



A constant-per-iteration likelihood ratio test for online changepoint detection for exponential family models

Kes Ward¹ · Gaetano Romano¹ · Idris Eckley¹ · Paul Fearnhead¹

Received: 9 February 2023 / Accepted: 21 February 2024
© Crown 2024

Abstract

Online changepoint detection algorithms that are based on (generalised) likelihood-ratio tests have been shown to have excellent statistical properties. However, a simple online implementation is computationally infeasible as, at time T , it involves considering $O(T)$ possible locations for the change. Recently, the FOCuS algorithm has been introduced for detecting changes in mean in Gaussian data that decreases the per-iteration cost to $O(\log T)$. This is possible by using pruning ideas, which reduce the set of changepoint locations that need to be considered at time T to approximately $\log T$. We show that if one wishes to perform the likelihood ratio test for a different one-parameter exponential family model, then exactly the same pruning rule can be used, and again one need only consider approximately $\log T$ locations at iteration T . Furthermore, we show how we can adaptively perform the maximisation step of the algorithm so that we need only maximise the test statistic over a small subset of these possible locations. Empirical results show that the resulting online algorithm, which can detect changes under a wide range of models, has a constant-per-iteration cost on average.

Keywords Online changepoint · Real-time analysis · Ex-FOCuS · Time series

1 Introduction

Detecting changes in data streams is an important statistical and machine learning challenge that arises in applications as diverse as climate records (Beaulieu and Killick, 2018), financial time-series (Andreou and Ghysels, 2002), monitoring performance of virtual machines (Barrett et al, 2017) and detecting concept drift of inputs to classifiers (Sakamoto et al, 2015). In many contemporary applications there is a need to detect changes online. In such settings we sequentially monitor a data stream over time, seeking to flag that a

change has occurred as soon as possible. Often online change algorithms need to run under limited computational resource. For example, Ward et al (2023) detect gamma ray bursts using the local computing resource onboard small cube satellites, and Varghese et al (2016) work with sensor networks where computations need to be performed locally by the sensors. Alternatively algorithms may need to be run for ultra high-frequency data (Iwata et al, 2018), or need to be run concurrently across a large number of separate data streams. These settings share a common theme of tight constraints on the computational complexity of viable algorithms.

There have been a number of procedures that have been suggested for online detection of changes. For detecting simple changes in univariate data, these include Bayesian approaches (Adams and MacKay, 2007; Lai and Xing, 2010), and approaches based on recursively applying likelihood ratio, or similar tests (e.g. Lai 1995; Kirch et al 2018; Aue and Kirch 2024, and references therein). In high-dimensional settings often the type of test to use, and how to combine evidence for a change across difference components of the data, is important (see e.g. Mei 2010; Keriven et al 2020; Keshavarz et al 2020; Chen et al 2022).

We focus on detecting changes in univariate data. Even in this setting, computational constraints can impact on the

Kes Ward, Gaetano Romano, Idris Eckley and Paul Fearnhead have contributed equally to this work.

✉ Gaetano Romano
g.romano@lancaster.ac.uk

Kes Ward
k.ward4@lancaster.ac.uk

Idris Eckley
i.eckley@lancaster.ac.uk

Paul Fearnhead
p.fearnhead@lancaster.ac.uk

¹ Department of Mathematics and Statistics, Lancaster University, Lancaster LA1 4YF, England, UK

approach used. For example the Bayesian approach of Adams and MacKay (2007) has a computational cost of processing each iteration that increases linearly with time, which has led to approximate algorithms that have a constant cost per observation (Fearnhead and Liu, 2007).

Similar trade-offs between statistical efficiency and computational cost can be seen for the frequentist approaches that we focus on. For example, Yu et al (2023) proposed a likelihood-ratio test with excellent statistical properties, but the natural implementation of this method has a computational cost per iteration that increases linearly with time. However, for online applications we need the computational cost to be constant. There exist algorithms with a constant computational cost per iteration, but they need one to only test for changes that are a pre-specified time in the past (e.g. Eichinger and Kirch 2018; Ross et al 2011; Ross and Adams 2012; Chen and Tian 2010), or specify the distribution of the data after a change (e.g. Page 1954; Lucas 1985). If the choices made in implementing these algorithms are inappropriate for the actual change one wishes to detect, this can lead to a substantial loss of power.

Recently Romano et al (2023b) proposed a new algorithm called Functional Online Cumulative Sum (FOCuS). This algorithm is able to perform the (generalised) likelihood-ratio test with a computational cost that only increases logarithmically with time. FOCuS was developed for detecting a change in mean in Gaussian data and has been extended to Poisson (Ward et al, 2023) and Binomial (Romano et al, 2024) data. FOCuS has two components: one that does pruning of past changepoint times that need not be considered in the future, and a maximisation step that considers all past changepoint times that have not been pruned. Interestingly, the pruning step for Poisson and Binomial data is identical to that for Gaussian data, and it is only the maximisation step that changes.

In this paper we show that this correspondence extends to other one-parameter exponential family models. Furthermore, we show how to substantially speed up FOCuS. In previous implementations the pruning step has a fixed average cost per iteration, and the computational bottleneck is the maximisation step that, at time T , needs to consider on average $O(\log T)$ possible changepoint locations. We show how previous calculations can be stored so that the maximisation step can consider fewer past changepoint locations. Empirically this leads to a maximisation step whose per iteration computational cost is $O(1)$. To our knowledge this is the first algorithm that exactly performs the likelihood-ratio test for detecting a change with an average constant-per-iteration cost.

To summarise, the contribution of our paper is purely computational. We present an algorithm that exactly performs a likelihood-ratio test for a change as each new data-point is observed, but with a cost that is much lower than the

$O(T)$ cost of naively considering all possible change-point locations prior to time T . We do not discuss the theoretical properties of the likelihood-ratio test, as this has been studied extensively, see for example Yu et al (2023), Lorden (1971) and Lai and Xing (2010). When implementing the test we need to choose a threshold for the test statistic that determines whether we flag that there has been a change. This can be chosen to control the rate of falsely detecting a change. There are two general approaches to do so, one is to have a fixed threshold and choose this to control the average run length, the average time until we incorrectly detect a change if there is none (see e.g. Li et al 2014, and references therein). Alternatively we can have a threshold that increases with T so that we control the overall probability of detecting a change for an infinite data stream with no change (see Aue and Kirch 2024, and references therein). In our simulations we control the average run length, with an appropriate threshold being chosen by simulation. However the FOCuS algorithm we develop can be applied under either setting.

2 Background

2.1 Problem statement

Assume we observe a univariate time series signal x_1, x_2, \dots , and wish to analyse the data online and detect any change in the distribution of the data as quickly as possible. We will let T denote the current time point.

A natural approach to this problem is to model the data as being independent realisations from some parametric family with density $f(x | \theta)$. Let θ_0 be the parameter of the density before any change. If there is a change, denote the time of the change as τ and the parameter after the change as θ_1 . We can then test for a change using a likelihood-ratio test statistic. Such test statistics have often been termed generalised likelihood-ratio test statistics in the literature (e.g. Lai and Xing 2010).

There are two scenarios for such a test. First, we can assume the pre-change distribution, and hence θ_0 is known (Eichinger and Kirch, 2018). This simplifying assumption is commonly made when we have substantial training data from the pre-change distribution with which to estimate θ_0 . Alternatively, we can let θ_0 be unknown. We will initially focus on the pre-change distribution known case, and explain how to extend ideas to the pre-change distribution unknown case in Sect. 4.

The log-likelihood for the data $x_{1:T} = (x_1, \dots, x_T)$, which depends on the pre-change parameter, θ_0 , the post-change parameter, θ_1 , and the location of a change, τ , is

$$\ell(x_{1:T}|\theta_0, \theta_1, \tau) := \sum_{t=1}^{\tau} \log f(x_t|\theta_0) + \sum_{t=\tau+1}^T \log f(x_t|\theta_1).$$

The log-likelihood ratio test statistic for a change prior to T is thus

$$LR_T := 2 \left\{ \max_{\theta_1, \tau} \ell(x_{1:T}|\theta_0, \theta_1, \tau) - \ell(x_{1:T}|\theta_0, \cdot, T) \right\}.$$

Naively calculating the log-likelihood ratio statistic involves maximising over a set of T terms at time T . This makes it computationally prohibitive to calculate in an online setting when T is large. There are two simple pre-existing approaches to overcome this, and make the computational cost per iteration constant. First, MOSUM approaches (e.g. Chu et al 1995; Eichinger and Kirch 2018) fix a number, K say, of changepoint times to be tested, with these being of the form $\tau = T - h_i$ for a suitable choice of h_1, \dots, h_K . Alternatively one can use Page's recursion (Page, 1954, 1955) that calculates the likelihood-ratio test statistic for a pre-specified post-change parameter. Again we can use a grid of K possible post-change parameters. Both these approaches lose statistical power if the choice of either changepoint location (i.e. the h_i values for MOSUM) or the post-change parameter are inappropriate for the actual change in the data we are analysing.

2.2 FOCuS for Gaussian data

As an alternative to MOSUM or Page's recursion, Romano et al (2023b) introduce the FOCuS algorithm that can efficiently calculate the log-likelihood ratio statistic for univariate Gaussian data where θ denotes the data mean.

In this setting, it is simple to see that

$$\ell(x_{1:T}|\theta_0, \theta_1, \tau) - \ell(x_{1:T}|\theta_0, \cdot, T) = \sum_{t=\tau+1}^T \{\log f(x_t|\theta_1) - \log f(x_t|\theta_0)\}.$$

We can then introduce a function

$$Q_T(\theta_1) = \max_{\tau} \left\{ \sum_{t=\tau+1}^T \left(\log f(x_t|\theta_1) - \log f(x_t|\theta_0) \right) \right\},$$

which is the log-likelihood ratio statistic if the post-change parameter, θ_1 , is known. Obviously, $LR_T = \max_{\theta_1} 2Q_T(\theta_1)$.

For Gaussian data with known mean, θ_0 , and variance, σ^2 , we can standardise the data so that the pre-change mean is 0 and the variance is 1. In this case, each term in the sum of the log-likelihood ratio statistic simplifies to $\theta_1(x_t - \theta_1/2)$, and

$$Q_T(\theta_1) = \max_{\tau} \left\{ \sum_{t=\tau+1}^T \theta_1(x_t - \theta_1/2) \right\}. \quad (1)$$

This is the point-wise maximum of $T - 1$ quadratics. We can thus store $Q_T(\theta_1)$ by storing the coefficients of the quadratics.

The idea of FOCuS is to recursively calculate $Q_T(\theta_1)$. Whilst we have written $Q_T(\theta_1)$ as the maximum of $T - 1$ quadratics in θ_1 , each corresponding to a different location of the putative change, in practice there are only $\approx \log T$ quadratics that contribute to Q_T (Romano et al, 2023b). This means that, if we can identify this set of quadratics, we can maximise Q_T , and hence calculate the test statistic, in $O(\log T)$ operations. Furthermore Romano et al (2023b) show that we can recursively calculate Q_T , and the minimal set of quadratics we need, with a cost that is $O(1)$ per iteration on average.

The FOCuS recursion is easiest described for the case where we want a positive change, i.e. $\theta_1 > \theta_0$. An identical recursion can then be applied for $\theta_1 < \theta_0$ and the results combined to get Q_T . This approach to calculating Q_T uses the recursion of Page (1954),

$$Q_T(\theta_1) = \max \{Q_{T-1}(\theta_1), 0\} + \theta_1(x_T - \theta_1/2).$$

To explain how to efficiently solve this recursion, it is helpful to introduce some notation. For $\tau_i < \tau_j$ define

$$C_{\tau_i}^{(\tau_j)}(\theta_1) = \sum_{t=\tau_i+1}^{\tau_j} \theta_1(x_t - \theta_1/2). \quad (2)$$

At time $T - 1$ let the quadratics that contribute to Q_{T-1} , for $\theta_1 > \theta_0$, correspond to changes at times $\tau \in \mathcal{I}_{T-1}$. Then

$$Q_{T-1}(\theta_1) = \max_{\tau \in \mathcal{I}_{T-1}} \left\{ C_{\tau}^{(T-1)}(\theta_1) \right\}.$$

Substituting into Page's recursion we obtain

$$Q_T(\theta_1) = \max \left\{ \max_{\tau \in \mathcal{I}_{T-1}} \left\{ C_{\tau}^{(T)}(\theta_1) \right\}, C_{T-1}^T(\theta_1) \right\},$$

from which we have that $\mathcal{I}_T \subseteq \mathcal{I}_{T-1} \cup \{T - 1\}$.

The key step now is deciding which changepoint locations in $\mathcal{I}_{T-1} \cup \{T - 1\}$ no longer contribute to Q_T . To be consistent with ideas we present in Sect. 3 we will present the

FOCuS algorithm in a slightly different way to Romano et al (2023b). Assume that $\mathcal{I}_{T-1} = \{\tau_1, \dots, \tau_n\}$, with the candidate locations ordered so that $\tau_1 < \tau_2 < \dots < \tau_n$. We can now define the difference between successive quadratics as

$$\begin{aligned} C_{\tau_i}^{(T)}(\theta_1) - C_{\tau_{i+1}}^{(T)}(\theta_1) \\ &= C_{\tau_i}^{(T-1)}(\theta_1) - C_{\tau_{i+1}}^{(T-1)}(\theta_1) \\ &= C_{\tau_i}^{(\tau_{i+1})}(\theta_1). \end{aligned}$$

These differences do not change from time $T-1$ to time T .

For the difference between quadratics associated with changes at τ_i and τ_{i+1} , let $l_i \geq 0$ denote the largest value of θ_1 such that $C_{\tau_i}^{(\tau_{i+1})}(\theta_1) \geq 0$. By definition $C_{\tau_i}^{(\tau_{i+1})}(\theta_0) = 0$. Hence it is readily shown that

$$C_{\tau_i}^{(T)}(\theta_1) \geq C_{\tau_{i+1}}^{(T)}(\theta_1),$$

on $\theta \in [\theta_0, l_i]$. For $\theta_1 \geq l_i$ compare $C_{\tau_{i+1}}^{(T)}(\theta_1)$ with $C_{T-1}^{(T)}(\theta_1)$. If $C_{\tau_{i+1}}^{(T)}(\theta_1) \leq C_{T-1}^{(T)}(\theta_1)$ then

$$\begin{aligned} C_{\tau_{i+1}}^{(T)}(\theta_1) - C_{T-1}^{(T)}(\theta_1) &\leq 0 \\ \Leftrightarrow C_{\tau_{i+1}}^{(T-1)}(\theta_1) &\leq 0. \end{aligned}$$

A sufficient condition for $C_{\tau_{i+1}}^{(T-1)}(\theta_1) \leq 0$ for all $\theta_1 > l_i$ is for the largest root of $C_{\tau_{i+1}}^{(T-1)}(\theta_1)$ to be smaller than l_i . In this case we have that $C_{\tau_{i+1}}^{(T)}(\theta_1)$ does not contribute to $Q_T(\cdot)$ and thus can be pruned.

This suggests Algorithm 1. Note that this algorithm is presented differently from that in Romano et al (2023b), as the way the quadratics are stored is different. Specifically, here we store the difference in the quadratics, rather than use summary statistics. The input is just the difference of the quadratics that contribute to Q_{T-1} . The main loop of the algorithm just checks whether the root of $C_{\tau_j}^{(T-1)}$ is smaller than that of $C_{\tau_{j-1}}^{(\tau_j)}$, which is our condition for pruning the quadratic associated with τ_j . If not, we stop any further pruning and return the set of quadratic differences plus the quadratic $C_{T-1}^{(T)}$. If it is, then the quadratic associated with τ_j is removed and the quadratic difference associated with τ_{j-1} is updated – by adding on the quadratic difference associated with τ_j . We then loop to consider removing the next quadratic (if there is one).

A pictorial description of the algorithm is shown in Fig. 1. It is simple to see that this algorithm has an average cost per iteration that is $O(1)$. This is because, at each iteration, the number of steps of the while loop is one more than the number of quadratics that are pruned. As only one quadratic is added at each iteration, and a quadratic can only be removed once, the overall number of steps of the while loop by time T will be less than $2T$.

Algorithm 1 FOCuS update at time T for $\theta_1 > \theta_0$ and $\theta_0 = 0$. Algorithm based on storing quadratic differences.

Input: A set of n quadratic differences, $C_{\tau_i}^{(\tau_{i+1})}(\theta_1)$, for $i = 1, \dots, n$, with $\tau_i < \tau_{i+1}$ and $\tau_{n+1} = T-1$ such that

$$Q_{T-1}(\theta_1) = \max_i \{C_{\tau_i}^{(\tau_{i+1})}\}.$$

The set of largest roots, l_i , such that $C_{\tau_i}^{(\tau_{i+1})}(l_i) = 0$, for $i = 1, \dots, n$.

```

1:  $j \leftarrow n$ 
2:  $l_0 \leftarrow \theta_0$ 
3: while  $j > 0$  do
4:   if  $l_j \leq l_{j-1}$  then
5:      $C_{\tau_{j-1}}^{(T-1)}(\theta_1) \leftarrow C_{\tau_{j-1}}^{\tau_j}(\theta_1) + C_{\tau_j}^{(T-1)}(\theta_1)$ 
6:     Recalculate  $l_{j-1}$ , largest root of  $C_{\tau_{j-1}}^{(T-1)}(\theta_1) = 0$ 
7:      $\tau_j \leftarrow T-1$ 
8:      $j \leftarrow j-1$ 
9:   end if
10:  Break
11: end while
12:  $C_{T-1}^{(T)}(\theta_1) \leftarrow \theta_1(x_T - \theta_1/2)$ 
13:  $\tau_{j+1} \leftarrow T-1$  and  $\tau_{j+2} \leftarrow T$ 
14:  $l_{j+1} \leftarrow 2x_T$ 
15:  $n \leftarrow j+1$ 

```

Output: The set of n quadratic differences, $C_{\tau_i}^{(\tau_{i+1})}(\theta_1)$ and roots l_i for $i = 1, \dots, n$.

3 FOCuS for exponential family models

Different parametric families will have different likelihoods, and likelihood ratio statistics. However the idea behind FOCuS can still be applied in these cases provided we are considering a change in a univariate parameter, with different forms for the curves (described in Eq. 2) and hence different values for the roots of the curves. Whilst one would guess that the different values of the roots would lead to different pruning of curves when implementing Algorithm 1, Ward et al (2023) and Romano et al (2024) noted that the pruning, i.e. the changepoints associated with the functions that contribute to Q_T , are the same for a Poisson model or a Binomial model as for the Gaussian model; it is only the shape of the functions that changes. Here we show that this is a general property for many one-parameter exponential family models.

A one-parameter exponential family distribution can be written as

$$f(x | \theta) = \exp[\alpha(\theta) \cdot \gamma(x) - \beta(\theta) + \delta(x)],$$

for some one-parameter functions $\alpha(\theta)$, $\beta(\theta)$, $\gamma(x)$, $\delta(x)$ which are dependent on the specific distribution. Examples of one-parameter exponential family distributions are given in Table 1 and include Gaussian change in mean, Gaussian change in variance, Poisson, Gamma change in scale, and Binomial distributions, for which $\alpha(\theta)$ and $\beta(\theta)$ are increasing functions. $\gamma(x)$ is the sufficient statistic for the model,

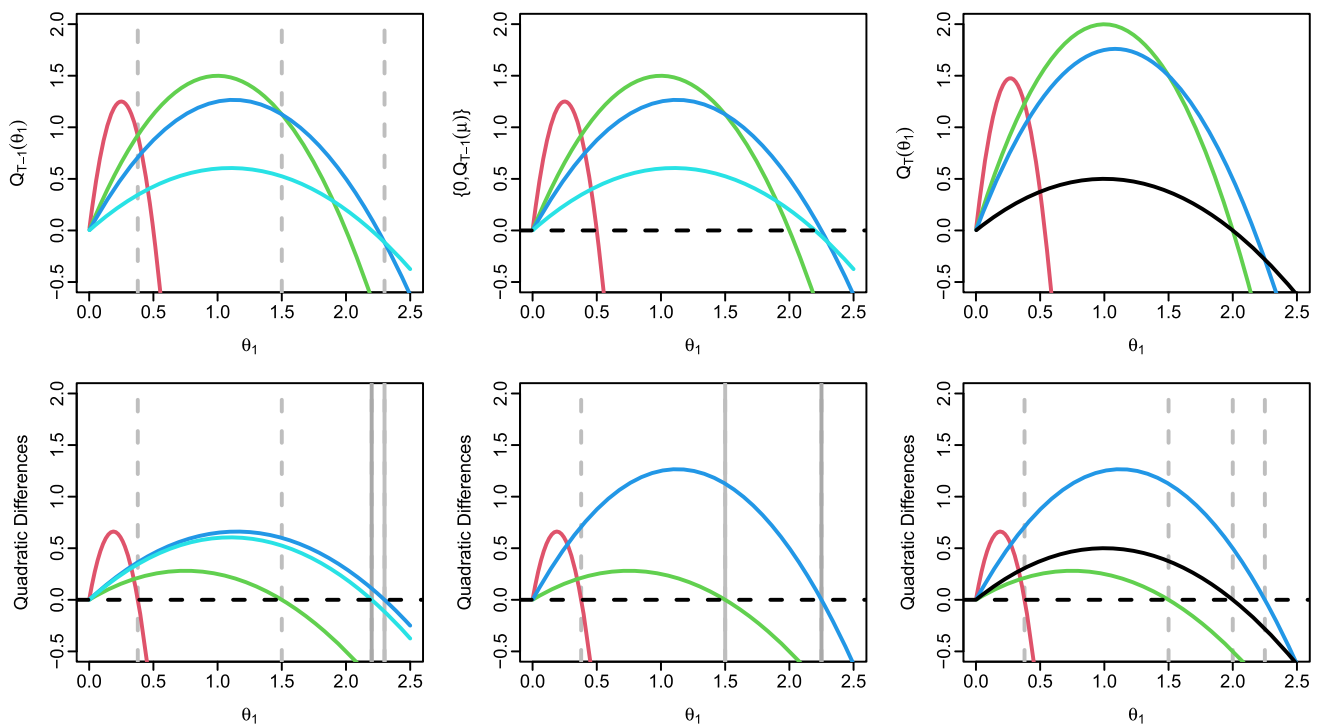


Fig. 1 Example of one iteration of FOCuS. The top row plots the quadratics $C_{\tau_1}^{(T-1)}$ (red), $C_{\tau_2}^{(T-1)}$ (green), $C_{\tau_3}^{(T-1)}$ (blue), $C_{\tau_4}^{(T-1)}$ (cyan) that contribute to $Q_T(\theta_1)$ directly, together with the intervals where each is optimal (demarked by grey vertical lines). To prune, we first add the zero line (dotted black), then prune $C_{\tau_4}^{(T-1)}$, as it is no longer optimal for any θ_1 . We then add $\theta_1(x_T - \theta_1/2)$ to all quadratics. The bottom-left plot shows the storage of quadratic differences $C_{\tau_1}^{(T-1)}$ (red), $C_{\tau_2}^{(T-1)}$ (green), $C_{\tau_3}^{(T-1)}$ (blue), $C_{\tau_4}^{(T-1)}$ (cyan) in Algorithm 1. The roots of these quadratic differences are shown by grey vertical lines. The roots

of the first three quadratic difference demark the intervals where the quadratics are optimal. The root of $C_{\tau_4}^{(T-1)}$ shows the region where that curve is above the zero-line. The algorithm considers pruning τ_4 based on whether the root of $C_{\tau_4}^{(T-1)}$ is smaller than the root of $C_{\tau_3}^{(T-1)}$. The pruning of τ_4 combines cyan with blue into the quadratic difference $C_{\tau_3}^{(T-1)}$ (bottom-middle, blue line). We then add $C_{\tau_1}^{(T-1)}$ (black) as its own quadratic difference (bottom-right). We require no iteration over the full quadratic list, as $C_{\tau_1}^{(T-1)}$ and $C_{\tau_2}^{(T-1)}$ remain untouched

Table 1 Examples of one-parameter exponential families and the corresponding forms of $\alpha(\theta)$, $\beta(\theta)$ and $\gamma(x)$

Distribution	$\alpha(\theta)$	$\beta(\theta)$	$\gamma(x)$
Gaussian (change in mean)	θ	θ^2	x
Gaussian (change in variance)	$-1/\theta^2$	$\log(\theta)$	x^2
Poisson	$\log(\theta)$	θ	x
Binomial	$\log(\theta) - \log(1-\theta)$	$-n \log(1-\theta)$	x
Gamma	$-1/\theta$	$k \log(\theta)$	x

The Gaussian change in mean model is for a variance of 1, the Gaussian change in variance model is for a mean of 0; the Binomial model assumes the number of trials is n ; and the Gamma model is for a change in scale parameter with shape parameter k

and is often the identity function. We do not need to consider $\delta(x)$ as it cancels out in all likelihood ratios.

There are various simple transformations that can be done to shift data points from one assumed exponential family form to another before applying change detection methods, for example binning Exponentially distributed data into time bins to give rise to Poisson data, approximating Binomial(n, θ) data as Poisson($n\theta$) for large n and small θ , or utilising the fact that if $x \sim N(0, 1)$ then $x^2 \sim$

Gamma(1/2, 1/2) to turn a Gaussian change in variance problem into a Gamma change in parameter problem (refer to Sect. 6 for an illustration of this). Nevertheless, the ability to work flexibly in all possible exponential family settings without requiring data pre-processing can be helpful.

The ideas from Sect. 2.2 can be applied to detecting a change in the parameter of a one-parameter exponential family. The main change is to the form of the log-likelihood. For Algorithm 1 we need to store the differences $C_{\tau_i}^{(\tau_j)}(\theta_1)$ in

the log-likelihood for different choices of the changepoint location. This becomes

$$\begin{aligned} \ell(x_{1:T}|\theta_0, \theta_1, \tau_i) \\ - \ell(x_{1:T}|\theta_0, \theta_1, \tau_j) = \\ = [\alpha(\theta_1) - \alpha(\theta_0)] \sum_{t=\tau_i+1}^{\tau_j} \gamma(x_t) \\ - [\beta(\theta_1) - \beta(\theta_0)](\tau_j - \tau_i). \end{aligned}$$

These curves can be summarised in terms of the coefficients of $\alpha(\theta_1) - \alpha(\theta_0)$ and $\beta(\theta_1) - \beta(\theta_0)$, that is $\sum_{t=\tau_i+1}^{\tau_j} \gamma(x_t)$ and $\tau_j - \tau_i$.

The pruning of Algorithm 1 is based on comparing roots of curves. One challenge with implementing the algorithm for general exponential family models is that the roots are often not available analytically, unlike for the Gaussian model, and thus require numerical root finders. However, pruning just depends on the ordering of the roots. The following proposition shows that we can often determine which of two curves has the larger root without having to calculate the value of the root.

Define

$$\bar{\gamma}_{\tau_i:\tau_j} = \frac{1}{\tau_j - \tau_i} \sum_{t=\tau_i+1}^{\tau_j} \gamma(x_t),$$

to be the average value of $\gamma(x_t)$ for $t = \tau_i + 1, \dots, \tau_j$, and define $\theta_1^\tau (\neq \theta_0)$ to be the root of

$$\ell(x_{1:T}|\theta_0, \theta_1^\tau, \tau) - \ell(x_{1:T}|\theta_0, \cdot, T) = 0.$$

Then the following proposition shows that the ordering of the roots is determined by the ordering of $\bar{\gamma}$ values.

Proposition 1 Suppose that for our choice of θ_0 the function

$$\theta_1 \mapsto \frac{\beta(\theta_1) - \beta(\theta_0)}{\alpha(\theta_1) - \alpha(\theta_0)}$$

is strictly increasing. Then the sign of $\bar{\gamma}_{\tau_i:\tau_j} - \bar{\gamma}_{\tau_j:T}$ is the same as the sign of $\theta_1^{\tau_i} - \theta_1^{\tau_j}$.

Proof See Supplementary Material.

In other words, $\theta_1^{\tau_i} > \theta_1^{\tau_j}$ if and only if $\bar{\gamma}_{\tau_i:\tau_j} > \bar{\gamma}_{\tau_j:T}$. Thus we can change the condition in Algorithm 1 that compares the roots of two curves with a condition that compares their $\bar{\gamma}$ values. Or equivalently we can implement Algorithm 1 but with $l_i = \bar{\gamma}_{\tau_i:\tau_{i+1}}$ rather than the root of $\mathcal{C}_{\tau_i+1}^{\tau_i} = 0$.

An immediate consequence of this result is that one-parameter exponential family models that satisfy the condition of Proposition 1 and that have the same value for $\gamma(x)$ will prune exactly the same set of curves. This leads to the

following corollary based on a set of exponential family models with $\gamma(x) = x$, the same as the Gaussian change in mean model of the original FOCuS algorithm.

Corollary 1 The Gaussian (change in mean), Poisson, Binomial, and Gamma variations of the FOCuS algorithm have the same pruning.

A graphical example of this corollary is shown in Fig. 2. More generally we have the following.

Corollary 2 If an exponential family model satisfies the condition of Proposition 1, then the pruning under this model will be identical to the pruning of FOCuS for the Gaussian change in mean model analysing data $\gamma(x_t)$.

So, for example, the pruning for the Gaussian change in variance model will be the same as for the Gaussian change in mean model run on data x_1^2, x_2^2, \dots .

One consequence of this corollary is that the strong guarantees on the number of curves that are kept at time T for the original FOCuS algorithm (Romano et al, 2023b) applies to these equivalent exponential family models.

Proposition 2 The computational cost per iteration of FOCuS for a one-parameter exponential family model satisfying the assumption of Proposition 1 will be $O(1)$ for updating the set of curves. Furthermore, if the data is independent and identically distributed, the number of curves stored at iteration T , N_T satisfies

$$E(N_T) < \log(T) + 1.$$

Proof The proof of this result is a direct consequence of Corollary 2 and the results for FOCuS for Gaussian data in Romano et al (2023b). The cost per iteration of updating the curves in FOCuS is $O(1)$ because at each iteration we have a loop over curves, but the number of curves we need to consider is at most one more than the number of curves we prune. Thus the total cost by iteration T is bounded by T plus the number of curves we have pruned. The maximum number of curves to prune is T . Thus the total cost is bounded by a constant times T , and the average per-iteration cost is thus $O(1)$. The bound on the number of curves we store comes directly from the same bound for Gaussian FOCuS, see Theorem 3 of Romano et al (2023b). This follows immediately from the fact that the algorithm for updating which curves to keep is identical to Gaussian FOCuS applied to data $\gamma(x_1), \dots, \gamma(x_T)$, and the bound in Theorem 3 of Romano et al (2023b) for Gaussian FOCuS makes no distributional assumptions on the data other than that they are exchangeable. Exchangeability holds by our assumption of independence. \square

The assumption that the data are identically distributed in this theorem corresponds to the an assumption of no change.

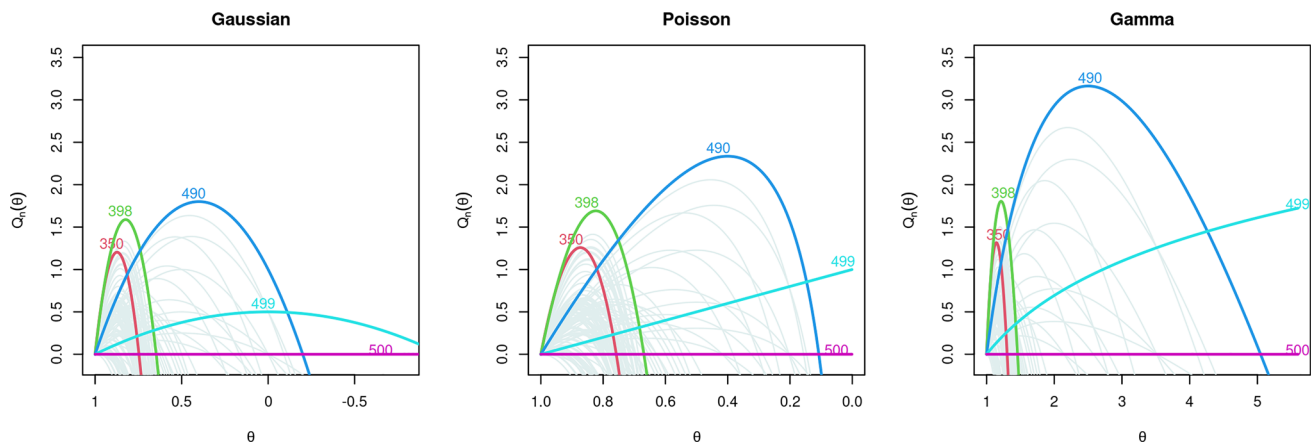


Fig. 2 Comparison of three different cost functions computed from the same realizations $y_1, \dots, y_{500} \sim \text{Poi}(1)$. The leftmost, center, and rightmost figures show the cost function $Q_n(\theta)$ should we assume

respectively a Gaussian, Poisson, or Gamma loss. The floating number refers to the timestep at which each curve was introduced. In gray, the curves that are no longer optimal and hence were pruned

Extending the bound on the number of curves to the case where there has been a change-point is possible, see Romano et al (2023b). The bound on the number of curves corresponds to a bound on the average cost of calculating the test statistic – as this involves finding the maxima of each curve, and then the maximum of these maxima. This has been observed to be the dominant cost of FOCuS Romano et al (2023b), but we show in Sect. 5 how to reduce this cost so that it is empirically $O(1)$.

4 Unknown pre-change parameter

We next turn to consider how to extend the methodology to the case where both pre-change and post-change parameters are unknown. When θ_0 is unknown, the log likelihood-ratio statistic, LR_T , satisfies

$$\frac{LR_T}{2} = \max_{\theta_0, \theta_1, \tau} \left\{ \sum_{t=1}^{\tau} \log f(x_t | \theta_0) + \sum_{t=\tau+1}^T \log f(x_t | \theta_1) \right\} - \max_{\theta_0} \sum_{t=1}^T \log f(x_t | \theta_0).$$

The challenge with calculating this is the first term. Define

$$Q_T^*(\theta_0, \theta_1) = \max_{\tau} \left\{ \sum_{t=1}^{\tau} \log f(x_t | \theta_0) + \sum_{t=\tau+1}^T \log f(x_t | \theta_1) \right\}.$$

If we can calculate this function of θ_0 and θ_1 , it will be straightforward to calculate the likelihood-ratio statistic. If we fix θ_0 and consider Q_T^* as a function of only θ_1 then this is just the function $Q_T(\theta_1)$ we considered in the known pre-change parameter.

As before, we can write $Q_T^*(\theta_0, \theta_1)$ as the maximum of a set of curves, now of two variables θ_0 and θ_1 , with each function relating to a specific value of τ . As before if we can easily determine the curves for which values of τ contribute to the maximum, we can remove the other functions and greatly speed-up the calculation of Q_T^* .

To do this, consider $Q_T^*(\theta_0, \theta_1)$ as a function of θ_1 only, and write this as $Q_{T, \theta_0}(\theta_1)$. Algorithm 1 gives us the curves that contribute to this function for $\theta_1 > \theta_0$. This set of curves is determined by the ordering of the roots of the curves, i.e. the l_i for $i \geq 1$ in Algorithm 1. If we now change θ_0 , the roots of the curves will change, but by Proposition 1 the orderings will not. The only difference will be with the definition of l_0 . That is as we reduce θ_0 we may have additional curves that contribute to the maximum, due to allowing a larger range of values for θ_1 , but as we increase θ_0 we can only ever remove curves. I.e. we never swap the curves that need to be kept. Thus if we run Algorithm 1 for $\theta_0 = -\infty$, then the set of curves we keep will be the set of curves that contribute to $Q_T^*(\theta_0, \theta_1)$ for $\theta_1 > \theta_0$.

In practice, this means that to implement the pruning of FOCuS with pre-change parameter unknown, we proceed as in Algorithm 1 but set $l_0 = -\infty$ when considering changes $\theta_1 > \theta_0$, and $l_0 = \infty$ when considering changes $\theta_1 < \theta_0$. The equivalence of Algorithm 1 across different exponential family models, that we demonstrated with Corollary 2, also immediately follows.

5 Adaptive maxima checking

The main computational cost of the FOCuS algorithm comes from maximising the curves at each iteration. This is particularly the case for non-Gaussian models, as maximising a curve requires evaluating $\max_{\theta_0, \theta_1} \ell(x_{1:T}|\theta_0, \theta_1, \tau)$, which involves computing at least one logarithm (as in the cases of Poisson, Binomial, Gamma data). As the number of curves kept by time T is of order $\log(T)$, calculating all maxima represents a (slowly) scaling cost. However we can reduce this cost by using information from previous iterations so that we need only maximise over fewer curves in order to detect whether Q_T is above or below our threshold. This is possible by obtaining an upper bound on Q_T that is easy to evaluate, as if this upper bound is less than our threshold we need not calculate Q_T .

The following proposition gives such an upper bound on the maximum of all, or a subset, of curves. First for $\tau_i < \tau_j$, we define the likelihood ratio statistic for a change at τ_i with the signal ending at τ_j . Define this likelihood ratio statistic as

$$m_{\tau_i, \tau_j} = \max_{\substack{\theta_0 \in H_0, \\ \theta_1}} \ell(x_{1:\tau_j}|\theta_0, \theta_1, \tau_i) - \max_{\theta_0 \in H_0} \ell(x_{1:\tau_j}|\theta_0, \cdot, \tau_j),$$

where H_0 denotes the set of possible values of θ_0 . H_0 will contain a single value in the pre-change parameter known case, or be \mathbb{R} for the pre-change parameter unknown case.

Proposition 3 For any $\tau_1 < \tau_2 < \dots < \tau_n < T$, we have

$$\max_{i=1, \dots, n} m_{\tau_i, T} \leq \sum_{i=1}^{n-1} m_{\tau_i, \tau_{i+1}} + m_{\tau_n, T}.$$

Proof See Supplementary Material. A pictorial explanation of the result is also shown in Fig. 3.

We can use this result as follows. The sum $M_{\tau_k} := \sum_{i=1}^{k-1} m_{\tau_i, \tau_{i+1}}$ can be stored as part of the likelihood curve for τ_k , and the maxima checking step can proceed as in Algorithm 2. The idea is that we can bound Q_T above by $m_{\tau_k, T} + M_{\tau_k}$. So, starting with the curve with largest τ_k value we check if $m_{\tau_k, T} + M_{\tau_k}$ is below the threshold. If it is, we know Q_T is below the threshold and we can output that no change is detected without considering any further curves. If not, we see if $m_{\tau_k, T}$, the likelihood-ratio test statistic for a change at τ_k is above the threshold. If it is we output that a change has been detected. If not then we proceed to the curve with the next largest τ_k value and repeat.

Empirical results suggest that for $\tau_1 \dots \tau_n \in \mathcal{I}_T$ when searching only for an up-change (or analogously only for a down-change), the upper bound in Proposition 3 is quite

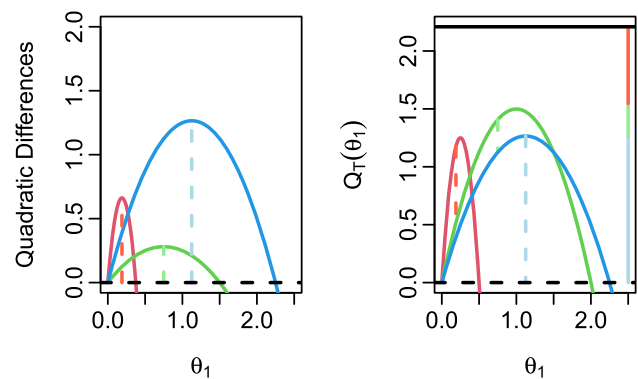


Fig. 3 Example of the bound of Proposition 1 for the pre-change mean known case. Left-hand plot shows the differences between the three curves that contribute to $Q_T(\theta_1)$. The m_{τ_i, τ_j} values correspond to the maximum of these curves (vertical lines). Right-hand plot shows $Q_T(\theta)$, the three curves that define it, and the maximum difference between the curves (vertical bars). The bound is the sum of the maximum differences (right-most stacked line)

Algorithm 2 FOCuS maxima check at time T for $\theta_1 \geq \theta_0$.

Input: A set of n likelihood curves and associated (τ_k, M_{τ_k}) values.

```

1: Set  $k = n$ 
2: while  $k > 0$  do
3:   Calculate  $m_{\tau_k, T}$ 
4:   if  $m_{\tau_k, T} + M_{\tau_k} < \text{Threshold}$  then
5:     break
6:   else
7:     Return change on  $[\tau_k, T]$ 
8:   end if
9:    $k \leftarrow k - 1$ 
10: end while

```

Output: Return no change.

tight under the underlying data scenario of no change because most of the $m_{\tau_i, \tau_{i+1}}$ are very small. Furthermore, as we show in Sect. 6, at the majority of time-steps only one curve needs to be checked before we know that Q_T is less than our threshold.

6 Numerical examples

We run some examples to empirically evaluate the computational complexity of the FOCuS procedure, comparing the various implementations presented in this paper with those already present in the literature.

In Fig. 4 we show the number of floating point operations as a function of time. The Figure was obtained by averaging results from 50 different sequences of length 1×10^6 . Results were obtained under the Bernoulli likelihood. Under this likelihood, the cost for an update is negligible, given that this involves integer operations alone, and this allows for a better comparison of the costs of pruning and checking the

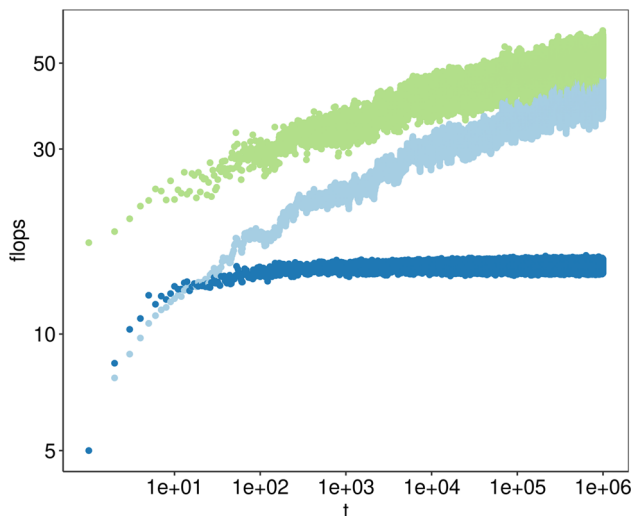


Fig. 4 Flops per iteration in function of time for three FOCuS implementations. In green, the flops for FOCuS with pruning based on calculating the roots l_1, \dots, l_n numerically. In light blue, FOCuS with the average value pruning. In blue, finally, FOCuS with the average value pruning and the adaptive maxima checking. Log-scale on both axes

maxima. We compare three different FOCuS implementations: (i) FOCuS with pruning based on the ordered roots l_1, \dots, l_n , where such roots are found numerically through the Newton–Raphson procedure, (ii) FOCuS with the average value pruning of Sect. 3 and lastly (iii) FOCuS with the average value pruning and the adaptive maxima checking of Sect. 5.

We note that avoiding explicitly calculating the roots leads to a lower computational overhead when compared to Newton–Raphson. The best performances are, however, achieved with the addition of the adaptive maxima checking procedure, where we find a constant per iteration computational cost under the null centered around 15 flops per iteration. Without the adaptive maxima checking, the maximisation step is the most computationally demanding step of the FOCuS procedure, as we need to evaluate $\mathcal{O}(\log(T))$ curves per iteration.

In Fig. 5 we place a change at time 1×10^5 and we focus on the number of curves stored by FOCuS, and the number of curves that need to be evaluated with the adaptive maxima checking. Furthermore, for comparison, we add a line for the naive cost of direct computation of the CUSUM likelihood-ratio test. We can see how, before we encounter a change, with the adaptive maxima checking routine we only need to maximise on average 1 curve per iteration, as compared to about 7.4 for the standard FOCuS implementation. After we encounter a change, then, the number of curves that need evaluation increases, as the likelihood ratio statistics increases and it is more likely to meet the condition of Proposition 3. As it can be seen from the short spike after the change, this

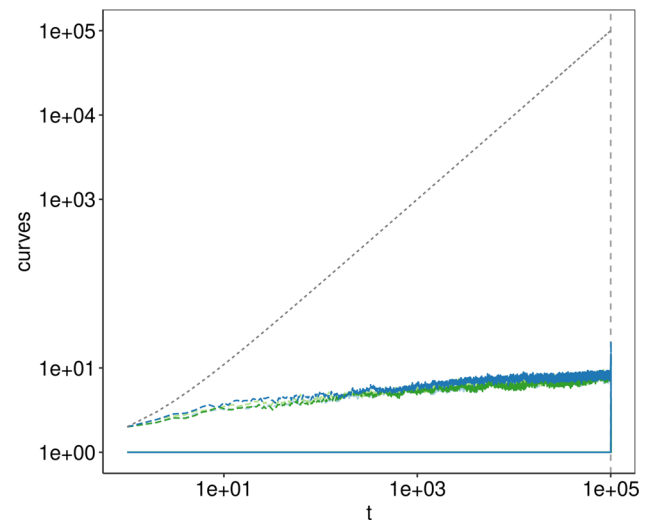


Fig. 5 Number of curves to store and evaluations per iteration in function of time. The grey dotted line is the naive cost of computing the CUSUM likelihood ratio test. The dashed line are the number of curves stored by FOCuS over some Gaussian (light-green), Poisson (dark-green), Bernoulli (light-blue) and Gamma (dark-blue) realizations. The solid lines are the number of curves that need to be evaluated at each iteration with the adaptive maxima checking. Log-scale on both axes

only occurs for a short period of time preceding a detection. This empirically shows that FOCuS is $\mathcal{O}(1)$ computational complexity per iteration while being $\mathcal{O}(\log T)$ in memory, as we still need to store in memory on average $\mathcal{O}(\log T)$ curves.

To illustrate the advantages of running FOCuS for the correct exponential family model, we consider detecting a change in variance in Gaussian data with known mean. We will assume that we have standardised the data so it has mean zero. A common approach to detecting a change in variance is to detect a change in mean in the square of the data (Inclan and Tiao, 1994), so we will compare FOCuS for Gaussian change in mean applied to the square of the data against FOCuS for the Gaussian change in variance model (as in Table 1).

For a process distributed under the null as a normal centered on 0 with variance $\theta_0 = 1$, we present 5 simulations scenarios for $\theta_1 = 0.75, 1.25, 1.5, 1.75$ and 2. Each experiment consists of 100 replicates. Thresholds were tuned via a Monte Carlo approach to achieve an average run length of 1×10^5 under the null in the same fashion of (Chen et al 2022, Section 4.1). We then introduce a change at time 1000 and measure performances in terms of detection delay (the difference between the detection time and the real change).

In Fig. 6 we illustrate the scenarios and present results in terms of the proportion of detections within t observations following the change.

For a positive change large enough, e.g. for $\theta_1 = 2$, there is only a small advantage in employing the Gaussian change-in-variance model over the Gaussian change-in-mean applied

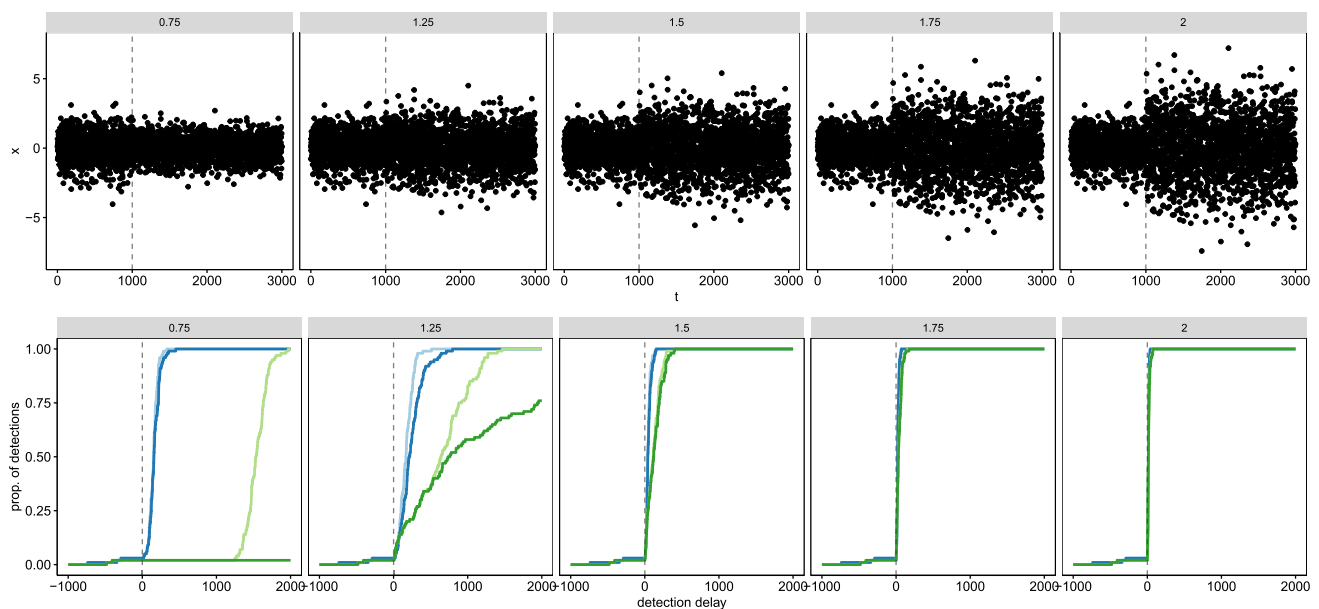


Fig. 6 Empirical evaluation of FOCuS for Gaussian change-in-variance. Top row: example sequences for our simulation scenarios, with labels indicating the post-change parameter θ_1 , whilst vertical dotted line refers to the changepoint location τ . Bottom row: proportion of detections as a function of the detection delay for Gaussian change-in-variance model with pre-change parameter known (light blue) and

unknown (dark blue), and Gaussian change-in-mean applied to square of the data with pre-change parameter known (light green) and unknown (dark green). The vertical dotted line this time indicates the start of the change: the faster we get to 1 following the change, the better. Prior to the vertical line, we are essentially counting false positives

to the square of the data. However, as we lower the signal-to-noise ratio and shift towards more subtle changes, we can see how using the correct model gives an increasing advantage in terms of reducing the detection delay.

7 Discussion

We have presented an algorithm for online changepoint detection for one-parameter exponential family models that (i) exactly performs the likelihood-ratio test at each iteration; and (ii) empirically has a constant cost per-iteration. To the best of our knowledge, it is the first algorithm that achieves both of these.

The algorithm can only detect changes in a single parameter, and thus can only analyse univariate data. However this can provide the building block for analysing multivariate data. For example Mei (2010) propose online monitoring multiple data streams by calculating statistics for a change for each individual data stream and then combining this information. There is an extensive literature on how one can combine such information in an efficient way (for example Cho and Fryzlewicz 2015; Enikeeva and Harchaoui 2019; Tickle et al 2021).

A further challenge would be to extend the algorithm to deal with time-dependent data. Often methods that assume independence work well even in the presence of autocor-

relation in the data providing one inflates the threshold for detecting a change (Lavielle and Moulines, 2000). If the autocorrelation is strong, such a simple approach can lose some power, and either applying a filter to the data to remove the autocorrelation (Chakar et al, 2017) or adapting FOCuS to model it (building on ideas in Romano et al 2022; Cho and Fryzlewicz 2024; Hallgren et al 2021) may be better.

Supplementary Information The online version contains supplementary material available at <https://doi.org/10.1007/s11222-024-10416-6>.

Acknowledgements This work was supported by the EPSRC grants EP/N031938/1 and EP/R004935/1, and BT as part of the Next Generation Converged Digital Infrastructure (NG-CDI) Prosperity Partnership.

Author Contributions All authors contributed to the main ideas and the writing of the paper. Kes Ward proposed the idea for adaptive maxima checking, and Gaetano Romano performed the simulation study.

Declarations

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material

in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Adams RP, MacKay DJ (2007) Bayesian online changepoint detection. arXiv preprint [arXiv:0710.3742](https://arxiv.org/abs/0710.3742)
- Andreou, E., Ghysels, E.: Detecting multiple breaks in financial market volatility dynamics. *J. Appl. Economet.* **17**(5), 579–600 (2002)
- Aue A, Kirch C (2024) The state of cumulative sum sequential change point testing seventy years after Page. *Biometrika* p To appear
- Barrett E, Bolz-Tereick CF, Killick R, et al (2017) Virtual machine warmup blows hot and cold. In: *Proceedings of the ACM on Programming Languages* 1(OOPSLA), pp. 1–27
- Beaulieu, C., Killick, R.: Distinguishing trends and shifts from memory in climate data. *J. Clim.* **31**(23), 9519–9543 (2018)
- Chakar, S., Lebarbier, E., Lévy-Leduc, C., et al.: A robust approach for estimating change-points in the mean of an AR(1) process. *Bernoulli* **23**(2), 1408–1447 (2017)
- Chen, Y., Wang, T., Samworth, R.J.: High-dimensional, multiscale online changepoint detection. *J. Royal Stat. Soc. (Series B)* **84**, 234–266 (2022)
- Chen, Z., Tian, Z.: Modified procedures for change point monitoring in linear models. *Math. Comput. Simul.* **81**(1), 62–75 (2010)
- Cho, H., Fryzlewicz, P.: Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *J. Royal Stat. Soc. Series B (Statistical Methodology)* **77**(2), 475–507 (2015)
- Cho, H. and Fryzlewicz, P. (2024), Multiple change point detection under serial dependence: Wild contrast maximisation and gappy Schwarz algorithm. *J. Time Ser. Anal.* <https://doi.org/10.1111/jtsa.12722>
- Chu, C.S.J., Hornik, K., Kaun, C.M.: MOSUM tests for parameter constancy. *Biometrika* **82**(3), 603–617 (1995)
- Eichinger, B., Kirch, C.: A mosum procedure for the estimation of multiple random change points. *Bernoulli* **24**(1), 526–564 (2018)
- Enikeeva, F., Harchaoui, Z.: High-dimensional change-point detection under sparse alternatives. *Ann. Stat.* **47**(4), 2051–2079 (2019)
- Fearnhead, P., Liu, Z.: On-line inference for multiple changepoint problems. *J. R. Stat. Soc. Ser. B Stat Methodol.* **69**(4), 589–605 (2007)
- Hallgren KL, Heard NA, Adams NM (2021) Changepoint detection in non-exchangeable data. arXiv preprint [arXiv:2111.05054](https://arxiv.org/abs/2111.05054)
- Inclan, C., Tiao, G.C.: Use of cumulative sums of squares for retrospective detection of changes of variance. *J. Am. Stat. Assoc.* **89**(427), 913–923 (1994)
- Iwata T, Nakamura K, Tokusashi Y, et al (2018) Accelerating online change-point detection algorithm using 10 GbE FPGA NIC. In: *European Conference on Parallel Processing*, Springer, pp 506–517
- Keriven, N., Garreau, D., Poli, I.: NEWMA: a new method for scalable model-free online change-point detection. *IEEE Trans. Signal Process.* **68**, 3515–3528 (2020)
- Keshavarz, H., Michailidis, G., Atchadé, Y.: Sequential change-point detection in high-dimensional Gaussian graphical models. *J. Mach. Learn. Res.* **21**(82), 1–57 (2020)
- Kirch, C., Weber, S., et al.: Modified sequential change point procedures based on estimating functions. *Electron. J. Stat.* **12**(1), 1579–1613 (2018)
- Lai, T.L.: Sequential changepoint detection in quality control and dynamical systems. *J. Roy. Stat. Soc.: Ser. B (Methodol.)* **57**(4), 613–644 (1995)
- Lai, T.L., Xing, H.: Sequential change-point detection when the pre-and post-change parameters are unknown. *Seq. Anal.* **29**(2), 162–175 (2010)
- Lavielle, M., Moulines, E.: Least-squares estimation of an unknown number of shifts in a time series. *J. Time Ser. Anal.* **21**(1), 33–59 (2000)
- Li, Z., Zou, C., Gong, Z., et al.: The computation of average run length and average time to signal: an overview. *J. Stat. Comput. Simul.* **84**(8), 1779–1802 (2014)
- Lorden, G.: Procedures for reacting to a change in distribution. *Ann. Math. Stat.* **42**(6), 1897–1908 (1971)
- Lucas, J.M.: Counted data CUSUMs. *Technometrics* **27**(2), 129–144 (1985)
- Mei, Y.: Efficient scalable schemes for monitoring a large number of data streams. *Biometrika* **97**(2), 419–433 (2010)
- Page, E.: A test for a change in a parameter occurring at an unknown point. *Biometrika* **42**(3/4), 523–527 (1955)
- Page, E.S.: Continuous inspection schemes. *Biometrika* **41**(1/2), 100–115 (1954)
- Romano, G., Rigai, G., Runge, V., et al.: Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise. *J. Am. Stat. Assoc.* **117**, 1–16 (2022)
- Romano G, Eckley I. A., Fearnhead P.: A log-linear non-parametric online changepoint detection algorithm based on functional pruning. *IEEE Trans. Signal Process.*, vol. 72, pp. 594–606, (2024) <https://doi.org/10.1109/TSP.2023.3343550>
- Romano, G., Eckley, I., Fearnhead, P., et al.: Fast online changepoint detection via functional pruning CUSUM statistics. *J. Mach. Learn. Res.* **24**(81), 1–36 (2023)
- Ross, G.J., Adams, N.M.: Two nonparametric control charts for detecting arbitrary distribution changes. *J. Qual. Technol.* **44**(2), 102–116 (2012). <https://doi.org/10.1080/00224065.2012.11917887>
- Ross, G.J., Tasoulis, D.K., Adams, N.M.: Nonparametric monitoring of data streams for changes in location and scale. *Technometrics* **53**(4), 379–389 (2011). <https://doi.org/10.1198/TECH.2011.10069>
- Sakamoto Y, Fukui KI, Gama J, et al (2015) Concept drift detection with clustering via statistical change detection methods. In: *2015 Seventh International Conference on Knowledge and Systems Engineering (KSE)*, IEEE, pp 37–42
- Tickle, S.O., Eckley, I., Fearnhead, P.: A computationally efficient, high-dimensional multiple changepoint procedure with application to global terrorism incidence. *J. Royal Stat. Soc. Series A (Statistics in Society)* **184**(4), 1303–1325 (2021)
- Varghese B, Wang N, Barbhuiya S, et al (2016) Challenges and opportunities in edge computing. In: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pp 20–26, 10.1109/SmartCloud.2016.18
- Ward, K., Dilillo, G., Eckley, I., & Fearnhead, P. (2023). Poisson-FOCuS: An efficient online method for detecting count bursts with application to gamma ray burst detection. *J. Am. Stat. Assoc.*, 1–13. <https://doi.org/10.1080/01621459.2023.2235059>
- Yu, Y., Madrid Padilla, O. H., Wang, D., & Rinaldo, A. (2023). A note on online change point detection. *Sequential Analysis*, 42(4), 438–471. <https://doi.org/10.1080/07474946.2023.2276170>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.