A Brief Comparison of Algorithms for Detecting Change Points in Data

Cody Buntain
Department of Computer Science
University of Maryland
College Park, MD 20742
cbuntain@cs.umd.edu

Christopher Natoli
Department of Statistics
University of Chicago
5801 S Ellis Ave, Chicago, IL 60637
chrisnatoli@gmail.com

Miroslav Živković
Institute of Informatics
University of Amsterdam
1012 WX Amsterdam, Netherlands
m.zivkovic@uva.nl

***Abstract*—Detecting points in data where the underlying distribution changes is not a new task, but much of the existing literature assumes univariate and independent data, assumptions often violated in real data sets. This work addresses this gap in the literature by implementing a set of change point detection algorithms and a test harness for evaluating their performance and relative strengths and weaknesses in multi-variate data of varying dimension and temporal dependence. We then apply our implementations to real-world data taken from structural sensors placed on laboratory a bridge and two years of Bitcoin market data from the Mt. Gox exchange. Though more work is necessary to explore these real-world data sets more thoroughly, our results demonstrate circumstances in which an online, non-parametric algorithm does and does not perform as well as offline, parametric algorithms and provides an early foundation for future investigations.**

I. Introduction

In many applications where monitoring plays a role (Internet, smart energy grids, reservoir engineering), it is crucial to be able to detect points in time in which anomalies occur (indicating potential failures or attacks). This task often involves multidimensionality (more than one sensor) with dependence between sensors and time. Many of the well-known anomaly detection methods, however, assume one-dimensional and independent input. These assumptions are of course too unrealistic in many cases, and straightforward application to more complex problems frequently leads to erroneous results. Dependence is thus a common characteristic of multidimensional data (measurement). There exist several generic models known to have good capabilities to model data originating from multidimensional and dependent measurements, especially multidimensional autoregressive, moving average (ARMA) time series models, which are popular in econometrics research. Existing literature on machine learning techniques show that violated model assumptions do not necessarily obviate the efficacy of a given algorithm. As such, the research documented herein compares three algorithms for detecting change points, two of which are parametric and make assumptions of the underlying data, and the third is non-parametric and does not try to model underlying data distributions. Additionally, our investigation focuses on multidimensional data that exhibit "abrupt" change points and can include multiple change points over time.

We start by giving a short overview of generic models for multidimensional data in Section 2, describing in particular the multidimensional ARMA processes. Then we will give a literature overview of change point detection methods for time series data, focusing in particular on multidimensional data, in Section 3. From this overview we will describe the methods we have chosen to implement in more detail in Section 4. Section 5 then covers the comparative performance among the algorithms we implemented before we close in Section 6.

II. Generic Models For Multidimensional Time Series Data

A. Auto-Regressive Moving Average (ARMA) models

1) One-dimensional ARMA: We will start our overview with one-dimensional ARMA model, which will be then generalized to the multidimensional one in Section II-A2

The Auto-Regressive (AR) model is a random process that describes the behavior of a variable in terms of its own past values. The pth-order auto-regressive or AR(p) process is defined by:

$$Y_t = \mu + \gamma_1 Y_{t-1} + \gamma_2 Y_{t-2} + \ldots + \gamma_p Y_{t-p} + \varepsilon_t,$$

where μ is an intercept, $Y_{t-1}, \ldots, Y_{t-p}$ are the lag values of the variable Y_t, and $\gamma_1, \ldots, \gamma_p$ the corresponding parameters, while $(\varepsilon_t)_t$ is a white noise sequence, also known as innovation or error term.

Next, consider the symmetrical model: the qth-order Moving Average or MA(q) process is described by the following equation:

$$Y_t = \mu + \varepsilon_t \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q},$$

where μ is an intercept, $\varepsilon_{t-1}, \ldots, \varepsilon_{t-p}$ are the lag values of the innovation ε_t, and $\theta_1, \ldots, \theta_q$ the corresponding parameters.

Many (stationary) time series processes can be represented in either an auto-regressive or moving average form. Furthermore, researchers have found that a combination of both forms, that is, the Auto-Regressive Moving Average process (ARMA), has proven to be quite effective in modelling many real-life data sets [1]. The (p, q)th-order Auto-Regressive Moving Average or ARMA(p, q) process is given by:

$$Y_t = \mu + \gamma_1 Y_{t-1} + \gamma_2 Y_{t-2} + \ldots \\ +\gamma_p Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots - \theta_q \varepsilon_{t-q}.$$

For the discussion on applicability of ARMA model, along with the important conditions like stationarity, invertibility or causality the reader is referred to [1]

The models discussed above can only be applied to stationary data. However, a non-stationary data series can often be transformed into a stationary one by taking first differences, that is, by subtracting Y_{t-1} from Y_t. A non-stationary series that becomes stationary after being first differenced d times is referred to as integrated of order d, denoted by $I(d)$. An $I(1)$ series in its un-differenced form will typically be constantly growing, while an $I(2)$ series is growing at an ever-increasing rate. Series that are $I(3)$ or greater are extremely unusual, but they do exist.

Using this insight, the ARMA models defined above can be generalized to be able to deal with non-stationary series. The resulting model is denoted an Auto-Regressive Integrated Moving Average model, or ARIMA(p, d, q). Let L be a lag operator such that $LY_t = Y_{t-1}$. In full, the model is then denoted by

$$\Delta^d Y_t = \mu + \gamma_1 \Delta^d Y_{t-1} + \gamma_2 \Delta^d Y_{t-2} + \ldots \\ +\gamma_p \Delta^d Y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \ldots \\ -\theta_q \varepsilon_{t-q},$$

where

$$\Delta^d Y_t = (1 - L)^d Y_t.$$

2) The Multivariate Auto-Regressive Moving Average (VARMA) model: Let $(Y_t)_t$ be the observation vectors, having dimensions r. Further, define z^{-i} as a backward shift operator: $z^{-i} Y_t = Y_{t-i}$ for all $i \in \mathbb{N}_+$. Then, the vector auto-regressive moving average model has a form:

$$A(z) Y_t = c + B(z) \varepsilon_t, \tag{1}$$

where A, B are polynomial matrices in the backward shift operator z^{-i}:

$$A(z) = I_r - \sum_{i=1}^{p} A_i z^{-i}$$
$$B(z) = I_r - \sum_{j=1}^{q} B_j z^{-j}$$

where A_i, B_j are $r \times r$ matrices, c is a r-dimensional constant vector, $(\varepsilon_t)_t$ is a r-dimensional vector white noise sequence with covariance matrix R and I_r is r-dimensional identity matrix. VARMA can model at the same time a dependency with respect to time index for a certain observation Y_t^k but also a dependency across the observations Y_t^k, Y_t^l.

The series Y_t is stationary if $|A(x)| \neq 0$ for all $|x| = 1$ and it is unit-root non-stationary if $|B(1)| = 0$. Assuming there is no input vector, the auto-regressive representation of model (1) is given by $\Pi(z) Y_t = \varepsilon_t$, where $\Pi(z) = B(z)^{-1} A(z) = I - \sum_{i=1}^{\infty} \Pi_i z^{-i}$. Note that the second equality only holds under the assumption that $A_0 = B_0 = I$. Under the same assumptions, we also have the moving-average representation, $Y_t = \Psi(z) \varepsilon_t$, where $\Psi(z) = I + \sum_{i=1}^{\infty} \Psi_i z^{-i}$ is defined by $A(z)\Psi(z) = B(z)$.

III. Related Work

In this section we give a general overview of the methods of anomaly detection for multidimensional models, not necessary limiting ourselves only to the models described in Section II. We focus here on a particular class of anomaly detection known as *change point detection*. A change-point in a time series (a data set with a time component) is a moment in which the underlying probability distribution changes. This means that we focus on data that evolve over time, showing sudden and insistent changes at one (or multiple) points in time. The aim is to detect when a null hypothesis H_0 is rejected, where H_0 assumes no changes occur in distribution, or in a specific part of the distribution. For instance, certain methods focus on detecting changes in mean, or change in variance. The point is not only to detect the *existence* of such a change point, but also its location (in time). There are quite a lot of different methods for change point detection in the literature, each with their own assumptions on the time series. The emphasis found in the literature is often put on one-dimensional time series, but our main focus will be multidimensional, as well as possible dependence *between* time steps. A good survey for different techniques of detecting change points in multidimensional time series is given in [2].

The most widely used change point detection methods, also for the multivariate case, are CUSUM and GLR algorithms and their derivatives.

A. CUSUM

CUSUM stands for Cumulative Sum. The basic CUSUM method detects shifts in mean for independent sequences in the 1-dimensional case, where the densities before and after the change are assumed to be known. There are many useful extensions to the basic CUSUM method; we mention several here, but for a broader overview see [2]. We first briefly describe the basic CUSUM method. Assume $(X_i)_{i=0}^n$ is a 1-dimensional time series. We consider the following two hypotheses:

- Null-hypothesis H_0: the $X_i (i = 0, \ldots, n)$ are i.i.d. realizations of a random variable with density f.
- Alternative hypothesis $H_1(k)$: there is a $1 \leq k \leq n$ such that the observations are i.i.d. samples from a distribution with density f up to $k-1$, while from observation k on they are i.i.d. with a different density g.

The basic CUSUM method is an online method, meaning that every time new observations are coming in, the possibility of a change point is checked. The log likelihood ratio (LLR) is used to compute the test statistic: $t_n := \max_{k \in \{1,\ldots,n\}} S_{k,n}$, where $S_{k,n} = \sum_{i=k}^{n} \log(\frac{g(X_i)}{f(X_i)})$. For a given threshold c, the CUSUM method raises an alarm for the first n_0 for which $t_{n_0} > c$.

The authors of [3] found that so-called *multi-chart CUSUM* to perform better in detecting shifts in mean than the original CUSUM but to be less computationally expensive than the original GLR (see below). The multi-chart CUSUM consists of several basic CUSUM charts with different reference values that are used simultaneously to detect the mean shift. For example, let the anticipated interval of the mean shift, θ, be in $[a, b]$. Then, we can create a CUSUM multi-chart with a number of CUSUM charts $T_C(\varrho_1), \ldots, T_C(\varrho_m)$ by choosing the parameters $\varrho_1, \ldots, \varrho_m$ in the interval. If one of the CUSUM charts triggers a mean shift, the CUSUM multi-chart would send an alarm. An optimal design of the CUSUM multi-chart is provided in [3].

Also, the CUSUM methodology has been applied to multidimensional problems. A multivariate version of CUSUM, the *MCUSUM*, is given in [4] (see [5] for the example of implementation). The method is sensitive to small and moderate shifts. In [4] two examples are discussed:

- shift in mean of multivariate normal (MCUSUM reduces to a set of univariate CUSUMs),
- shift in covariance for multivariate normal.

The method requires tuning to the size of the shift and should only be used if the mean of the anomalous state (or at least its direction compared to non-anomalous state) is known. If the specified direction of the shift is wrong, performance can be bad.

Independence between timesteps is assumed in the previous. In [6] a CUSUM procedure is introduced for detecting covariance changes in multivariate ARMA processes, so with possible dependence between time steps.

While in the previous framework the assumption is made that pre- and post-change densities are known, Tartakovsky *et al.* [7] propose a nonparametric (pre- and post-change densities are unknown) multi-chart CUSUM to detect a change in one out of N independent populations each consisting of a possibly dependent sequence of random variables (the *MNPA-CUSUM*). Since in the nonparametric case the log-likelihood ratio (LLR) cannot be computed, the authors suggest to replace them by some 'appropriate' score functions that have a negative mean before the changepoint and a positive mean after the changepoint. In this case the CUSUM type statistic remains close to zero or slightly negative in normal conditions while after a changepoint it is drifting upward until it crosses a certain threshold. Indeed, score functions can be chosen in many ways, and their selection depends crucially on the type of change that we intend to detect. For example, different score functions are used to detect changes in the mean and changes in the variance. Tartakovsky *et al.* do not explain how to choose the score function in general, but give an example for detection of abrupt change in mean (other parameters can also change but are treated as nuisance parameters). Pre-change mean values μ_i of observed data are assumed to be known (or at least estimated accurately) in advance (but should be re-estimated once in a while). Post-change mean values θ_i should be estimated on-line or pre-set to a reasonable number, depending on the applications (see end of Section 3.1 in [8] for more details and concrete examples).

In [9], *a non-parametric* version of CUSUM is introduced that can be used to detect (multiple) changes in the covariance structure of multidimensional, possibly non-linear time series. The authors consider the null hypothesis

$$H_0 : \text{Cov}(Y_1) = \ldots = \text{Cov}(Y_n),$$

where $(Y_j, j = 1, \ldots, n)$ is a sequence of d-dimensional random vectors with finite first and second moments where the mean is constant over time. As the alternative hypothesis the authors specify

$$H_0 : \text{Cov}(Y_1) = \ldots = \text{Cov}(Y_k) \neq \text{Cov}(Y_{k+1}) = \ldots = \text{Cov}(Y_n).$$

The asymptotic null distribution is related to the Brownian bridge after rescaling, from which thresholds can be computed.

In [10] a non-parametric CUSUM-like method is described for detecting changes in correlation matrices for multidimensional time series data with some possible dependence. The thresholds are computed via Brownian bridges.

In [11] a test is described for detecting change in correlation between two random variables X_t and Y_t, so for the bivariate setting. The bivariate vector (X_t, Y_t) is assumed to have finite first four moments, and some serial dependence is allowed. In a higher dimensional setting, this test might be applied for each of the pairs, leading to $d(d-1)/2$ different tests in the d-dimensional case. Similar as in [9] the asymptotic null distribution is related to the Brownian bridge after rescaling: $\sup_{0\leq z\leq 1} |B(z)|$.

In [12] a method is described for performing multiple change point analysis of a sequence of multi- variate observations. They are able to consistently detect any type of distributional change, and do not make any assumptions beyond the existence of the a-th absolute moment, for some $a \in (0, 2)$. In this setting however, the assumption must be made that the observations are independent *between* timesteps.

In a quite similar setting, [13] proposed a non-parametric method for offline detection of multiple change points in multivariate data; here as well different types of distributional changes are described, and independency between time steps is assumed.

Next we briefly mention other types of methods for anomaly detection encountered in the literature.

B. Generalised Likelihood Ratio

Generalised Likelihood Ratio (GLR) was developed in [14] but is also explained in [[2], Section 2.4.3]. A 'classic' CUSUM algorithm assumes that the distribution after a change point (just like the one before the change point) is fully specified, i.e., not only its form but also all the parameters are known. In practice this can often be found unrealistic, thus a change point detection method which can deal with unknown parameters after a change is desirable. The idea behind a Generalised Likelihood Ratio (GLR) method is to replace unknown parameters in the log-likelihood ratio by their maximum likelihood estimators. In [[2], Section 2.4.3] one can find further details of GLR method, along with the assumptions on the distributions class for which it can be applied.

A non-exhaustive list of recent developments on GLR will now follow.

- In [15] the reader finds recursive window-limited GLR algorithms for **State Space Models, ARX, Regression Models**. Alternatively see [[2], Chapter 7] for how to bring these models back to the multidimensional i.i.d. Gaussian i.i.d.
- There are also promising modifications for the GLR to account for the (possibly nonparametric) **multivariate** case (see [2]).
- There is no convenient recursive form for the basic GLR which can therefore be computationally complex. Therefore, [16] suggest to use limiting windows in the context of detecting additive changes in linear state-space models (see also [17], pp. 319).
- Also *window-limited GLR* involves many likelihood-ratio computations at each stage. Therefore, [18] establish a new suboptimal recursive approach which is based on a collection of L parallel recursive χ^2 tests instead of the window-limited GLR scheme. This new approach involves only a fixed number L of likelihood-ratio computations at each stage. According to the authors, by choosing an acceptable value of non-optimality, the designer can easily find a trade-off between the complexity of the quadratic change detection algorithm and its efficiency.
- Tuning problems (window size, threshold,...) have been addressed in [15] and [19]
- See [20] for generalization to the case that **pre- and post-change densities are unknown**.

This procedure can be extended easily to the multi-channel problem (multiple independent populations of independent sequences) as discussed in [21].

C. Exponentially Weighted Moving Average

In this method, abbreviated as EMWA, the following quantity is monitored (based on observations Y_t for $t = 1, \ldots, n$):

$$EWMA_t = \lambda Y_t + (1-\lambda) EWMA_{t-1}$$

where $EWMA_0$ is assumed to be a historical mean and $\lambda \in (0, 1]$ determines 'forgetting speed' (amount of the history taken into consideration). An alarm is issued when $EWMA_t$ exceeds a certain threshold. For details on the *multivariate* exponentially weighted moving average control charts (MEWMA) see [22] and [23]. This multidimensional version has the following properties:

- assumes that the process follows a (multi-dimensional) normal distribution with known mean and covariance;
- is sensitive to small persistent shifts in mean in any direction
- for the correlated data, if it is desired to detect small changes in process variability, so-called EWMA V-chart can be used, as proposed in [24];
- the method requires tuning to the size of the shift;
- pre-change distribution must be known.

Moreover, [25] proposed a generalization of the MEWMA to the case of multivariate time-dependent observations.

D. ChangeFinder

The *ChangeFinder* is a scheme for detecting outliers and change points in time series. For details see [26]. The ChangeFinder performs the following steps:

1) Read data x_t. As an example, [26] assume that x_t can be modeled as a **multidimensional AR process**.
2) Learn the probability density p_t ([26] propose an algorithm for doing so in the AR case).
3) Compute the *logarithmic loss score*

$$score(x_t) = -\log p_{t-1}(x_t|X_1^{t-1}),$$

where $X_1^{t-1} = x_1, \ldots, x_t$ (alternatively they propose the *quadratic loss score*

$$score(x_t) = (x_t - \hat{x}_t)^2,$$

where $\hat{x}_t$ is the prediction value for x_t given X_1^{t-1} based on model p_{t-1}). A higher score indicates that x_t is an outlier with higher probability.
4) Compute $y_t = \frac{1}{T}\sum_{i=t-T+1}^{t} score(x_i)$, the moving average of the scores.
5) Learn the density q_t of y_t (in the AR case by using the same algorithm as before).
6) Compute

$$score(t) = \frac{1}{T}\sum_{i=t-T+1}^{t} -\log q_{i-1}(y_i|Y_{i-k}^{i-1})$$

A higher score $score(t)$ indicates that time point t is a change point with higher probability.

The role of the two moving average processes is to reduce the influence of sole outliers. In the case where T is small, outliers and change points can be detected immediately after they appear. However, they may be difficult to discriminate from one another. In the case where T is large, it leads to time delay for detecting change points; however, outliers are filtered and only significant change points are detected accurately.

E. Subspace Identification

By using a pre-designed time-series model a subspace is discovered by principal component analysis from trajectories in past and present intervals, and their dissimilarity is measured by the distance between the subspaces. One of the major approaches called subspace identification [27], which compares the subspaces spanned by the columns of an 'extended observability matrix' (whose building blocks are the system matrices) generated by a **linear state-space model** with system noise. This paper discusses nonparametric algorithms for change-point detection in time-series data, based on the principle that the subspace spanned by the columns of an extended observability matrix is approximately equivalent to the one spanned by the sub-sequences of time-series data. Change-point detection is performed by estimating, on the basis of subspace identification, the column space of the extended observability matrix of the SSM behind time-series data, and the evaluating the subsequence of new-arrived data based on this subspace. The authors of [27] claim that their method can handle more abundant type of time-series data than conventional approaches because of implicitly utilizing generic SSMs, instead of AR models or constrained SSMs, as the model behind time series data.The method can explicitly handle interdependence over time. According to [27] the algorithm can be implemented easily also for on-line detection, they provide some pseudo-code. They also discuss the way their algorithm applies to input-output time-series data.

F. Direct Density Ratio Estimation

A common limitation of some of the above mentioned approaches is that they rely on pre-specified parametric models such as probability density models (CUSUM, GLR), auto-regressive models and state-space models. As mentioned earlier, to overcome this problem non-parametric method can be used. While non-parametric methods usually rely on non-parametric density estimation to calculate the log-likelihood ratio (LLR), in [28] the authors suggest to estimate the density ratio directly instead of estimating the individual densities and then they show how to use this approach in change-point detection. They use so- called Kullback-Leibler importance estimation procedure (KLIEP, see [29]). Later they extend this method and use methods denoted as unconstrained least-squares importance fitting (uLSIF) or relative uLSIF (RuLSIF) as building blocks (see [28]). The usefulness of the proposed method is demonstrated through experiments using artificial and real-world data sets. Furthermore, the dimensionality reduction approach for high-dimensional problems discussed in [30]. Also, [31] provide R and MATLAB implementations for density ratio estimation procedures for outlier detection.

G. Recipe for on-line changepoint detection

For on-line detection algorithms, the pre-change parameter is supposed to be known as it can be estimated accurately. Let θ be the parameter after change. Then, concerning this parameter we have to distinguish three cases:

1) pre-change parameter θ is known (or estimated from a training sample).
2) we have few information about θ, e.g. it is known that there exists a seperating hyperplane between the old parameter and θ.
3) nothing is known about θ.

Let k be the time that the change happens. Concerning k we distinguish (a)

1) k is a nonrandom unknown value or a random unknown with unknown distribution.
2) there is some a priori information about the distribution of k

IV. Implemented Algorithms

A. Likelihood Ratio Test

In Galeano and Peña's work on detecting covariances changes in multivariate data, they proposed two methods for calculating test statistics from which change points could be identified [6]. These methods model the given data as a vector autoregressive integrated moving average (vARIMA) popular in economics and financial market analysis, extracting the errors (or innovations) from this data, and applying these methods on this error data. The first such statistic, on which we focus here, uses a likelihood-ratio test (LRT) to compare two hypotheses: the null hypothesis H_n that the covariance of this error data is best characterized by a single covariance matrix Σ versus the alternative hypothesis H_a that, at some time point h, the data is best characterized by two separate covariances matrices Σ_1 before h and Σ_2 after h. The logarithm of a modified form of the ratio H_n/H_a then generates a test statistic LR_h that existing literature shows is governed by a chi-squared distribution with degrees of freedom proportional to the dimensionality k of the data. From simulations of this distribution, we can generate a critical value given some α against which to compare this test statistic to determine whether a change point actually exists at some time h.

1) Algorithm: Given some time-series data $\tilde{y}_t$ and confidence α, we use the following algorithm to identify points of change in covariance:

2) Implementation Details: To implement LRT, we used Python and Scikit's statsmodels package for fitting data to VAR() models. One should note this restriction to VAR() models is a result of an existing constraint in the statsmodels package.

We also implemented a version of the LRT algorithm that does not rely on calculating the W transformation matrix. Rather than evaluating W, we leveraged statsmodels and its maximum likelihood estimation to fit the data to two new VAR() models for each regime. The above algorithm performs better than this secondary implementation because it obviates the need for separate rounds of maximum likelihood estimation for each level of recursion.

B. CUSUM

The following changepoint detection algorithm, which focuses on changes in the covariance matrix, was designed by Galeano and Peña [6].

Function LRT($\tilde{y}_t, \alpha$) Algorithm by Galeano and Peña [6]

fit VARIMA(p, d', q) model to $\tilde{y}_t$;
compute residuals $\hat{e}_t$;

$k \leftarrow$ dimension$(\tilde{y}_t)$;
$d \leftarrow k(p+q+1) + \frac{k(k+1)}{2} + 1$; `/* minimum points needed */`
$n \leftarrow$ len$(\tilde{y}_t)$;
$df \leftarrow \frac{k(k+1)}{2}$; `/* degrees of freedom for` χ^2 `*/`

$C \leftarrow$ simulateChiSquareMax(df, α) ; `/* obtain the critical value */`

$LR \leftarrow$ zeros(n) ;
$S \leftarrow \frac{1}{n}\Sigma_{i=1}^{n} e_i \cdot e_i'$;
for $h \in [d, n-d-1]$ **do**
  $v \leftarrow h/n$;
  $S_1 \leftarrow \frac{1}{h}\Sigma_{i=1}^{h} e_i \cdot e_i'$;
  $S_2 \leftarrow \frac{1}{n-h}\Sigma_{i=h+1}^{n} e_i \cdot e_i'$;
  $LR[h] \leftarrow n \ln \frac{|S|}{|S_1|^v |S_2|^{1-v}}$;
end

$h_{max} \leftarrow argmax_h(LR)$;
$\Lambda_{max} \leftarrow LR[h_{max}]$;

changePoints $\leftarrow [\,]$;
if $\Lambda_{max} > C$ **then**
  changePoints += h_{max} ;
  $W \leftarrow$ transformation governing new data regime (see [6]);
  changePoints += apply LRT to $\hat{e}_t[0 : h_{max}]$;
  changePoints += apply LRT to $W \cdot \hat{e}_t[h_{max}+1 : n]$;
end

return changePoints

Suppose the k-dimensional time series $\boldsymbol{y}_t$ can be represented as a VARIMA process

$$\Phi(B)\boldsymbol{y}_t = \boldsymbol{c} + \Theta(B)\boldsymbol{a}_t,$$

where B is the backshift operator and the error or innovation $\boldsymbol{a}_t$ is an iid Gaussian random variable with mean $\mathbf{0}$ and covariance matrix Σ_i, where i indicates the particular regime.

The cusum test statistic is derived from accumulating the sum of squared errors. Since we do not know the true errors $\boldsymbol{a}_t$, we first fit a VARIMA process $\hat{\boldsymbol{y}}_t$ to the time series $\boldsymbol{y}_t$ and use the residuals $\boldsymbol{e}_t = \boldsymbol{y}_t - \hat{\boldsymbol{y}}_t$ as estimates for the errors. Consider the interval $\{\ell, \ldots, r\}$. We can estimate the covariance matrix of the time series in this

interval by

$$\hat{\Sigma}_\ell^r = \frac{1}{r-\ell}\sum_{t=\ell}^{r} \boldsymbol{e}_t \boldsymbol{e}_t^\top.$$

Denote the sum of squares accumulated up to time h by

$$A_\ell^r(h) = \sum_{t=\ell}^{h} \boldsymbol{e}_t^\top \left(\hat{\Sigma}_\ell^r\right)^{-1} \boldsymbol{e}_t,$$

where the squared errors are in some sense normalized by the empirical covariance $\hat{\Sigma}_\ell^r$ of the entire interval.

The cusum test statistic for a changepoint at time $h+1$ is then

$$C_\ell^r(h) = \frac{h}{\sqrt{2k(r-\ell+1)}}\left(\frac{A_\ell^r(h)}{h} - \frac{A_\ell^r(r-\ell+1)}{r-\ell+1}\right).$$

The term outside the parentheses is another normalization factor. The left-hand term inside the parentheses is the cumulative sum of squares up to time h. The right-hand term inside the parentheses is the cumulative sum of squares for the entire interval $\{\ell, \ldots, r\}$. The test statistic $C_\ell^r(h)$ compares these latter two terms. For example, if there is no changepoint in $\{\ell, \ldots, r\}$, then $\frac{A_\ell^r(h)}{h}$ and $\frac{A_\ell^r(r-\ell+1)}{r-\ell+1}$ will be approximately equal. However, if there is a single changepoint at time $t = h+1$, then the left-hand term is the sum of squares for all of the first regime (and only the first regime), while the right-hand term is the sum of squares over the entirety of both regimes. This suggests that the cumulative sums of squares are most different, and thus $C_\ell^r(h)$ is greatest in magnitude, at the changepoint $t = h+1$.

This observation is confirmed by plotting $|C_\ell^r(h)|$ vs. h for simulated data. Therefore, let

$$\Gamma_\ell^r = \max_{h\in\{\ell,\ldots,r\}} |C_\ell^r(h)|$$
$$\bar{h}_\ell^r = \operatorname*{argmax}_{h\in\{\ell,\ldots,r\}} |C_\ell^r(h)|.$$

Galeano and Peña prove that Γ_ℓ^r asymptotically has the distribution of the supremum over $[0,1]$ of a Brownian bridge. This is a known distribution, allowing us to calculate the critical value C_α for any given significance level α.

1) Algorithm for finding multiple changepoints: The approach to finding multiple changepoints is similar to binary segmentation. Let $d = k(p+q+1) + \frac{k(k+1)}{2} + 1$ be the resolution of the changepoint detection algorithm. First fit a VARIMA model to the data and computing the residuals. Then let $h_{\text{first}} = 1 + d$ and $h_{\text{last}} = T - d$, the final time step. Search for a changepoint in $\{h_{\text{first}}, \ldots, h_{\text{last}}\}$ by checking if $\Gamma_{h_{\text{first}}}^{h_{\text{last}}}$ exceeds the critical value. Let $\bar{h}_{\text{old}} = \bar{h}_{h_{\text{first}}}^{h_{\text{last}}}$.

If a candidate is found, split the entire time series at the time $t_2 = \bar{h}_{\text{old}}$ of the candidate changepoint. Check $\Gamma_{h_{\text{first}}}^{t_2}$ if greater than the critical value; if so, redefine t_2 as $\bar{h}_{h_{\text{first}}}^{t_2}$. Repeat until $\Gamma_{h_{\text{first}}}^{t_2}$ is no longer significant. This procedure thus finds the earliest point in the time series at which a changepoint could occur. Redefine h_{first} as the last significant value of $\bar{h}_{h_{\text{first}}}^{t_2}$. Perform the same procedure over the interval $\{\bar{h}_{\text{old}}, \ldots, h_{\text{last}}\}$, acquiring a new h_{last} as the latest point in the time series at which a changepoint could occur. If $|h_{\text{first}} - h_{\text{last}}| < d$, i.e., the resolution of the algorithm is not high enough to distinguish between h_{first} and h_{last}, then record $\bar{h}_{\text{old}}$ as a candidate changepoint. Otherwise, record both h_{first} and h_{last} as candidate changepoints. Then repeat this procedure in the narrower interval $\{h_{\text{first}}, \ldots, h_{\text{last}}\}$. Thus, rather than continually cutting the interval in two and repeating the procedure in each part, the algorithm instead narrows down the interval in which changepoints can occur, using previous changepoints as the new endpoints of the narrower interval.

The algorithm ends up find excess candidates. To solve this retrospectively, let $x_1 = 1$, $x_s = T$, and $\{x_2, \ldots, x_{s-1}\}$ be the sorted list of candidate changepoints. In each interval $\{x_i + 1, \ldots, x_{i+2} - 1\}$, drop x_i from the list of candidates if $\Gamma_{x_i+1}^{x_{i+2}-1}$ is insignificant. Repeat this procedure until convergence. Then remove x_1 and x_s from the list of candidate changepoints. Denote the winnowed list of candidates by X. Then the final changepoints detected by the algorithm are $\{x+1 : x \in X\}$.

The entire procedure is detailed more explicitly in the following algorithm:

C. Kernel Change Detection

Offline, parametric methods for change point detection often outperform their online, non-parametric competitors, but the flexibility gained by looser model restrictions and the ability to run in a streaming context are at times more valuable. The kernel-based change detection (KCD) algorithm proposed by Desbory et al. is one such online algorithm [32]. Rather than rely on a priori knowledge of the generating distribution for a given series of data, KCD instead leverages support vector machines (SVMs) to construct descriptions of the data in a higher dimensional space defined by some given kernel (for our cases, the radial basis function, or RBF, kernel). One can then use these high-dimensional descriptions to develop a dissimilarity statistic that characterizes the differences in center and spread of two possible regimes in the data. Additionally, SVM's popular kernel trick allows one to generate this dissimilarity statistic in input space rather than feature space.

1) Geometric Interpretation: Desobry et al. provide a compelling geometric interpretation of the KCD based on angles between vectors as a measure of dissimilarity and spread (see Figure 1 for a two-dimensional view). This geometric interpretation exists in feature

Algorithm 1: Cusum algorithm by Galeano and Peña. The four steps correspond to Galeano and Peña's enumeration.

Step 1 fit VARIMA model to $\boldsymbol{y}_t$;
compute residuals $\boldsymbol{e}_t$;
$candidates \leftarrow \{1, T\}$;
$h_{\text{first}} \leftarrow 1 + d$; $h_{\text{last}} \leftarrow T - d$;
while *True* **do**
Step 2 **if** $\Gamma_{h_{\text{first}}}^{h_{\text{last}}} < C_\alpha$ **then**
break;
else
$\Gamma_{\text{old}} \leftarrow \Gamma_{h_{\text{first}}}^{h_{\text{last}}}$; $\bar{h}_{\text{old}} \leftarrow \bar{h}_{h_{\text{first}}}^{h_{\text{last}}}$;
$\Gamma \leftarrow \Gamma_{\text{old}}$; $\bar{h} \leftarrow \bar{h}_{\text{old}}$;
Step 3a **while** $\Gamma > C_\alpha$ **do**
$t_2 \leftarrow \bar{h} - 1$;
$\Gamma = \Gamma_{h_{\text{first}}}^{t_2}$;
end
$h_{\text{first}} \leftarrow t_2$;
$\Gamma \leftarrow \Gamma_{\text{old}}$; $\bar{h} \leftarrow \bar{h}_{\text{old}}$;
Step 3b **while** $\Gamma > C_\alpha$ **do**
$t_1 \leftarrow \bar{h} + 1$;
$\Gamma = \Gamma_{t_1}^{h_{\text{last}}}$;
end
$h_{\text{last}} \leftarrow t_1$;
Step 3c **if** $|h_{\text{last}} - h_{\text{first}}| > d$ **then**
append $h_{\text{first}}, h_{\text{last}}$ to $candidates$;
$h_{\text{first}} = h_{\text{first}} + d$; $h_{\text{last}} = h_{\text{last}} - d$;
else
append $\bar{h}_{\text{old}}$ to $candidates$;
break;
end
end
end
sort $candidates$;
$\{x_1, \ldots, x_s\} \leftarrow candidates$;
Step 4 **repeat**
for $i \in \{1, \ldots, s-2\}$ **do**
if $\Gamma_{x_i+1}^{x_{i+2}-1} < C_\alpha$ **then**
remove x_{i+1} from $candidates$;
end
end
until *convergence*;
remove $1, T$ from $candidates$;
$changepoints \leftarrow \{x + 1 : x \in candidates\}$;

space and assumes that two one-class SVMs were used to describe data from a possible past regime and a possible future regime respectively. The angle between the vectors normal to the decision planes $\widehat{c_p c_f}$ can then be used as a dissimilarity metric between the two sets of data. To account for within-class spread, the authors then normalize this angle by the sum of the angles between the respective normal vectors a support vector of each regime $\widehat{c_p s_p}$ and $\widehat{c_f s_f}$. The result is the KCD dissimilarity statistic shown in Eq. 2.

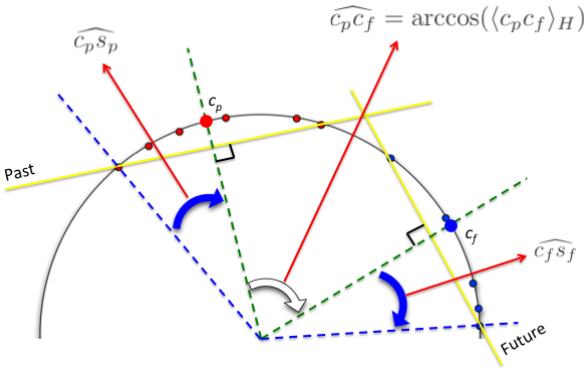

Fig. 1: KCD Geometry in Feature Space

$$KCD_{stat} = \frac{\widehat{c_p c_f}}{\widehat{c_f s_f} + \widehat{c_p s_p}} \tag{2}$$

These angles can be calculated using the inverse cosine of the dot product between the weight vectors, as shown in Eq. 3. One should note the denominator in Eq. 3 is for normalizing the vectors to unit length. This normalized dot product can be calculated in input space using SVM's learned weights $\boldsymbol{\alpha_p}$ and $\boldsymbol{\alpha_f}$ and the kernel matrices K as shown in Eq. 4. To obtain the within-class spread for each class, we use a similar form but rely on the SVM's intercept ρ rather than a dot product, as shown in Eq. 5.

$$\widehat{c_p c_f} = \arccos\left(\frac{\langle w_p, w_f \rangle_F}{\|w_p\| \|w_f\|}\right) \tag{3}$$

$$\frac{\langle w_p, w_f \rangle_F}{\|w_p\| \|w_f\|} = \frac{\boldsymbol{\alpha}_p^T K_{p,f} \boldsymbol{\alpha}_f}{\sqrt{\boldsymbol{\alpha}_p^T K_{p,p} \boldsymbol{\alpha}_p} \sqrt{\boldsymbol{\alpha}_f^T K_{f,f} \boldsymbol{\alpha}_f}} \tag{4}$$

$$\widehat{c_i s_i} = \arccos\left(\frac{\rho_i}{\sqrt{\boldsymbol{\alpha}_i^T K_{i,i} \boldsymbol{\alpha}_i}}\right), i \in \{p, f\} \tag{5}$$

2) Algorithm: KCD has four parameters:

- m – Window size, or the number of points on either side of a candidate change point.
- γ – SVM parameter governing bandwidth for the RBF kernel (other kernels are possible, but we did not experiment with them).
- ν – One-class SVM parameter governing proportion of points that should be counted as outliers when training.
- η – Threshold parameter such that, if $KCD_{stat} \geq \eta$ for some time point h, h is considered a change point.

We assume X is the input data of size $n \times k$.

Function KCD(X, m, γ, ν, η) Algorithm by Desobry et al.[32]

$n \leftarrow \text{len}(X)$;

changePoints $\leftarrow [\,]$;

for $h \in [m, n-m-1]$ **do**

$X_p \leftarrow X[h-m, h]$;

$X_f \leftarrow X[h, h+m]$;

$\text{svm}_p \leftarrow \text{SVM.fit}(X_p, \gamma, \nu)$;

$\text{svm}_f \leftarrow \text{SVM.fit}(X_f, \gamma, \nu)$;

$\boldsymbol{\alpha}_p \leftarrow \text{svm}_p.\alpha$;

$\boldsymbol{\alpha}_f \leftarrow \text{svm}_f.\alpha$;

$\boldsymbol{\rho}_p \leftarrow \text{svm}_p.\rho$;

$\boldsymbol{\rho}_f \leftarrow \text{svm}_f.\rho$;

$K_{p,p} \leftarrow \text{rbf}(X_p, X_p)$;

$K_{f,f} \leftarrow \text{rbf}(X_f, X_f)$;

$K_{p,f} \leftarrow \text{rbf}(X_p, X_f)$;

$\widehat{c_p c_f} \leftarrow \arccos\left(\frac{\boldsymbol{\alpha}_p^T K_{p,f} \boldsymbol{\alpha}_f}{\sqrt{\boldsymbol{\alpha}_p^T K_{p,p} \boldsymbol{\alpha}_p}\sqrt{\boldsymbol{\alpha}_f^T K_{f,f} \boldsymbol{\alpha}_f}}\right)$;

$\widehat{c_p s_p} \leftarrow \arccos\left(\frac{|\boldsymbol{\rho}_p|}{\sqrt{\boldsymbol{\alpha}_p^T K_{p,p} \boldsymbol{\alpha}_p}}\right)$;

$\widehat{c_f s_f} \leftarrow \arccos\left(\frac{|\boldsymbol{\rho}_f|}{\sqrt{\boldsymbol{\alpha}_f^T K_{f,f} \boldsymbol{\alpha}_f}}\right)$;

$KCD_{stat} \leftarrow \frac{c_p c_f}{c_p s_p + c_f s_f}$;

if $KCD_{stat} > \eta$ **then**

changePoints += h

end

end

return changePoints

V. Performance Comparison

For covariance changes, we generated two regimes of data with constant mean and different covariance matrices. KCD then fit one-class SVMs to the covariance matrices within the past and future windows. Mean shifts used random means and constant covariance. We simulated 500 bi-variate data points with a change point at h=250, KCD window size of 400 (m=200), and compared the LRT and CUSUM test statistics at the 95% confidence level. Results for this comparison of change point types is shown in Figure 2.

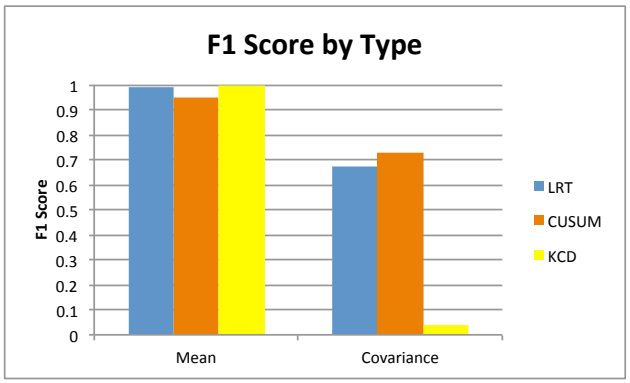

Fig. 2: F1 Scores by Type

To explore sensitivity to dimensionality, we simulated 500 multi-variate data points but included change points at $h = \{125, 250, 375\}$. We left the KCD window size at 400 (m=200), and compared the LRT and CUSUM test statistics at the 95% confidence level. We then varied dimensionality from $k = [2, 10]$. Results for both types of change points are shown in Figures 3 and 4.

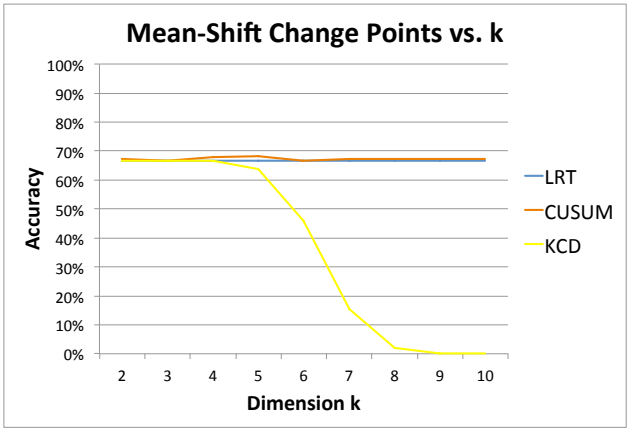

Fig. 3: Mean-Shift Accuracy versus Dimension k

It seems the LRT and CUSUM-based algorithms are realatively insensitive to increases in dimensionality. KCD, on the other hand, seems quite sensitive with its accuracy falling to near 0% by $k = 9$.

To explore how well these algorithms adapted to multiple change points, we simulated 3,000 bi-variate data points with 2 to 12 change points distributed evenly throughout the data set. We left the KCD window size at 400 (m=200), and compared the LRT and CUSUM test

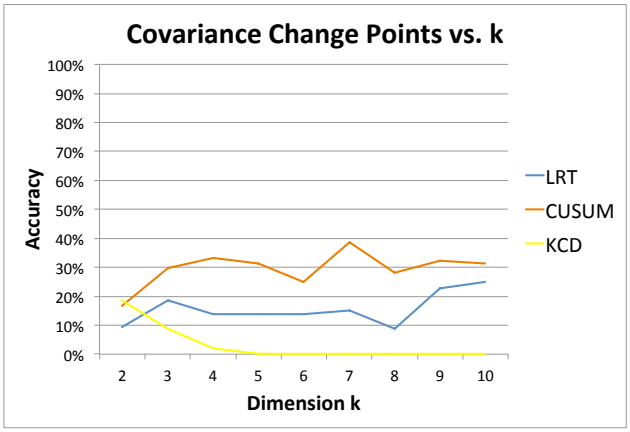

Fig. 4: Covariance Accuracy versus Dimension k

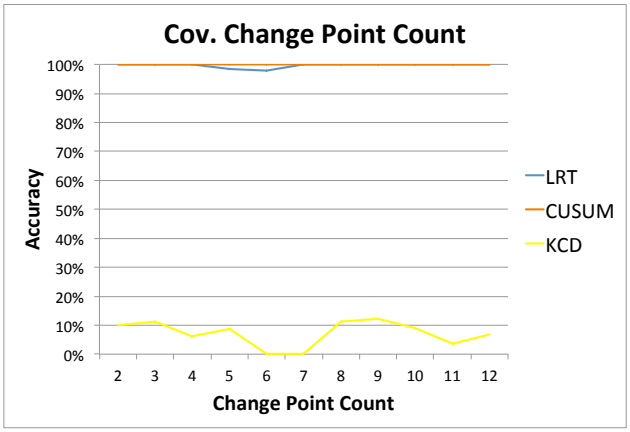

Fig. 6: Covariance Accuracy versus Change Points

statistics at the 95% confidence level. Results are shown in Figures 5 and 6.

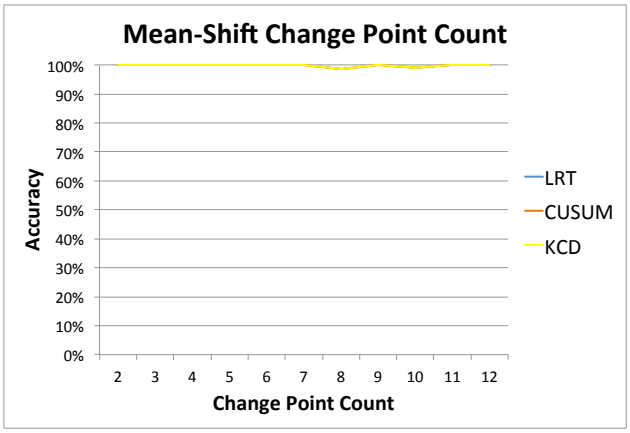

Fig. 5: Mean-Shift Accuracy versus Change Points

VI. Conclusions

Our performance data suggests several interesting results. First, the parametric LRT and CUSUM algorithms outperform the non-parametric KCD algorithm when detecting changes in covariance. Second, KCD is competitive in detecting shifts in mean even with relatively small window sizes. Also, LRT and CUSUM are more robust to increases in dimensionality of the data.

Acknowledgment

This work made use of the Open Science Data Cloud (OSDC) which is an Open Cloud Consortium (OCC)-sponsored project. The OSDC is supported in part by grants from Gordon and Betty Moore Foundation and the National Science Foundation and major contributions from OCC members like the University of Chicago. Both Cody Buntain and Christopher Natoli were supported by the National Science Foundation Partnerships for Research and Education (PIRE) Award Number 1129076. Any opinions, findings, and conclusions or recommendations expressed are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

This work was made possible using the resources of the Open Science Data Cloud [33].

References

[1] P. J. Brockwell and R. A. Davis, *Time series: theory and methods*. Springer, 2009.

[2] M. Basseville, I. V. Nikiforov *et al.*, *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs, 1993, vol. 104.

[3] D. Han, F. Tsung, X. Hu, and K. Wang, "Cusum and ewma multi-charts for detecting a range of mean shifts," *Statistica Sinica*, vol. 17, no. 3, p. 1139, 2007.

[4] J. D. Healy, "A note on multivariate cusum procedures," *Technometrics*, vol. 29, no. 4, pp. 409–412, 1987.

[5] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014. [Online]. Available: http://www.R-project.org

[6] P. Galeano and D. Peña, "Covariance changes detection in multivariate time series," *Journal of Statistical Planning and Inference*, vol. 137, no. 1, pp. 194–211, Jan. 2007. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S0378375805002673

[7] A. G. Tartakovsky, B. Rozovskii, and K. Shah, "A nonparametric multichart cusum test for rapid intrusion detection," in *Proceedings of Joint Statistical Meetings, Minneapolis, MN*. Citeseer, 2005.

[8] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blažek, and H. Kim, "Detection of intrusions in information systems by sequential change-point methods," *Statistical Methodology*, vol. 3, no. 3, pp. 252–293, 2006.

[9] A. Aue, S. Hörmann, L. Horváth, M. Reimherr *et al.*, "Break detection in the covariance structure of multivariate time series models," *The Annals of Statistics*, vol. 37, no. 6B, pp. 4046–4087, 2009.

[10] D. Wied, "A nonparametric test for a constant correlation matrix," *arXiv preprint arXiv:1210.1412*, 2012.

[11] M. Arnold, N. Bissantz, D. Wied, and D. Ziggel, *A new online-test for changes in correlations between assets*. SFB 823, 2010.

[12] D. S. Matteson and N. A. James, "A nonparametric approach for multiple change point analysis of multivariate data," *Journal of the American Statistical Association*, vol. 109, no. 505, pp. 334–345, 2014.

[13] A. Lung-Yut-Fong, C. Lévy-Leduc, and O. Cappé, "Homogeneity and change-point detection tests for multivariate data using rank statistics," *arXiv preprint arXiv:1107.1971*, 2011.

[14] D. Siegmund and E. Venkatraman, "Using the generalized likelihood ratio statistic for sequential detection of a change-point," *The Annals of Statistics*, pp. 255–271, 1995.

[15] T. L. Lai, "Sequential changepoint detection in quality control and dynamical systems," *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 613–658, 1995.

[16] A. S. Willsky and H. L. Jones, "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *Automatic Control, IEEE Transactions on*, vol. 21, no. 1, pp. 108–112, 1976.

[17] T. L. Lai, "Sequential analysis: some classical problems and new challenges," *Statistica Sinica*, vol. 11, no. 2, pp. 303–350, 2001.

[18] I. V. Nikiforov, "A suboptimal quadratic change detection scheme," *Information Theory, IEEE Transactions on*, vol. 46, no. 6, pp. 2095–2107, 2000.

[19] T. L. Lai and J. Z. Shan, "Efficient recursive algorithms for detection of abrupt changes in signals and control systems," *Automatic Control, IEEE Transactions on*, vol. 44, no. 5, pp. 952–966, 1999.

[20] T. L. Lai and H. Xing, "Sequential change-point detection when the pre-and post-change parameters are unknown," *Sequential Analysis*, vol. 29, no. 2, pp. 162–175, 2010.

[21] A. G. Tartakovsky, "Multidecision quickest change-point detection: Previous achievements and open problems," *Sequential Analysis*, vol. 27, no. 2, pp. 201–231, 2008.

[22] C. A. Lowry, W. H. Woodall, C. W. Champ, and S. E. Rigdon, "A multivariate exponentially weighted moving average control chart," *Technometrics*, vol. 34, no. 1, pp. 46–53, 1992.

[23] S. S. Prabhu and G. C. Runger, "Designing a multivariate ewma control chart," *Journal of Quality Technology*, vol. 29, no. 1, 1997.

[24] A. Yeh, D. Lin, H. Zhou, and C. Venkataramani, "A multivariate exponentially weighted moving average control chart for monitoring process variability," *Journal of Applied Statistics*, vol. 30, no. 5, pp. 507–536, 2003.

[25] H. G. Kramer and L. Schmid, "Ewma charts for multivariate time series," *Sequential Analysis*, vol. 16, no. 2, pp. 131–154, 1997.

[26] J.-i. Takeuchi and K. Yamanishi, "A unifying framework for detecting outliers and change points from time series," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 18, no. 4, pp. 482–492, 2006.

[27] Y. Kawahara, T. Yairi, and K. Machida, "Change-point detection in time-series data based on subspace identification," in *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*. IEEE, 2007, pp. 559–564.

[28] S. Liu, M. Yamada, N. Collier, and M. Sugiyama, "Change-point detection in time-series data by relative density-ratio estimation," *Neural Networks*, vol. 43, pp. 72–83, 2013.

[29] Y. Kawahara and M. Sugiyama, "Change-point detection in time-series data by direct density-ratio estimation." in *SDM*, vol. 9. SIAM, 2009, pp. 389–400.

[30] M. Yamada and M. Sugiyama, "Direct density-ratio estimation with dimensionality reduction via hetero-distributional subspace analysis." in *AAAI*, 2011.

[31] S. Hido, Y. Tsuboi, H. Kashima, M. Sugiyama, and T. Kanamori, "Statistical outlier detection using direct density ratio estimation," *Knowledge and information systems*, vol. 26, no. 2, pp. 309–336, 2011.

[32] F. Desobry, M. Davy, and C. Doncarli, "An online kernel change detection algorithm," *Signal Processing, IEEE Transactions on*, vol. 53, no. 8, pp. 2961–2974, Aug. 2005.

[33] R. L. Grossman, M. Greenway, A. P. Heath, R. Powell, R. D. Suarez, W. Wells, K. P. White, M. P. Atkinson, I. A. Klampanos, H. L. Alvarez, C. Harvey, and J. Mambretti, "The design of a community science cloud: The open science data cloud perspective," in *SC Companion*, 2012, pp. 1051–1057.