - P(X \le 1/h - 1)
\end{aligned}$$

It follows that $P(X \le 1/h - 1) = 1 - c$, which leads to $1/h - 1 \approx \chi_d^2(1 - c)$. Thus, the threshold h can be approximated by $(1 + \chi_d^2(1 - c))^{-1}$. ■