Statistical Anomaly Detection for Streaming Data under Computational Constraint

Kes Ward, B.A.(Hons.), M.Res

This thesis is submitted for the degree of Doctor of Philosophy at Lancaster University,

Department of Mathematics and Statistics

September 2024

STOR-i
excellence with impact

Abstract

This thesis develops the Functional Online Cumulative Sum (FOCuS) anomaly detection method for detecting collective anomalies in streaming data under conditions of computational constraint. FOCuS performs a sequential likelihood ratio test for the presence of an anomaly in all intervals within a data stream, while only requiring a small constant cost per update. The FOCuS method is adapted from its original Gaussian form to work with Poisson-distributed count data, and extended to the wider one-parameter Exponential family model setting. Further extensions to FOCuS to find collective anomalies in multivariate streaming data are also examined. This thesis contains real applications with different kinds of computational constraints. These include handheld radiation monitoring devices with a limited battery life, and cube satellites detecting gamma ray bursts with limited processing power on board.

Acknowledgements

"An anomaly is not an abnormality.

Diversity does not signify sickness."

Georges Canguilhem, medical philosopher.

This thesis is dedicated to everyone I've ever complained about my PhD struggles to, for the support you've all given me to pull me through it.

Declaration

I, Kes Ward, confirm that this thesis is the result of my own work and has not been submitted for a higher degree at this or any other university. This thesis includes nothing that is the outcome of work done in collaboration except where specifically indicated here:

1. The ideas in this thesis were the product and subject of discussions with my academic supervisors Idris Eckley and Paul Fearnhead of Lancaster University, and my industrial supervisor Trevor Burbridge of BT.

2. My thesis examiners David Leslie of Lancaster University, Nick Heard of Imperial College London, and Ronni Bowman of the Southampton Statistical Sciences Research Institute have also contributed to this final thesis through my viva examination and feedback.

3. The simulated lightcurve generation and computational search of the HEASARC Fermi-GBM archive within Chapter 3 is work by Giuseppe Dilillo of the University of Udine, Italy, as part of a joint collaboration on an academic paper.

4. Access to and use of the data in Chapter 4 was enabled by data curation work done by the NuSec Sigma Data Challenge team at the University of Surrey.

5. The simulations within Chapter 5 are work by Gaetano Romano of Lancaster University, as part of a joint collaboration on an academic paper.

Excerpts of this thesis and the work which underlies it have been previously published in the following academic publications:

1. 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.' *Journal of the American Statistical Association.* 1–13 (with discussion at Joint Statistical Meetings 2024)

2. Crupi, R., Dilillo, G., Bissaldi, E., Ward, K., Fiore, F. & Vacchi, A. (2023) 'Searching for long faint astronomical high energy transients: a data driven approach.' *Experimental Astronomy*

3. Dilillo, G. Ward, K., Eckley, I., Fearnhead, P. et al. (2024) 'Gamma-ray burst detection with Poisson-FOCuS and other trigger algorithms.' *The Astrophysical Journal* 962, 137

4. Ward, K., Romano, G., Eckley, I. & Fearnhead, P. (2024) 'A constant-per-iteration likelihood ratio test for online changepoint detection for exponential family models.' *Statistics and Computing*

I confirm that I have followed to the best of my ability all relevant Lancaster University and EPSRC guidelines on responsible and ethical research conduct, including but not limited to the UKRI Open Access policy and the Concordat to Support Research Integrity.

Signed: **Kes Ward**

Date: 16/07/2025

This thesis is approximately 57,000 words.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Why do anomalies matter? This is a fundamental question that extends beyond algorithms or statistical distributions. Anomaly detection is, at its core, the pursuit of understanding the unusual. Its importance can be viewed as a reflection of how we strive to comprehend the unexpected in life to adapt to the challenges it brings. In every system — whether it be a computer system, a physical measurement of the world, or even our own consciousness — anomalies represent moments of disruption, signaling that something significant is occurring.

Much has been written about the topic of anomaly detection over the past hundred years. With the rise in available computers, there has been a large increase in the amount of data being collected. This, in turn, has created the need for algorithms able to process and work with such data. In many areas of monitoring, the aim is to ensure a steady state of operations, and any deviations from that state are of interest to inform a decision. This may be tackling an emerging issue, or finding something novel.

This thesis is about anomaly detection in the streaming setting, with a particular focus on methods that scale well to large volumes of data. It is also about the practical realities of the situations in which anomaly detection methods are implemented, and contains a number of industrial applications.

1.1 The central tradeoff in anomaly detection

The anomaly detection methods in this thesis are for use on high-frequency time series data. These applications deal with a central three-way tradeoff, as in Figure 1.1.1:

1. We must find as many true anomalies as we can, as quickly and accurately as possible.

2. We must have low rates of false detections when anomalies are not present.

3. The method must be able to run fast: on a lot of data, using limited computational resources, taking a short amount of time.

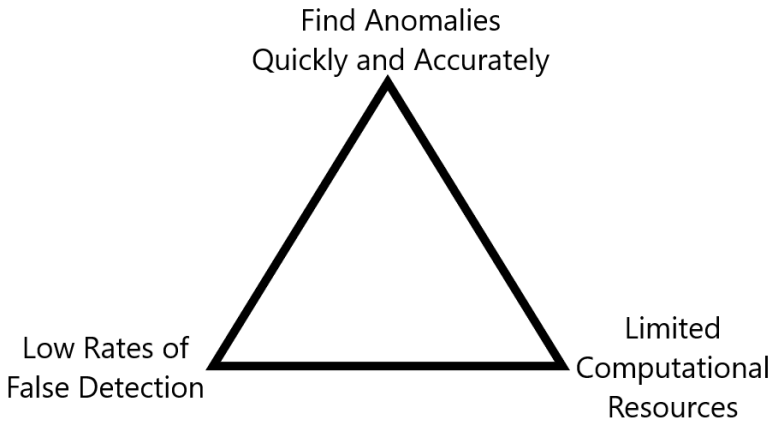


Figure 1.1.1: The central three-way tradeoff of anomaly detection considered in this thesis.

Designing a fast anomaly detection method mainly requires the method to perform fast in the presence of no anomalies (which, due to the rare nature of anomalies, is likely to be the vast majority of the time). It does not require the method to perform fast when an anomaly is present, only to identify that anomaly accurately as soon as possible after it occurs.

Similarly, because anomalies are so rare, even a small increase in the probability of detecting an anomaly when one is not actually present can have a large decrease on the likelihood of a detection representing a true anomaly.

1.2 What makes up an anomaly detection method?

An anomaly is an observation in a dataset that is sufficiently surprising that it gives good reason to suspect it was generated using a different process than was used to generate the rest of the data (Atkinson and Hawkins 1981). In order to find these unusual observations, practitioners can employ any one of a large variety of anomaly detection methods.

Any anomaly detection method answers four distinct questions:

1. How do we *conceptualise* an anomaly?
2. How can we represent this general concept with a *specific mathematical definition*?
3. What is the *algorithmic procedure* we use to process the data in order to compute this definition, and what is the computational complexity of that algorithm?
4. What *parameters* does the general method have, and how does varying them produce different specific methods?

Computational complexity is often represented in terms of overall dataset size. In a time series, computational complexity can instead be recorded for the computation required at each timestep. We may also care about the memory cost of an algorithm in terms of the amount of space required in working memory, although this is often less important than computational complexity.

1.3 What anomaly detection method is this thesis about?

This thesis is primarily about the Functional Online Cumulative Sum (FOCuS) method (Romano, Eckley, Fearnhead, and Rigaill 2023), and its extensions to different settings. These include the count data setting with varying background rates (Ward, Dilillo, et al. 2023), the general one-parameter exponential family model setting (Ward, Romano, et al. 2024), the specific multivariate setting of spectral radiation counts for the detection of specific isotopes, and the general multivariate setting to address scaling challenges. A

description of the prior methods from which FOCuS is derived can be found in section 2.3.8, and the algorithm itself is defined in Chapter 5.

We place FOCuS in its wider context by answering the questions given above.

1. The concept: An anomaly is an interval in a time series signal where the signal is higher than you would expect. We want to find these intervals as soon as we can.

2. The mathematical definition: An anomaly is an interval $X_\tau, ..., X_T$ where the generalised log-likelihood ratio test statistic for the data generating process shows that we have a significant increase of the mean of the signal over the background mean. For more explanation on this, see Section 2.3.5. While FOCuS is often used with data that can be assumed to follow a Normal distribution, other likelihood functions are possible to implement with the method. We work with Poisson likelihoods for count data in Chapter 3 and Chapter 4, a general one-parameter exponential family likelihood in Chapter 5, and multidimensional Normal likelihoods in Chapter 6.

3. The algorithm: Using ideas of functional pruning, we avoid ever iterating over most of the signal intervals even though we are computing a statistic maximised over them. For more explanation, see Section 3.3 or Section 5.5. In particular, the expected computational and memory costs for FOCuS are $O(1)$ at each timestep, and low. This allows FOCuS and methods based on it to address settings where we are computationally constrained, and more computation-heavy methods would not be viable.

4. The parameters: we have a statistical threshold parameter k detailing what significance we should use for our likelihood ratio test. We relate k to the concept of a k-sigma event: if the interval was of length one, then it would be a point roughly k standard deviations from the mean. Longer intervals can be less intense while still being detectable at this statistical threshold, as shown in Figure 1.3.2.

   We have an (optional) minimum anomaly intensity parameter $\mu_{\min} > \mu_0$, which is

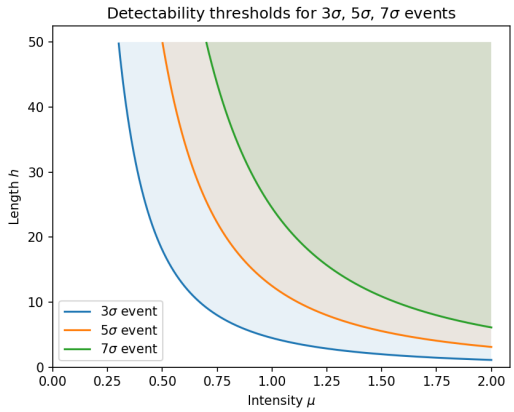


Figure 1.3.2: Graph showing the tradeoff between the length h and mean μ of an anomaly in a Normally distributed time series needed to be detectable at different statistical thresholds.

used to restrict our algorithm to only considering anomalies of intensity $\mu \geq \mu_{\min}$ that are statistically detectable in intervals under a chosen length. This helps our method be robust to the artefacts caused by estimating a moving background rate, which is a common problem with many practical applications. Raising $\mu_{\min}$ looks for anomalies that are detectable over shorter intervals. This is discussed in Chapters 3 and 6.

We have an (optional) clearing parameter h_{clear}, which we use to reset the algorithm after passing over large anomalies. If there is no evidence for an anomaly beginning or continuing over the last h_{clear} time steps, then we stop considering any intervals beginning further in the past. By default we have $h_{\text{clear}} = \infty$ if we are not resetting at all, which can cause passing over large anomalies to overshadow intervals a long time into the future. Smaller values of h_{clear} reset the algorithm more often, and should be used if we are interested in telling apart separate anomalies that occur near each other, whereas larger values of h_{clear} will cause them to be regarded as two parts of the same anomaly and therefore more overall statistically detectable. This parameter is discussed in Chapter 4.

We also have suggestions for other parameters specific to the general multivariate setting. We have a local thresholding parameter a that determines how often different signals communicate and how dense or sparse an anomaly would look

like across signals. We also have a backscan parameter W that determines how closely in time anomalies in different signals should be expected to start. These are discussed in more detail in Chapter 6.

1.4 The structure of this thesis

Anomaly detection is a very practical area. This thesis is structured in a way where theory and methods chapters are interspersed with chapters about specific applications.

Chapter 2 is a literature review of the current state of anomaly detection, with a focus on methods that scale and that could be applied to time series data. It provides precise definitions of concepts around anomalies, which are often defined vaguely in the wider literature. It explains the various challenges that can make an anomaly detection problem difficult. These include the computational challenges that come when designing an anomaly detection method, such as searching over intervals and working with data in more than one dimension. It also includes the need to account for the changing background context of your data, and the data artefacts that can arise even when you try to account for it. The literature review surveys the standard, well-used approaches that tackle these problems, as well as some more complex modern methods that can be useful when simpler approaches fail. Finally, the review covers ways to assess an anomaly detection method, and looks at some application domains where specific types of anomaly detection problems may arise.

Chapter 3 is based on work done with the HERMES Scientific Pathfinder team about finding gamma-ray bursts using a scintillation detector on a cube satellite. The main challenge for this application is the high velocity of the data and the resulting challenges around computational cost when working with intervals. We adapt the FOCuS algorithm for Normally distributed data in order to work in a Poisson count data setting. We show how this can smooth out the ability to detect anomalies of different shapes and sizes compared to using a grid of windows running over the data. This application contains a varying background rate and we show how to ensure that collec-

tive anomaly detection methods are robust to the data artefacts caused by background rate estimation. This chapter was published in the Journal of the American Statistical Association (Ward, Dilillo, et al. 2023), and received an Editor's Choice award. It was selected for discussion for the American Statistical Association's 'Applications and Case Studies' session at the Joint Statistical Meetings 2024.

Chapter 5 introduces more general theoretical improvements to the FOCuS algorithm. We adapt the algorithm more broadly to work with data drawn from a one-parameter exponential family distribution, and prove that in many cases the internal computations of the algorithm are identical across different assumed data forms. We also show how to reduce the computational cost of running FOCuS to a constant cost per iteration. This chapter has been published in the Journal of Statistics and Computing (Ward, Romano, et al. 2024).

Chapter 4 works with radiation data from the NuSec Sigma Data Challenge. Here, we look to build on FOCuS to develop an algorithm that can use small amounts of battery on a handheld device, and can tell different isotope radiation sources apart from each other using the specific multidimensionality of this data source. We ensure that our algorithm is robust to the changing radiation patterns in differing weather conditions. We also develop a methodology for correctly resetting FOCuS after it passes over an anomaly.

Chapter 6, is a theoretical exploration of ways to extend FOCuS to the more general multidimensional setting. We begin by providing further refinements on the constant memory cost bound for different minimum anomaly intensities, which is of particular relevance to the overall total memory cost of a multidimensional setting. We then look at the various ways that the multidimensional problem is hard, including the fact that we must balance scaling well with time and scaling well with dimension, and the fact that methods optimised for one are often not good for the other. We also need to find ways to combine information across data streams about where a collective anomaly starts, given that this estimated location may be different in different data streams. Our approaches point to ways to improve on modern methods by showing how and

why they work and improving their scaling.

We conclude with Chapter 7, which summarises the novel theoretical contributions and impact made in specific applications, and points to possible future work in this area.

Chapter 2

Anomaly detection literature review

2.1 Introduction

A lot has been written about anomaly detection over the past century, because of its wide usefulness in many domains of application. The topic is quite broad, and even with an extensive discussion, it would be difficult to cover every aspect in detail. This literature review focuses on the conceptual underpinnings of different anomaly detection methods, aiming to identify similarities across a wide variety of anomaly detection problems. It aims to be most of use to a person designing an anomaly detection method, by providing a framework against which they can evaluate the method design for different aspects of anomaly detection. The history of specific methods related to anomaly detection in computationally constrained streaming data are also developed in more detail.

2.2 Anomaly detection concepts

There are lots of different types of anomalies and anomaly detection methods we could care about. They are referred to by different names in the anomaly detection literature, with the exact language often depending on the domain.

We begin by defining an anomaly.

Definition 2.2.1 (Anomaly)**.** *Given a dataset* $\{X_1, ..., X_n\}$*, a subset of this dataset is an* ***anomaly*** *if it is both relatively small as a proportion of the dataset and unusual with respect to the other points in the dataset.*

Definition 2.2.1 is vague. Exact definitions of what an anomaly is vary. For example, Atkinson and Hawkins (1981) call anomalies "observations which deviate so much from the other observations as to arouse suspicions that they were generated by a different mechanism than the rest of the data". Chandola, Banerjee, and Kumar (2009) call anomalies "patterns in data that do not conform to a well-defined notion of normal behaviour". However, these definitions have the following in common:

Firstly, something is either an anomaly or it is not. We may also have interest in qualifying different types of non-anomalous data, or different types of anomalies, or providing a quantitative estimate of how anomalous an anomaly is. However, this is not the core purpose of an anomaly detection method as opposed to other methods that operate on a dataset. Anomaly detection problems are examples of binary classification problems: they sort data into exactly two categories.

Secondly, anomalies should be relatively rare in the dataset. It would be impossible for a substantial proportion of the dataset to be an anomaly. It is this characteristic of class imbalance (He and Garcia 2009) that separates anomaly detection problems from other binary classification problems.

Anomalies should also be different from the non-anomalous parts of the dataset in a way that is identifiable only from processing the dataset. If a part of the dataset really is generated by a different mechanism, but this cannot be made apparent from processing the data in any way, then it's not an anomaly. We separate an anomaly (a signature in the data) from the presumed cause of the anomaly (a process occurring in the real world).

Often we will refer to anomalies as precisely what is found by the specific anomaly detection method we are using. However, this can become unclear when we are trying to evaluate an anomaly detection method's performance or to compare more than one anomaly detection method against one another on the same dataset. In these cases, we

would refer to detections by a specific method.

2.2.1 Point anomalies

Atkinson and Hawkins (1981) refers to anomalies as individual observations, rather than as subsets of data. We will refer to individual observations that are anomalies as point anomalies, see Definition 2.2.2.

Definition 2.2.2 (Point anomaly)**.** *Given a dataset* $\{X_1, ..., X_n\}$*, a point* X_k *is a* ***point anomaly*** *if it is unusual with respect to the other points in the dataset.*

This thesis distinguishes between point anomalies and other kinds of anomalies. Many anomaly detection methods are set up to detect point anomalies. Much of this thesis is designed around methods for detecting collective anomalies (see Definition 2.2.8), but methods used for point anomalies are also relevant to the topic, particularly when they occur in multivariate data (see Definition 2.2.6) or are used to make a collective anomaly detection method more robust (see Definition 2.2.5).

Outliers and inliers

Definition 2.2.3 (Outlier)**.** *Given a dataset* $\{X_1, ..., X_n\}$*, a point* X_k *is an* ***outlier*** *if it lies unusually far outside the convex hull created by the non-outlier points in the dataset.*

Definition 2.2.3 is circular and is often of little practical use in classifying points as outliers or not. In practice, outliers are often defined by a specific method as being greater than some multiple of data spread from the center of the data (see Section 2.3.1 for a discussion). For the simple one-dimensional scenario where the convex hull is an interval, there exist tests based on removing a set of outliers and then checking that they do lie unusually far outside the interval containing non-outlier points (Grubbs 1950), and these formed part of the early basis of outlier detection. However, we primarily use this definition of an outlier in order to consider the challenges of detecting other kinds of anomalies that are not outliers.

Definition 2.2.4 (Inlier)**.** *Given a dataset* $\{X_1, ..., X_n\}$*, a point* X_k *is an* ***inlier*** *if it is a point anomaly, but not an outlier.*

Some definitions will refer to all anomalies as outliers (Atkinson and Hawkins 1981; Breunig et al. 2000), and some will differentiate between the two concepts (Li, Zhao, et al. 2020). We differentiate in this thesis because we are often specifically looking for inliers: if the anomalies we were searching for were sufficiently extreme as to be (or contain) outliers in the whole dataset, the problem of detecting them would not require the methods we use.

Inliers can be particularly relevant in the case of more complex data structures such as multivariate (see Figure 2.2.2) or time series (see Figure 2.2.3) data. In the case of one-dimensional data that is not a time series, the conceptual difference between outliers and inliers is that if data is multimodal, outliers must be above or below all modes, whereas inliers could be between the modes, as shown in Figure 2.2.1.

Often, an anomaly detection method of finding inliers consists of two parts: first, a transformation is applied that turns the inliers into outliers in the transformed dataset; then, an outlier detection method is used on the transformed data.

Outliers can cause their own problems. For example, the presence of one extreme outlier can make it hard to detect inliers or other, less extreme outliers. This is because outliers present in the raw data can warp the effects of transformations applied to that data. It's often best to identify and remove outliers before applying such transformations, or to make sure that the transformations applied are robust.

Definition 2.2.5 (Robust method)**.** *Given a mathematical method operating on a dataset* $\{X_1, ..., X_n\}$*, that method is robust if a small number of the* X_i *being arbitrarily large outliers does not cause the method to perform badly on the rest of the data.*

The field of robust statistics was developed in the 1960s (Huber 1964) to address the problem of statistical estimation in the presence of outliers. For a simple statistical method such as estimating the mean of the data, Definition 2.2.5 can be taken

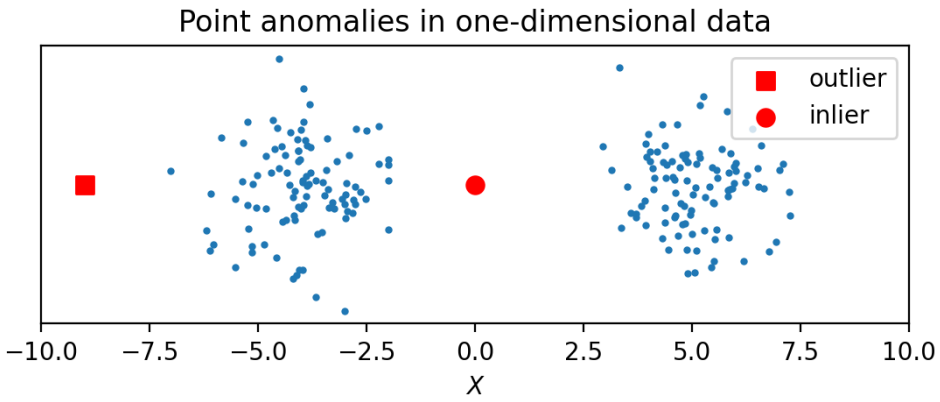


Figure 2.2.1: An example of an outlier and an inlier in a one-dimensional bimodal dataset. The outlier lies outside the modes whereas the inlier is between the modes. Jitter has been added on the y axis for means of easier display.

to mean a bounded influence function (Huber 1964) if "small" means one, or a high breakdown point (Hampel 1971) if more than one. As we focus on methods that may be used in anomaly detection, there is a conceptual difference between Definition 2.2.5 and other definitions within the literature: we are interested not in ignoring outliers but in highlighting them, so we permit the method to give unbounded results on the outliers themselves and still be called robust. Often such methods contain within them robust methods with bounded influence functions or high breakdown points to fit a statistical model and then report the resulting residual. We address one such use of robust statistics for anomaly detection in Section 2.3.4. Chapter 4 also contains the development of a robust method for use with the anomaly detection method developed in this thesis.

Multivariate data

Definition 2.2.6 (Multivariate data)**.** *A dataset* $\{\vec{X_1}, ..., \vec{X_n}\}$ *is multivariate, multidimensional, p-dimensional, or of dimension p if each* $\vec{X_i} := (X_i^1, ..., X_i^p)$ *is a vector of finite dimension p (that does not vary with i).*

A multivariate dataset is one where each point in the dataset has multiple coordinate dimensions, which may or may not have an associated structure to them. This structure may be linear, such as a spectrum of electromagnetic energy bands for radiation data. It might be spatial or network-based, such as sensor data of the same kind of measurement from different sensors at different locations. Or it might be qualitatively different kinds of measurement such as rainfall and wind speed being taken by the same sensor.

Multivariate data may contain different types of anomalies, as illustrated in Figure 2.2.2. In particular, we will concern ourselves with three different types of point anomaly:

1. A point anomaly which is an outlier in at least one of the coordinates alone.
2. An outlier in some subset of the coordinates, that is not an outlier in any one coordinate.
3. An inlier, that only looks anomalous when considering the interaction of those coordinates together. This can happen when the structure of the dataset is irregular.

To find outliers in one coordinate only, it is sufficient to treat that coordinate individually, and use a method that is sufficiently fast that it can be easily repeated for every coordinate.

To find outliers in some subset of coordinates, you first need to select the coordinates that you think contribute to the outlier. This is important for reducing noise in your detection method, which impacts the tradeoff between detection power and false positive rate. You then need to combine information about the outlier across these coordinates. This is often done by calculating an anomaly score for each coordinate,

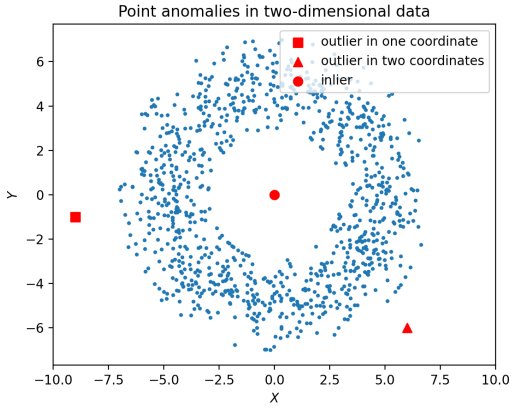


Figure 2.2.2: Three different types of point anomalies in a two-dimensional dataset.

and then comparing the sum of the scores to a threshold. It can also be done by constructing a new coordinate in the direction most useful for identifying the anomaly, and considering the score along this coordinate. For an anomaly score equal to the squared distance between a point and the source of the direction vector, by Pythagoras' theorem these methods give equivalent results.

To find inliers, there are a variety of possible methods in the literature, which are surveyed in Section 2.5. While the precise definition of an anomaly varies between method, generally inlier points are considered anomalous if they lie in some sparse part of the dataset, or are easy to separate out from the rest of the data using a computational procedure.

2.2.2 Time series concepts

Definition 2.2.7 (Ordered data)**.** *A dataset* $(X_1, ..., X_T)$ *is considered ordered, or a sequence, if it relevantly contains the two strict total order relations*

- X_{t_i} *before* X_{t_j}*, defined as* $t_i < t_j$*, and*
- X_{t_i} *after* X_{t_j}*, defined as* $t_i > t_j$*.*

A time series dataset is an ordered dataset indexed by time. Many ordered data sequences are time series, and we will work with time series in this thesis. Other kinds of ordered data include the DNA of a chromosome or natural language.

When finding an anomaly in a time series dataset, there are various problems that arise specific to this setting. These are roughly divided into the presence of temporal structure in the non-anomalous data, the anomalies themselves possibly being collective anomalies rather than point anomalies, and the data being a data stream where the anomaly detection algorithm must run online.

Collective anomalies

Definition 2.2.8 (Collective anomaly)**.** *Given a time series dataset* $(X_1, ..., X_T)$*, a collective anomaly is an anomalous interval* $(X_s, ..., X_e)$*. An interval may be a collective anomaly even if none of the points* $X_s, ..., X_e$ *are point anomalies.*

In the univariate setting, collective anomalies are often characterised by a change in the parameter or distributional family used to generate the data. That is, different points within a collective anomaly share similarities with each other, not just differences with the non-anomalous data.

More generally, collective anomalies represent anomalous batches of data. If these data are all anomalous in the same way, they will not be detected by any algorithm looking for multivariate inliers, as each point is a normal part of the anomalous batch. Collective anomalies can also mask themselves by skewing the results of outlier detection (see Figure 2.3.8). Therefore it can be very important to choose a method specifically looking for collective anomalies rather than relying on point anomaly detection methods.

Temporal structure and contextual anomalies

Definition 2.2.9 (Temporal Structure)**.** *Given a time series dataset* $(X_1, ..., X_T)$ *without anomalies, we say that* X *does not have temporal structure if we can consider the* X_i *independent and identically distributed realisations of the same random variable. We may also call this white noise. Otherwise, we say that* $(X_1, ..., X_T)$ *has some temporal structure.*

Most time series datasets have some temporal structure. Often, this takes the form

of autocorrelation, a trend, or a seasonal pattern. More complicated kinds of temporal structure, for example a cyclic pattern that doesn't have a constant seasonal period, are also possible.

Definition 2.2.10 (Contextual Anomaly)**.** *Given a time series dataset* $(X_1, ..., X_T)$*, a point* X_t *is a* ***contextual anomaly*** *if it is unusual with respect to the other points in the dataset and their ordering.* X_t *might not be considered a point anomaly in the dataset* $\{X_1, ..., X_T\}$ *without the additional context of this ordering.*

There are broadly two types of contextual anomalies we care about which are detected in different ways. The first is an anomaly X_t which is an outlier with respect to its neighbourhood $(X_{t-i}, ..., X_t, ..., X_{t+j})$. Usually, these anomalies are identified by calculating and removing some estimate of trend for the time series.

The second is an anomaly X_t which is an outlier only with respect to its time context t, and not in its neighbourhood. For example, sales figures for December looking similar to those for November and January may indicate an anomalous lack of Christmas bump. These kinds of contextual anomalies are found by calculating and removing some estimate of seasonality or cyclic pattern from the time series.

Trend and seasonality are often both present in a time series. For this reason, they are often calculated together using algorithms that can intelligently separate one from the other. You have to be careful that the presence of outliers doesn't skew this calculation.

This thesis will concentrate on dealing with trend as the most important context for the anomaly detection applications it contains.

Streaming anomaly detection

Definition 2.2.11 (Data stream)**.** *A data stream is a time series dataset* $(X_1, ...)$*, where at time* T *only the dataset* $(X_1, ...X_T)$ *has been observed.* T *is often called the present time.*

Anomaly detection methods for data streams have additional considerations beyond

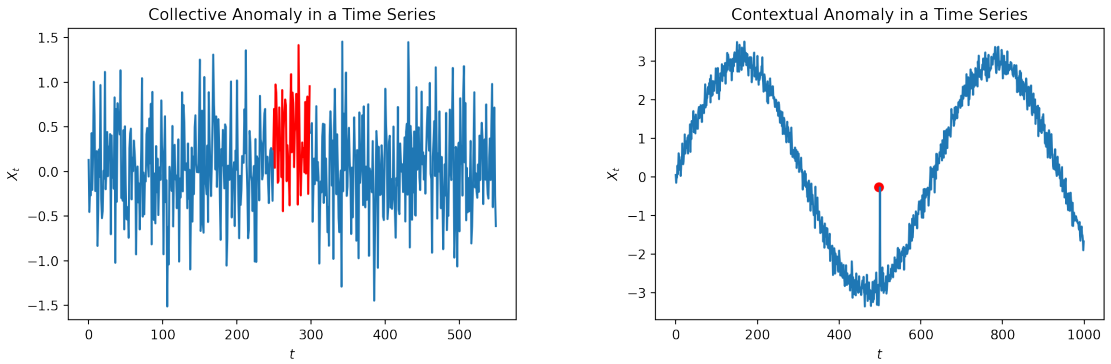


Figure 2.2.3: Two kinds of anomalies in a time series dataset. Neither of these anomalies are outliers in the non-ordered data, and so methods specific to time series must be employed in order to detect them.

those relevant to other time series datasets. For example, we may be interested in at what time an anomaly became detectable, which is different from the estimated time of the anomaly. These are addressed in Section 2.6.4.

Multivariate time series

Definition 2.2.12 (Multivariate time series)**.** *A dataset* $(\vec{X}_1, ..., \vec{X}_T)$ *is a multivariate time series if the dataset* $\{\vec{X}_1, ..., \vec{X}_T\}$ *is multivariate, and for each coordinate* i *the dataset* $(X_1^i, ..., X_T^i)$ *is a time series.*

With a multivariate time series, the definition of things we might wish to consider an anomaly expands further. For example, an anomaly could refer to:

1. One time point in one time series.
2. The same time point across multiple time series.
3. One interval in one time series.
4. The same time interval in multiple time series.
5. One time series, with respect to the others.

If we are hunting for anomalies that represent the same time interval in multiple time series, then the beginning and end of that interval may or may not line up exactly in

the different series.

The anomaly detection method developed in this thesis can be applied to multivariate time series data. The anomalies it detects represent the same time interval in multiple time series. Further exploration of ways to do this well are in Chapter 6.

2.3 The evidence-gathering problem

In order to assess whether or not something is anomalous, you first have to gather evidence from your dataset about what normal data looks like, and then decide how far away from normal should make something anomalous. Here, we look at some ways to do this for a time series.

2.3.1 The three-sigma paradigm

One of the most standard and well-known ways of detecting anomalies in a noisy signal is to calculate an estimate for the mean μ and standard deviation σ of the signal, and label a point as anomalous if it lies more than n standard deviations from the mean. Often $n = 3$ is chosen, and such anomaly detectors are sometimes called three-sigma anomaly detectors.

If the signal is already known to be stationary, and the mean and standard deviation are known in advance, then they can be given as fixed bounds and the same test is applied to every point. For example, if our signal is assumed to have mean $\mu = 0$ and variance $\sigma^2 = 1$, then any points outside the interval $[-3, 3]$ will be marked anomalous. For example, Figure 2.3.4 shows a signal of size $T = 50$ drawn from a $N(0, 1)$ distribution compared to these bounds.

Three-sigma anomaly detectors are quick to implement, easy to understand, and give good results on a wide variety of practical applications.

However, often the mean and variance of the data are not known in advance, and need to be estimated. In the offline setting it's possible to use the entire dataset to estimate a constant mean and variance. For example, Figure 2.3.5 shows the three

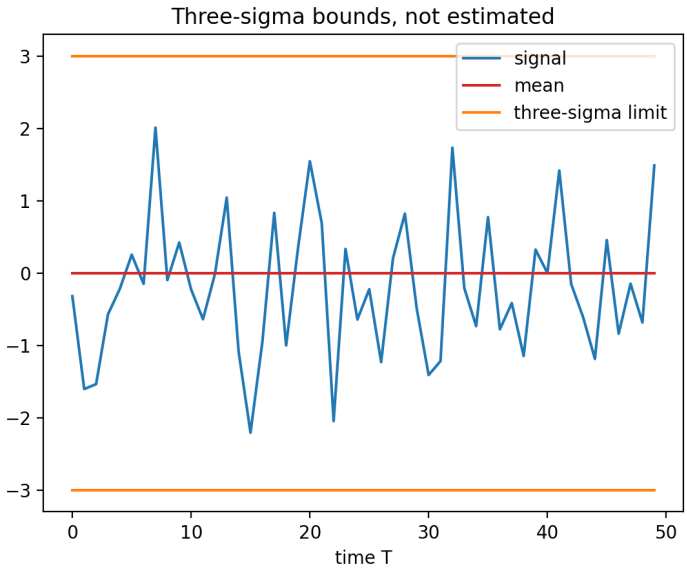


Figure 2.3.4: The flat bounds of a three-sigma anomaly detector

sigma bounds given by the sample mean and sample variance, as well as extended bounds given by using the upper point of 95% confidence intervals for both estimators.

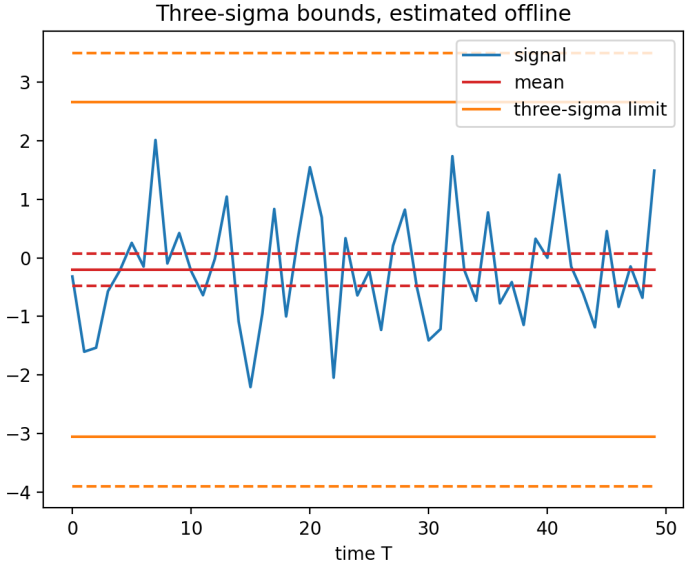


Figure 2.3.5: Three-sigma bounds estimated from the data, with 95% confidence intervals shown with dashed lines.

2.3.2 Burn-in periods

In the online setting when working with a data stream, only the data from time $t \leq T$ is available to estimate the mean and variance at time T. Because we are checking if point x_T is anomalous, we would only want to use $t < T$ to construct our estimators. This means our estimates will start out with uncertainty that resolves itself as we scan through more of the dataset. If we do not correct for this, points labelled anomalous in the beginning of the algorithm's run may not actually be so, having been mislabelled due to imprecise estimates with small sample sizes.

For example, Figure 2.3.6 shows the same sample mean and sample variance estimates for the three sigma bounds as in Figure 2.3.5, but this time only calculated using the sample $t < T$ at time T. Here, the signal point at time $T = 7$ would be incorrectly marked as anomalous if using only the estimators, but this point is well within the bounds given by using the confidence intervals.

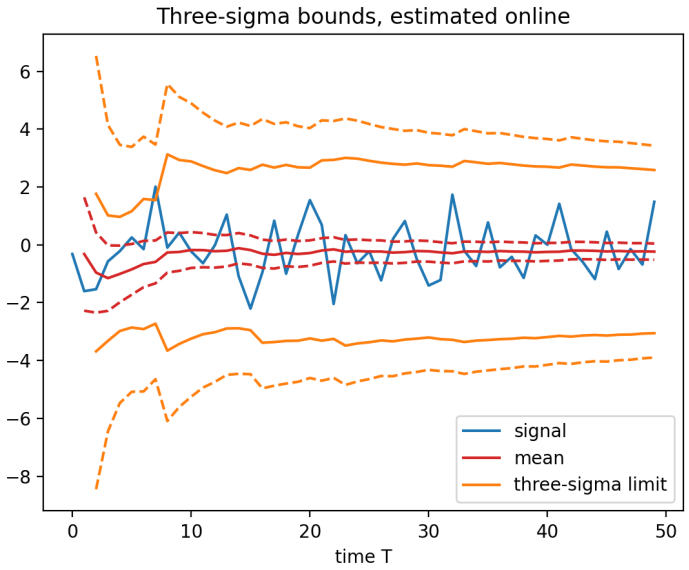


Figure 2.3.6: Three-sigma bounds estimated online using only data $t < T$ at time T. The signal point $T = 7$ lies outside the estimated three-sigma limit.

An online anomaly detection method may need to have a burn-in period to construct a good belief about what normality looks like. If we want to try to detect anomalies during this burn-in period, it makes sense to use a wider threshold to take into account

estimate uncertainty.

In practice, if anomalies are assumed to be rare, it is often the case that the anomaly detection method discounts the burn-in period entirely and only begins to look for anomalies after sufficient data is available to estimate parameters well.

2.3.3 Expanding thresholds to control false positives

The three-sigma paradigm essentially considers each point independently, which means that under the null hypothesis of no anomalies present in the dataset, the number of anomalies found by the method will be proportional to the size of the dataset. In a dataset of a million points, a one-in-a-million event would not be that unusual, and whether it should be classified as "anomalous" will depend on why you are looking for anomalies. This question is related to family-wise or experiment-wise error rate (Ryan 1959), the idea that you should control the overall probability of a false positive across a family of hypothesis tests. Therefore, you may wish to choose your thresholds for anomaly detection in a way that is dependent on your sample size.

We might consider the distribution of the most extreme point in the sample. For example, in a sample of size T where the X_t are independent and identically distributed, we have that

$$\mathbb{P}\left[\max_{t\leq T} X_t < B\right] = \mathbb{P}\left[X_1 < B\right]^T.$$

We can therefore adjust our threshold accordingly (Šidák 1967) and use a higher threshold for a larger sample size, although this threshold would still be flat. However, in the online setting we do not know the total number of hypothesis tests we will be performing, and if we send $T \to \infty$ then we would use an infinite threshold everywhere and never detect anomalies.

One way to deal with this is to have a threshold that expands as you scan over the sample. For example, instead of considering the distribution of the most extreme point in the whole sample, we may consider the distribution of the most extreme point in the

sample so far. In the case of known mean and variance, we can use this to generate a set of thresholds as in Figure 2.3.7.

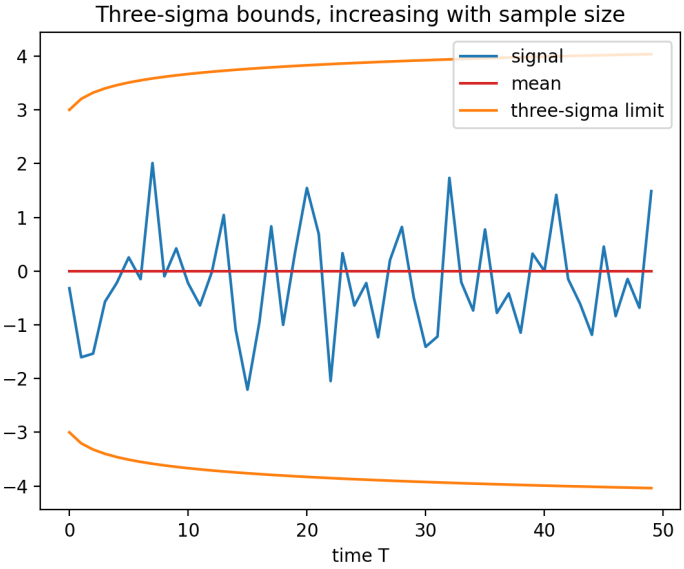


Figure 2.3.7: The threshold for an anomaly expanding with our sample size.

For the case where the mean and variance are unknown and must be estimated, we may instead use Grubbs' test (Grubbs 1950) for a single anomaly, and its generalisation the Extreme Studentised Deviate (ESD) test (Rosner 1983) for multiple anomalies, as the anomalies themselves can bias our estimates. Grubbs' test removes the most extreme point from the sample and then uses the rest of the sample to test whether the point was anomalous by comparing it to a Normally distributed sample with its most extreme point removed. The ESD test does similar for up to the k most extreme points. These methods were developed for the offline setting, but can be adapted for the online setting with fast sequential updates (Ryan, Parnell, and Mahoney 2019).

Online algorithms making on-the-spot decisions using ESD methods will, by design, have thresholds that expand as the sample size grows. This may well be an undesirable quality to have in a detection algorithm that is running on an online time series. However, methods using expanding thresholds have been developed for real applications working with large amounts of data, for example by Twitter (Hochenbaum, Vallis, and Kejariwal 2017). They have the advantage of needing fewer user-set parameters when

working online as a user can aim to control the probability of a false positive but does not need to set the assumed sample size.

Most time series applications get around this by specifying a different metric to choose the thresholds rather than the probability of a false positive (see Section 2.6.2 for a discussion). This means the threshold choices for the anomaly detection method are less sensitive to the amount of data collected.

2.3.4 Robust three-sigma

We expect that our signal will contain outliers. These outliers can contaminate our estimators for μ and σ^2. Specifically, μ will be biased in the direction of the outliers, and σ^2 will be artificially inflated. These distortions can affect our anomaly detection method in two ways:

1. A large anomaly may obscure the presence of smaller anomalies.
2. Multiple anomalies in the same direction (e.g., a collective anomaly) may completely mask themselves.

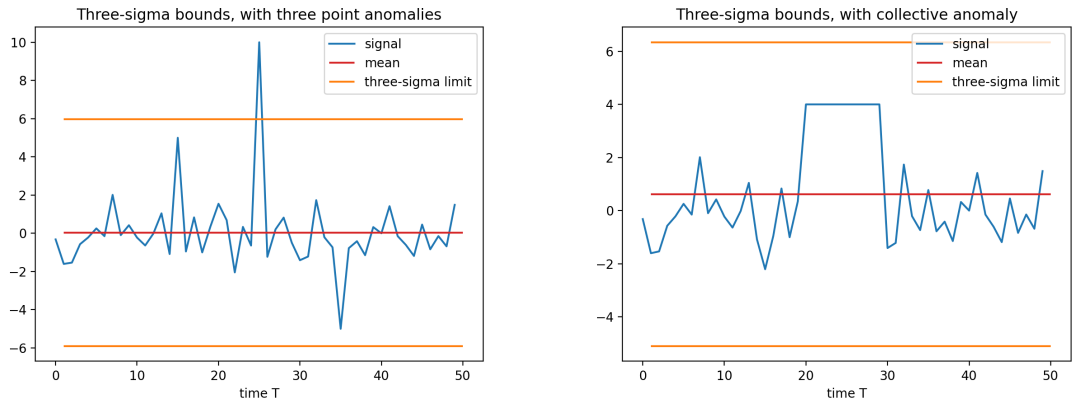


Figure 2.3.8: Point anomalies masking each other, and a collective anomaly masking itself, when bounds are estimated in a non-robust way.

Figure 2.3.8 displays graphs of an N(0,1) white noise signal of length 50 containing outliers. The dashed lines illustrate the application of the three-sigma rule using non-robust estimates, which are inflated by the presence of outliers in the signal. While one

point outlier extreme enough to skew the estimates will usually be detected, that outlier may mask the presence of other, less extreme point outliers. A collective anomaly may inflate estimates by so much as to mask itself. To address this issue, we need robust estimates for μ and σ^2 that are less affected by outliers.

We define the *breakdown point* of an estimator $\hat{\theta}$ to be the smallest fraction of anomalies in the dataset is it possible to have before the difference $|\hat{\theta}_{anom} - \hat{\theta}|$ between an anomaly-free estimate and the anomaly-contaminated estimate can become arbitrarily large (Huber 2004).

The breakdown point of the sample mean is 0. To see why this is, let's say that observation k is an anomaly.

$$\frac{1}{T}\sum_{t=1}^{T} x_t = \frac{1}{T}\sum_{t \neq k} x_t + \frac{x_k}{T}$$

Since the anomalous observation x_k could be anything, even arbitrarily large, the estimator itself could also become arbitrarily large. In a similar way, our sample variance estimator also has a breakdown point of 0. In general, any estimator which has $\sum_{t=1}^{T} X_t$ as a part of its minimal sufficient statistic will have a breakdown point of 0, as one anomalous data point can arbitrarily skew the entire sum.

Breakdown points are not the only way to quantify the robustness of an estimator. Other ways of quantifying and examining robustness also exist in the literature, including notions of *influence function* (how sensitive the estimate is to finite changes in any one data point), and *bias curves* (a visual plot of how the estimator begins to break down as more and more anomalous points are added to the dataset). (Rousseeuw and Hubert 2018)

For our purposes of anomaly detection, an estimator's breakdown point gives us a useful summary of its robust nature. This is because when working with streaming data we often apply estimators to small parts of a data signal, and we may expect point anomalies, where present, to be one or more of the points we are attempting to process.

One such possible set of robust estimators is the median m, and the median absolute

deviation (MAD), defined as the median of the values $|x_t - m|$ (Hampel 1974). Both of these estimators have a breakdown point of 50%, as up to half the sample can be anomalous. We can estimate the mean by the median, and the standard deviation by about 1.5 times the MAD (Leys et al. 2013).

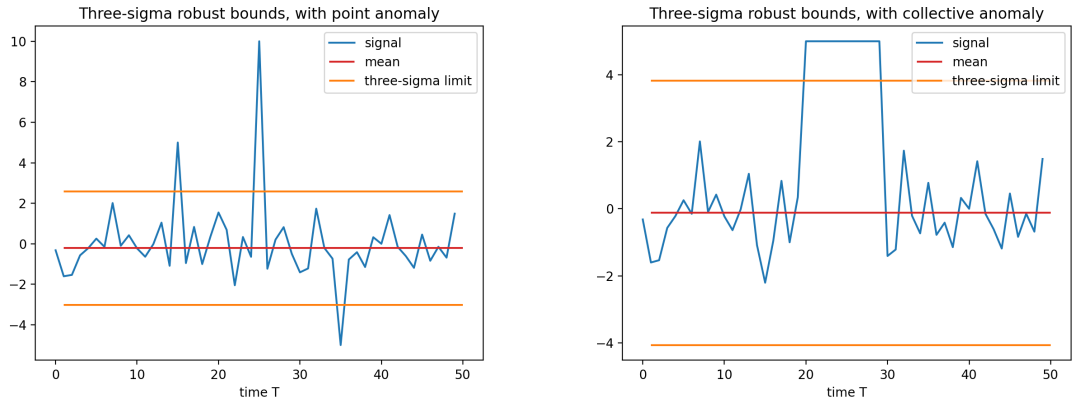


Figure 2.3.9: Graph showing three-sigma bounds given by the median plus or minus about 4.5 times the MAD. Point anomalies and collective anomalies are detected when bounds are estimated robustly.

Figure 2.3.9 shows the impact of using these robust estimates for mean and standard deviation on our anomaly detection method. Here, the same anomalies as in Figure 2.3.8 are now detectable at a three-sigma level, because the anomalies themselves do not greatly inflate the thresholds used to detect them.

A collective anomaly can consist of many points. Even when a robust method is used as in Figure 2.3.9, and the size of the collective anomaly is below the method's breakdown point, the presence of so many outlying points can cause misestimations. This means that in order to detect collective anomalies well, we should be considering them as intervals rather than individual points.

2.3.5 Detecting collective anomalies with three-sigma

An interval may certainly be said to be a collective anomaly if all of the points in that interval are themselves point anomalies. However, we are often interested in detecting anomalous intervals where not all of the points in the interval are point anomalies, or

even where no points in the interval are point anomalies.

Detecting collective anomalies requires methods different from detecting point anomalies, but these methods can sometimes coincide if you consider a point anomaly to be a collective anomaly of length one.

Consider the generalisation of the three-sigma framework above. We have an independent and identically distributed Normal signal with known mean $\mu = 0$ and variance $\sigma^2 = 1$, and we wish to label as anomalous not just any point outside the three-sigma boundaries, but any interval whose mean is outside the three-sigma boundaries when performing the appropriately scaled statistical test.

If the X_t are i.i.d. $N(0, 1)$, we would expect the mean $\bar{X}_{t+1:t+h}$ of an interval of length h to be distributed as $N(0, 1/h)$. Therefore, the appropriately scaled test is to check if $\bar{x}_{t+1:t+h}\sqrt{h}$ is within the three-sigma boundaries. When $h = 1$, this reduces to the point anomaly test. Figure 2.3.10 shows a visualisation of how this length-intensity tradeoff looks.

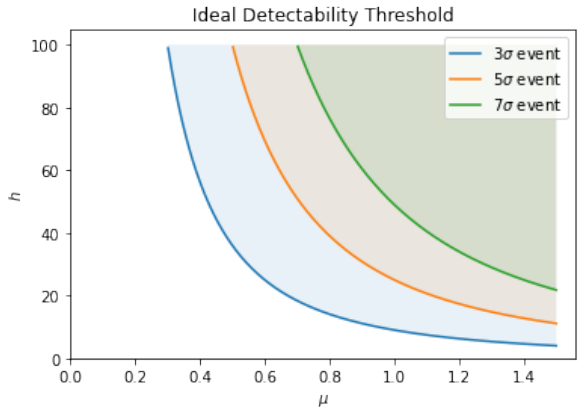


Figure 2.3.10: The tradeoff between intensity μ and length h of an anomaly.

2.3.6 Interval search

When we search for collective anomalies, we end up with a natural increase in the computational complexity of our problem that particularly works against us in the online setting. This arises because the natural units of collective anomalies are intervals, not points themselves. Intervals have start points and end points, and in a signal of

length T there are $\binom{T}{2} = O(T^2)$ intervals. Checking all these intervals can require very large amounts of computation when working with high volumes of data.

This also contrasts with our requirement for an algorithm to be able to run in the online setting: its computational complexity must be $O(1)$ at each timestep so that it does not slow down as the data stream lengthens. The last point X_T to arrive in a data stream of current length T creates a set $\{[X_1, X_T], [X_2, X_T], ..., [X_{T-1}, X_T], [X_T, X_T]\}$ of new intervals. Doing anything that loops over each element in this set will by necessity result in an algorithm with computational complexity at least $O(T)$ at each timestep. This will cause the method to slow down on long signals.

When finding collective anomalies, we are often only interested in intervals up to a maximum anomaly length $h_{\max}$. If $h_{\max}$ is not too large, this can be reasonable. However when working on real-time signals and checking all intervals up to a fixed maximum time interval, the computation required per second becomes proportional to the square of the signal velocity (how many points the data signal contains per second), which can become infeasible on high-velocity signals.

Consider an example where we must choose the processing velocity of a signal, and we are interested in collective anomalies up to 5 minutes (300s) long. If we process the signal once a second, we then have 300 intervals to check each second. If we process the signal twice a second, we have 600 intervals to check each half-second, for a total of 1200 intervals processed each second. If we are interested in a minimum anomaly length of approximately 100ms, we may wish to process the signal every 20ms to ensure a good resolution, which would require checking 750,000 intervals each second.

We have presented three formulations: $O(T^2)$ intervals in total, $O(T)$ intervals at each timestep, and $O(v^2)$ intervals per second for a signal velocity v even if intervals are bounded above in length. We will refer to these interchangeably as the interval search problem.

Many collective anomaly detection methods (see for examples Austin et al. (2023), Ding and Fei (2013), Ryan, Parnell, and Mahoney (2019), and Yang, Eckley, and Fearnhead (2024)) overcome the interval search problem by using a sliding window. As

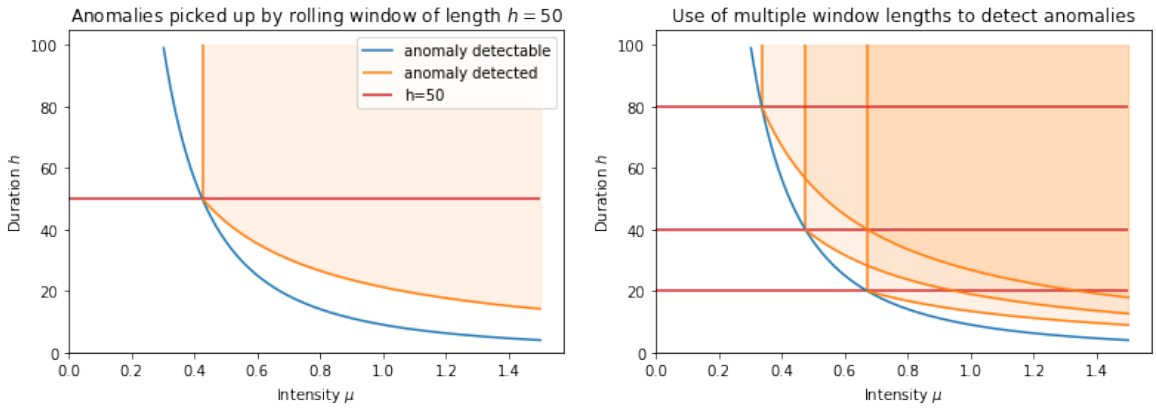


Figure 2.3.11: Use of one or more sliding windows to detect anomalies.

shown in Figure 2.3.11, this will cause a loss of detection power for collective anomalies that occur over shorter or longer lengths than the sliding window. To compensate for this, some anomaly detection methods use a grid of windows, often logarithmically spaced of size $1, 2, 4, 8, 16, ...$ or near-logarithmic at $1, 2, 5, 10, 20, ...$ in order to give round numbers.

2.3.7 Changepoints

Definition 2.3.1 (Changepoint)**.** *Given an ordered dataset* $(X_1, ..., X_T)$*, a point* τ *is a* ***changepoint*** *if the distributional properties of the two parts of the signal* $(X_1, ..., X_{\tau-1})$ *and* $(X_\tau, ..., X_T)$ *are different from each other in a statistically significant way.*

A distributional property of interest might be the mean, variance, or slope of the signal. Alternatively, the two parts may be drawn from the same distributional family using different parameters. A signal may have multiple changepoints $1 = \tau_0 < \tau_1 < ... < \tau_k = T$ *that divide the signal into* k *different parts.*

Some definitions of changepoints (see for example Killick, Fearnhead, and Eckley (2012)) will define a changepoint as the last point of the prior signal part, rather than the first point of the latter signal part as we have done. This essentially shifts all τ backwards by 1 relative to the way they are presented in this thesis. For the purposes of anomaly detection it makes sense to have the τ included in the anomaly we detect, so we adopt this convention throughout the thesis.

If we are passing over a signal containing a collective anomaly $(X_s, ..., X_e)$ and we are at the present time $T = e$ when the collective anomaly ends, then changepoint methods that detect a start time $\tau = s$ should be well-placed to pick up our collective anomaly. Although we don't know e in advance, if we are running the method online and testing all present times T, this is not a problem for us. However, there are a number of differences between the anomaly detection and changepoint detection setttings.

The biggest difference between a changepoint and a collective anomaly is the return to baseline assumption. Collective anomalies have an end, after which the signal is assumed to return to its normal behaviour. Changepoints, however, represent an underlying shift to a new type of signal. There may be another changepoint later on, but no assumption that this causes a return to the original signal pattern.

Changepoint models are usually interested in capturing good information about what a signal has moved to after a change has been made. This is because, in fitting multiple changepoints, it needs to be able to use information gained about the new state to help decide whether it should then fit more changepoints after that state move. In contrast, anomaly detection methods are really only interested in the fact that an anomaly has occurred, and not exactly what that anomaly looks like.

This conceptual difference is reflected in how changepoint fitting models are set up. One common model (Jackson et al. 2005), fits changepoints by minimising the following penalised cost function:

$$\min_{0=\tau_0<\tau_1<...<\tau_k=T} \left[\sum_{i=0}^{k-1} \text{Cost}(X_{\tau_i}, ..., X_{\tau_{i+1}-1}) + \beta(k) \right].$$

Here, $\beta(k)$ is a penalty parameter for fitting more changepoints. For computational reasons it is often linear, i.e. $\beta(k) = \beta k$ for some constant penalty β for fitting an additional changepoint (Killick, Fearnhead, and Eckley 2012), that becomes a tuneable parameter of the algorithm.

Contrast this with how the Collective And Point Anomalies (CAPA) method (Fisch, Eckley, and Fearnhead 2022) sets up its model, with a known baseline from which

Framework	Point anomalies	Collective anomalies	Changepoints
Length of anomalous segment	1	$2...h_{\max}$	2+
Naive computational complexity	$O(T)$	$O(T^2)$	$O(2^T)$
Requirement to estimate background?	Yes	Yes	No
Multiple could occur nearby in a signal?	Yes	No	Yes
Magnitude of change detected	Large (outlier)	Medium (bounded >0)	Small (no limit)
Time to detect change online	Immediate	Prompt	Eventual

Table 2.3.1: Summary of comparisons between frameworks for detecting point anomalies, collective anomalies, and changepoints in a time series signal.

anomalies deviate away:

$$\min_{0<\tau_1<...<\tau_{k-1}<T} \left[\sum_{i=1}^{k-1} \text{Cost}(X_{\tau_i}, ..., X_{\tau_i+h_i}) + \text{Cost}(\text{Non-anomalous points}) + \beta(k) \right].$$

Here, the τ_i represent anomaly start times and the h_i represent anomaly lengths. This increases the method's statistical power, as it is fitting one anomaly rather than two changepoints (start and end).

Changepoint models are also set up so that they gain statistical power as the distance h between changepoints becomes larger, tending to infinity. They are just as interested in detecting very small changes that happen over very long timescales as they are in detecting shorter, sharper changes. In contrast, when detecting collective anomalies we often have a maximum length $h_{\max}$ of anomaly that we are interested in detecting. This is because when working in anomaly detection, the slow background evolution of our signal over time is not something we want to cause our algorithm to report as anomalous. We are often interested in specifically removing or compensating for this background. Therefore, changepoint methods methods designed to perform well on very large values of h may perform poorly in anomaly detection tasks as they end up measuring background shifts rather than anomalies (see Chapter 3 for a discussion on this).

Finally, there is often a substantial difference between the naive computational complexity of a multiple changepoint detection model and a collective anomaly detection model which only looks for the presence of two changepoints (the start and end of the collective anomaly). When fitting multiple changepoints on a signal of length T there

are 2^T possible combinations - any point could be a changepoint. In contrast, when fitting a collective anomaly there are $T^2/2$ locations for the start and end.

In order to take advantage of this on real data which may contain multiple collective anomalies, careful considerations must be given to using the return to baseline assumption to reset the anomaly detection algorithm between passing over anomalies, something which is impossible when using a changepoint model. These are discussed more in Chapter 4.

These comparisons between point anomaly detection, collective anomaly detection, and changepoint detection are summed up in Table 2.3.1.

2.3.8 Collective anomaly detection methods

The most basic collective anomaly detection method is the window method, as described in Figure 2.3.11, which relies on knowing the length of the collective anomaly in advance but is able to detect all possible intensities of anomaly. One other way to detect collective anomalies is the CUSUM control chart (Aue and Kirch 2024), based upon the use of the Page-CUSUM statistic (Page 1954). Here, we assume we know the intensity of the anomaly in advance, and are therefore able to detect all possible lengths of anomaly.

The Page-CUSUM statistic is calculated as

$$S_0 = [aX_0]^+, \quad S_{T+1} = [S_T + aX_{T+1} - b]^+$$

for some positive constants a and b, where we denote by $[]^+$ the greater of the term within the brackets and zero. The choice of a and b determines the statistical model we are fitting to the data. For example, $a = 1/\sigma$ and $b = \mu/\sigma$ performs a likelihood ratio test of $N(0, \sigma^2)$ against $N(\mu, \sigma^2)$ (Page 1955). The choice of $a = \mu \log(\mu)$ and $b = \lambda(\mu - 1)$ performs a likelihood ratio test of Poisson(λ) against Poisson($\mu\lambda$) (Lucas 1985). Due to rescaling, only the ratio a/b affects the way data is stored and collected by this statistic.

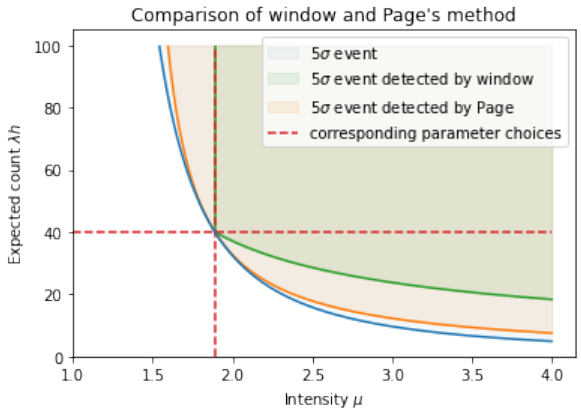


Figure 2.3.12: Equivalences between anomaly length and anomaly intensity for Poisson distributed data.

For a given statistical significance threshold, there is a one-to-one equivalence between window tests optimised for detecting a specific length of anomaly and Page-CUSUM tests optimised for detecting a specific intensity of anomaly. Specifically, a length pairs with the intensity at which it would be exactly detectable at the given significance threshold. Page-CUSUM tests at this intensity will detect all anomalies that the specific window would detect, while not detecting any anomalies that some window length would not. An example for Poisson distributed data is given by Figure 2.3.12, showing the output of a log-likelihood ratio test using Wilks' Theorem (Wilks 1938).

For Normally distributed data, our likelihood ratio statistic can be computed exactly as $\mu^2 h$ for an anomaly of mean μ and length h, by treating the mean over the interval as a $N(0, 1/h)$ random variable. In both these cases our statistical threshold is k^2 for a k-sigma event: a statistical significance of a single point at least k standard deviations from its mean. Our three-sigma paradigm would choose $k = 3$. This means that a Page-CUSUM statistic optimised to detect anomalies of length h in Normally distributed data should choose $\mu = \frac{k}{\sqrt{h}}$. For data that is not Normally distributed, a numerical root finder can be required to compute the equivalence.

Page-CUSUM statistics have three main advantages over window methods:

1. If the anomaly is of a sufficient intensity where it provides evidence in favour of

the likelihood ratio test (for example, of mean greater than $\mu/2$ in the Normally distributed example above), then this evidence will continue to collect over a time horizon for as long as the anomaly lasts, and it will eventually become detectable at any given significance threshold.

2. A Page-CUSUM test requires much less memory usage than a window test. This is because all data points in the window must be stored individually in the window algorithm's memory, as a point must be removed from the window each time the window advances. Page-CUSUM tests never remove any points and therefore do not need to store points individually in memory.

3. A Page-CUSUM test does not require expensive computations (such as computing logarithms) in order to perform its likelihood ratio test approximation. The values of a and b are fixed and can be stored in memory, so the only computations required are sums and products.

The Functional Online Cumulative Sum (FOCuS) method (Romano, Eckley, Fearnhead, and Rigaill 2023), which this thesis is primarily about, is a generalisation of the Page-CUSUM test to all possible values for the ratio a/b simultaneously. Section 5.2.2 provides a detailed description of how FOCuS works for Gaussian data, before generalising to a variety of one-parameter exponential family models.

The main disadvantage of Page-CUSUM tests over window tests with a maximum window size is that Page-CUSUM tests are not robust to passing over large collective anomalies in the signal. Such anomalies cause an overshadowing effect, where the most statistically significant interval ending at the present time will contain this anomaly for a very long time afterwards. To avoid highlighting this weakness they are often considered as methods that stop and return a detection when a significance threshold is passed (Aue and Kirch 2024). However, in practice anomaly detection methods are often run with infinite significance thresholds to generate a significance trace for further inspection (e.g. plotting on a graph). Here, the presence of a large earlier anomaly can hinder the detection of smaller, later anomalies. In contrast to this, window methods

have a greater degree of robustness to passing over large collective anomalies. Once a collective anomaly is far enough in the past as to not be contained in the window, it no longer affects any computations from the points in that window. Chapter 4 explores and addresses this weakness in the Page-CUSUM test and therefore in the FOCuS algorithm through the lens of a nuclear security monitoring application.

We have used the term "Page-CUSUM" in order to avoid confusion between the CUSUM control chart (which uses the Page-CUSUM statistic as defined above) and a different CUSUM statistic which is used to compute a likelihood ratio test while simultaneously estimating a fixed unknown pre-change parameter (see for example Yu et al. (2023)). We address the way in which FOCuS can be used to sequentially calculate this alternative CUSUM statistic in Chapter 5. In contrast, the Page-CUSUM statistic assumes a known pre-change parameter and measures deviations from that parameter. This is often more useful in practical anomaly detection applications because the baseline signal is itself evolving.

2.4 Dealing with data shapes and evolving baselines

When a new anomaly detection method is designed or developed, it is often tested assuming the background signal is distributed independently and identically, when in real-world applications this is usually not the case. The structure in a signal can take many different forms, but is roughly divisible into global effects such as an overall trend, and local effects such as autocorrelation between nearby signal points, anomalies themselves, and random noise:

$$\text{signal} = \text{global effects} + \text{local effects} + \text{anomalies} + \text{noise}.$$

The above equation implies that these structural factors are additive and can be handled somewhat independently of each other, but in practice they often influence each other in non-additive ways. Methods to estimate and account for this additional structure in a signal will mean that an anomaly detection method can be tuned to pick up actual

anomalies, rather than false positives which represent known signal structure.

2.4.1 Autocorrelation

Autocorrelation in a signal is when neighbouring points of the signal are correlated, usually with a positive correlation. This means that when a signal is high by chance, nearby points are also more likely to be high without this indicating the presence of an underlying anomaly in the signal.

Autocorrelation is common in signals that consist of discrete measurements X_t of an underlying continuous process $X(t)$ where there isn't much noise associated with taking a measurement. In this case, the shorter the time interval between measurements, the higher the autocorrelation is expected to be, as the underlying signal does not change much in the meantime.

In contrast to this, autocorrelation is less of a problem in signals where X_t represents successive measurements of a count data process $X(t)$, representing the number of counts in small intervals of time. If the underlying data process is not self-exciting, then no autocorrelation will be present.

To model autocorrelation, we can use autoregressive and moving average models. An autoregressive model of length p is a linear model where the forecast X_{t+1} is based on a linear combination of the previous p values, plus a constant c chosen to achieve the desired mean, plus a Gaussian white noise error term:

$$X_{T+1} = \sum_{i=T-h}^{T} a_i X_i + c + \epsilon_{T+1}.$$

A simple example is the autoregressive model of length 1, also called the $AR(1)$ model:

$$X_0 = \epsilon_0, \quad X_{T+1} = aX_T + \sqrt{(1-a^2)}\epsilon_{T+1}.$$

Here, $0 \leq a < 1$ is a parameter that determines how much the series depends on its previous values. $a = 0$ gives an entirely uncorrelated series that is suitable to model as independently identically distributed data.If the ϵ_t are all independently $N(0, 1)$, We

have that each $X_t \sim N(0, 1)$ individually.

Autocorrelation and collective anomalies

If a is close to 1, then nearby points in the series are likely to be close to each other. This means that the mean of the series on intervals, particularly on short intervals, is going to have higher variance than it would be for independent data. For independent $N(0, 1)$ data we should expect the variance of an interval of size h to be $1/h$, but in the presence of autocorrelation this is elevated.

We often define a collective anomaly to be an interval when the mean of the series is significantly above (or below) its usual value. The statistical significance of an interval will depend on the distribution of the interval means under the null hypothesis. Autocorrelation will affect these distributions, generally requiring the anomaly detection thresholds to be elevated in order to achieve the same false positive rate. For instance, the CAPA algorithm (Fisch, Eckley, and Fearnhead 2022) recommends inflating the penalty for fitting an anomaly by $\frac{1+\rho}{1-\rho}$ in order to work with first-order autocorrelation ρ.

2.4.2 Trend

If the signal exhibits a time-evolving trend, this must be accounted for when detecting anomalies. There are two main methods for doing this: signal differencing, which is appropriate for more obvious point anomalies, and trend estimation, which is necessary for less obvious or collective anomalies.

Signal differencing

When searching for point anomalies in a signal X_t, an easy method to remove the trend is to work with the differenced signal $X_t^{'} := X_t - X_{t-1}$. This approach can make the time series stationary, particularly when observations are frequent, and the incremental change due to the trend is minimal.

A point anomaly X of size $\Delta = X - \mu$ in a noisy signal with mean μ and standard deviation σ has a signal-to-noise ratio of Δ/σ. This ratio is critical for distinguishing anomalies from the background noise, essentially reflecting the three-sigma rule.

In a signal with independently distributed noise of standard deviation σ and a point anomaly of size Δ, differencing once increases the noise to $\sqrt{2}\sigma$ while maintaining the anomaly size Δ. This reduces the signal-to-noise ratio to $\sqrt{\frac{1}{2}}\frac{\Delta}{\sigma}$, thereby diminishing the ability to detect anomalies. However, in a differenced signal, a point anomaly appears as a jump of Δ followed by a return to baseline $-\Delta$. By double-differencing the signal, $X_t'' := X_t' - X_{t+1}' = -X_{t+1} + 2X_t - X_{t-2}$, we regain some statistical power. Here, the noise becomes $\sqrt{6}\sigma$ and the anomaly size becomes 2Δ, leading to a signal-to-noise ratio of $\sqrt{\frac{2}{3}}\frac{\Delta}{\sigma}$. Point anomalies will also introduce artefacts in the surrounding double-differenced points.

Signal differencing is an effective preprocessing technique for detecting contextual anomalies, where only the immediate neighboring points are relevant. Signal differencing plus an autoregressive and moving average model is the basis of the popular ARIMA method for forecasting time series. However, collective anomalies will not be detectable in a differenced signal, because only the beginning and end of the anomaly will show in the differenced signal.

Signal differencing is appropriate under the following conditions:

1. There are no seasonal effects, so it is valid to assume that non-anomalous neighboring points should be approximately equal apart from noise.

2. The focus is on contextual outliers that can be clearly separated from noise, so a reduction in detection ability by a factor of $\sqrt{3/2}$ is acceptable.

3. You are looking for point anomalies rather than collective anomalies.

Trend estimation

Trend estimation is a statistical method that we can use to identify and measure patterns or directions in data over time. It helps us distinguish long-term movements

from short-term fluctuations, and it's necessary for finding collective anomalies and less obvious point anomalies.

We often use two methods for trend estimation: a moving average and exponential smoothing. Although we frequently use linear regression in other contexts, it is less suitable for time series data. This is because linear regression is highly sensitive to outliers, especially near the beginning or end of the data series. These outliers can have a strong influence on the results.

The simple moving average calculates the mean of a rolling window with a fixed number of consecutive data points. We can use a more robust version, which takes the median of the points in the window instead of the mean. If we want to balance these two approaches, we can use a trimmed mean. In this method, we remove a fixed percentage of the largest and smallest values before calculating the mean of the remainder. For instance, if our data contains well-spaced point anomalies, we might remove the maximum and minimum values before calculating the mean. However, if we encounter collective anomalies, this method may not be suitable, and we would need to trim additional points. All these methods treat the data points in the window equally, without considering their order.

Exponential smoothing (Brown and Meyer 1961) is another method that estimates trends by applying decreasing weights to older data points. We give more importance to recent observations. This method uses a smoothing factor α, which is similar to a window size parameter. The smoothing factor α ranges between 0 and 1, and is usually close to 0. A higher α gives more weight to recent data, allowing the estimate to respond quickly to changes. A lower α smooths the data more heavily. We update the evolving estimate E_T for the trend at time T using the formula

$$E_{T+1} = \alpha x_{T+1} + (1 - \alpha) E_T.$$

Exponential smoothing requires less memory than moving average methods. We only need to keep one value at each time step. In contrast, moving average methods

require us to store all points in the window so we can remove the oldest point as the window advances.

However, exponential smoothing is a biased method. Since we only use data up to time T to estimate the trend at time T, the estimates will lag behind the actual signal. To remove this bias, we can use double exponential smoothing, where we once again use exponential smoothing to estimate this lag and then correct for it. While this appropriate for trends that are roughly linear, it can cause additional problems when the signal is turning.

Once we've estimated the trend of a signal, we usually subtract the trend from the signal to leave a residual that we hope contains our anomalies. We can then run other anomaly detection methods on the residual.

2.4.3 Artefacts in data

Whenever estimates of trend are removed from a signal, they can cause artefacts to appear in the data residual. These can take various forms. We will concentrate on three types of artefact: negative autocorrelation, noise pulses, and artefacts around an anomaly.

Negative autocorrelation

Removing trend from a signal can introduce negative autocorrelation in the signal residual. This happens both with signal differencing, and also with trend estimation and subtraction. For the differenced signal this is easy to show, as we have that if the X_T are independent then $X_{T+1} - X_T$ is negatively correlated with $X_T - X_{T-1}$ due to the differing signs of the X_T.

Rolling trend estimation methods using a window will introduce negative autocorrelation at all lags up to half the window size used for the estimation, as this is the spacing at which two points are used for the same trend estimate. This can be seen in Figure 2.4.13. In this figure we see that increasing the window size decreases the magnitude of the negative autocorrelation introduced to the signal, by comparing the

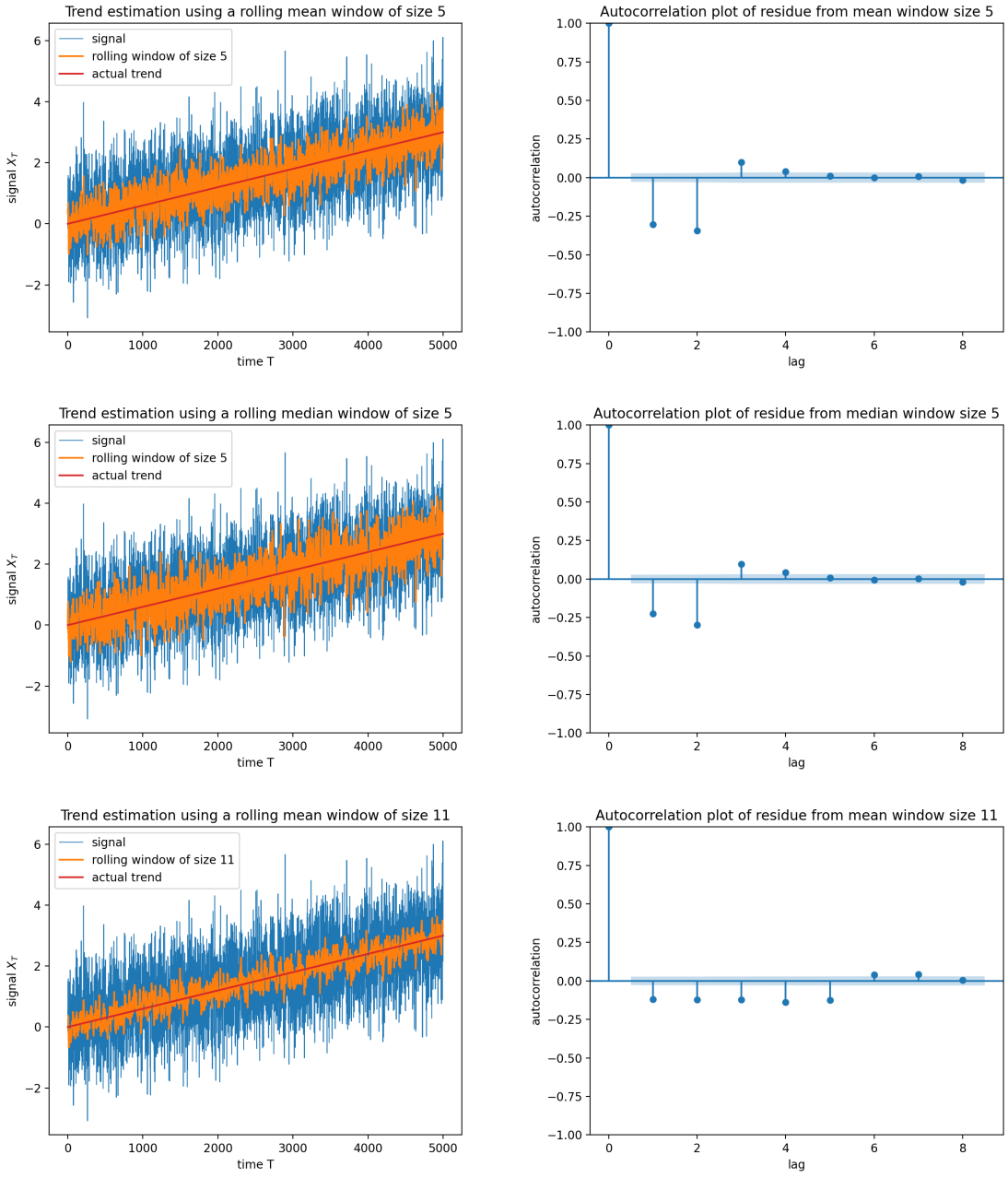


Figure 2.4.13: Autocorrelation plots showing the negative autocorrelation in the signal residual introduced by rolling trend estimation, and how it is affected by choice of estimation method or different window sizes.

first row to the second. However, looking at the third row we see that using a robust estimation method (as described in Section 2.3.4) does not reduce the autocorrelation introduced.

This is particularly important for anomaly detection as many statistical methods

for detecting anomalies require a threshold adjustment to account for autocorrelation, particularly for collective anomalies (see Section 2.4.1 for a discussion).

Noise pulses

Often, there is more noise in data taking higher values, so subtracting off a trend will lead to a constant signal containing noise pulses. For instance, daily noise fluctuations in the value of a stock may be expected to be proportional to the value of that stock. Another example is data that represents a background arrival process, where noise is associated to randomness in the number of arrivals within any given time period. This could be modelled as a Poisson process with varying rate parameter $\lambda(t)$. If the model is correct the standard deviation of the arrivals should be $\sqrt{\lambda(t)}$, so subtracting off the trend could lead to a signal where the variability is no longer accurately represented (see Figure 2.4.14).

In order to process these noise pulses, you could calculate a rolling estimate of the standard deviation of the de-trended statistic and then divide by it. However, this then introduces more estimation error.

In the Poisson case, you might wish to apply the variance-stabilising transformation $x \to \sqrt{x}$ to your raw data before estimating and subtracting your trend. This can be improved on by using the Anscombe transform $x \to 2\sqrt{x + 3/8}$ (Anscombe 1948), which further stabilises the variance for small count sizes. The effects of doing this can be seen in Figure 2.4.14. In the proportional to value case, you may wish to divide by your trend estimate rather than subtract it. These methods will avoid noise pulses in the residual if those data models are correctly specified.

Artefacts around anomalies

A point anomaly in a singly differenced signal will show up as two points - one unusually high, the other unusually low, as shown in Figure 2.4.15. In a doubly differenced signal, the main anomaly will be surrounded by two smaller anomalies of opposite signs.

When using non-robust trend estimation, large point outliers can also skew the

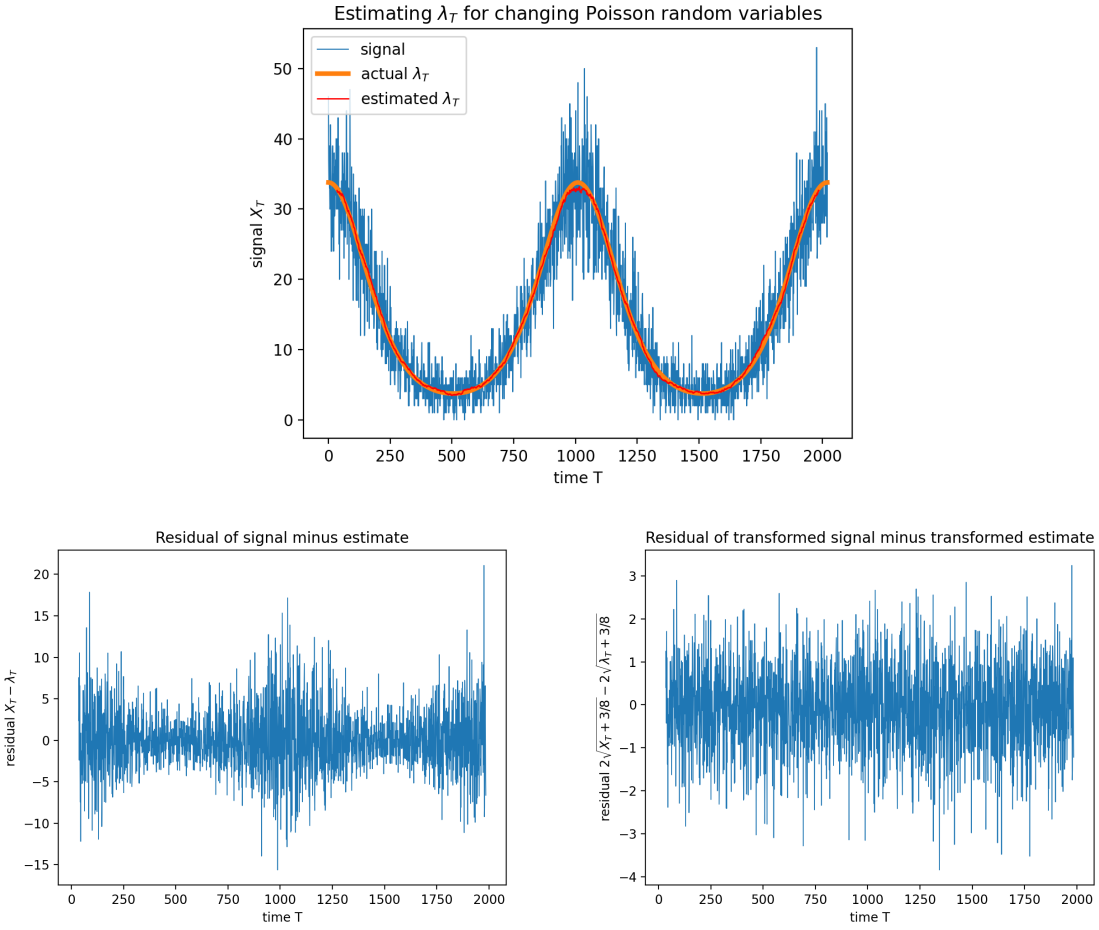


Figure 2.4.14: Poisson random variables with λ_T distributed as a log-sine wave, which is then estimated well using a centred moving mean. When subtracting off the estimate from the signal, we are left with clear noise pulses in our signal residual. When using an Anscombe transform we are not left with noise pulses.

trend estimate. For example, Figure 2.4.15 shows how the use of a rolling mean causes a negative lag in the signal residual. This may be important for avoiding false detections of non-anomalous points that lie nearby large anomalies.

The use of robust methods for calculating trend, such as those given in Section 2.3.4, can be applied to reduce the effect of artefacts around anomalies. In cases where we expect large point ouliers to be present, these methods may be preferred. However, in some cases this must be balanced against the additional computational complexity of computing a rolling robust statistic (for example median) versus a rolling mean (Juhola, Katajainen, and Raita 1991).

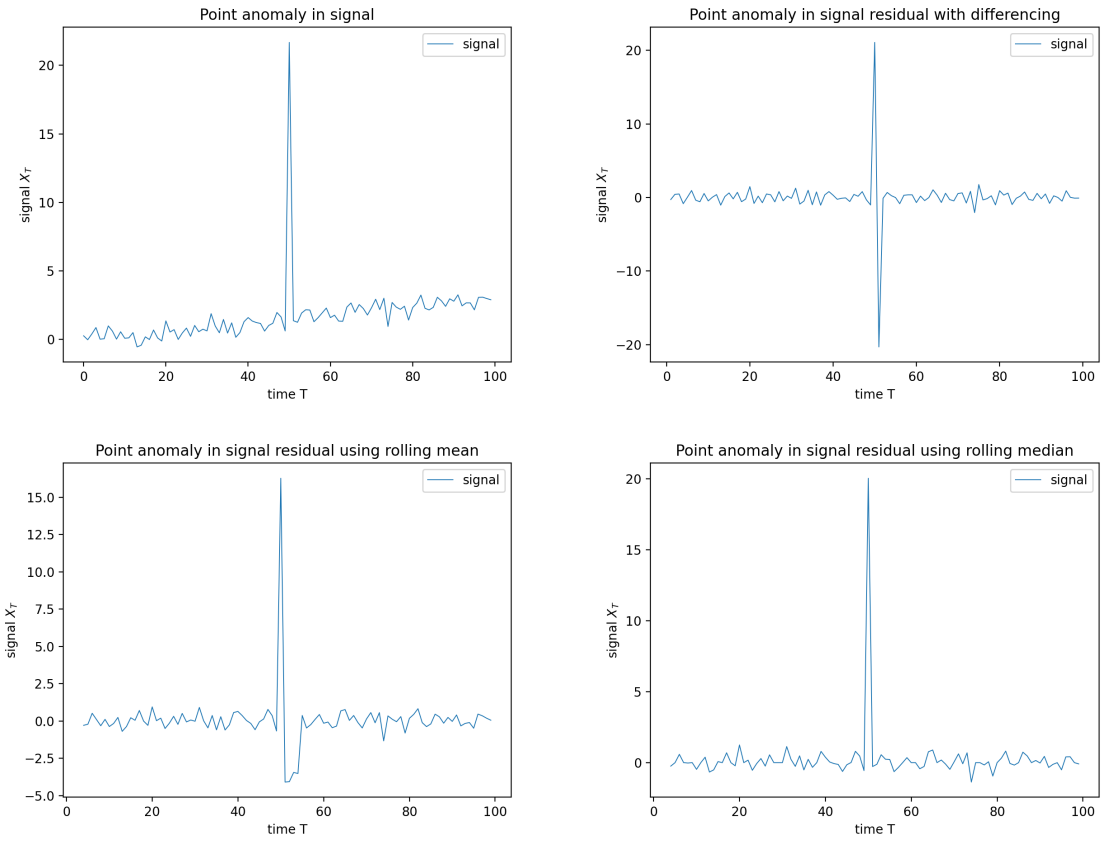


Figure 2.4.15: A signal with a rising trend and a large point outlier (top left). The use of signal differencing (top right) or non-robust trend estimation (bottom left) can lead to artefacts present in the data residual around an anomaly. Robust trend estimation (bottom right) avoids these artefacts.

2.5 Multivariate anomaly detection

There are a large number of methods used to find anomalies in multivariate data. Here, we survey some of those methods, with a particular focus on how well they scale and the methods they use to do so.

2.5.1 Stationary anomaly detectors

The simplest way of scaling an anomaly detector to use on a large dataset is by constructing a stationary anomaly detector from a sufficiently small sample.

Definition 2.5.1 (Stationary Anomaly Detector)**.** *Given a training dataset* $\{X_1, ..., X_n\}$ *and a live, or test, dataset* $\{Y_1, ..., Y_m\}$*, a stationary anomaly detector is a function* F

constructed from only the training set, such that $F(Y_k) \in \{0, 1\}$ *tells us whether or not* Y_k *is, or contains, an anomaly.*

The problem of constructing a stationary anomaly detector is essentially the same as constructing a binary classifier, and has been well studied in the machine learning literature.

The points X_k could be labelled with whether they are anomalous or not, making this a supervised problem if enough anomalous points are included in the training set. More likely it is the case that the training set does not contain enough labelled examples of every type of anomaly we would want F to detect, and we would instead approach the problem from a semi-supervised or unsupervised perspective. In a semi-supervised paradigm, we would have a training set of only non-anomalous points, and construct F by learning only the definition of "normal", treating everything that doesn't fit this as an anomaly. In an unsupervised paradigm, our training set may itself be contaminated by unlabelled anomalies, and our construction of F must be robust to this.

Stationary anomaly detectors are used a lot in online anomaly detection problems. This is because it doesn't matter how much time is taken to construct F (which happens offline), only the time taken to calculate each $F(Y_t)$ as it arrives. However, stationary anomaly detectors do not handle data with a temporal structure well. This is referred to as concept drift, where the behaviour of the live dataset changes over time until it no longer resembles the training data that was used to construct F, causing increasing numbers of false positives. Sometimes this problem is dealt with by reconstructing F on a block of new data once concept drift has been detected. While this requires the construction of F to be sufficiently fast that it can take place online, if the concept drift is slow then reconstructing F only once per block rather than once per point can give significant computational savings.

Examples of methods used as a stationary anomaly detector include support vector machines, clustering, isolation forest, neural networks, as well as simpler methods such as the three sigma method.

2.5.2 Dimension reduction

For datasets that are very high-dimensional, dimension reduction techniques can be used to pre-process the dataset before applying anomaly detection methods. This is because many multivariate inlier detection methods do not scale well with dimension, and are only suitable for using on datasets with a small number of dimensions.

If we already have a set of labelled anomalies, then we can use dimension reduction techniques such as discriminant analysis. This is a supervised method aiming to find a small set of dimensions that maximise the differences between two classes of data points. Linear discriminant analysis (Fisher 1936; Mardia 2024) works by finding the eigenvectors of a matrix designed to maximise the ratio of between-group scatter to within group scatter. However, its linear form assumes that both anomalous and non-anomalous data are normally distributed, and that there are a large enough number of anomalous data samples compared to the number of input dimensions. Generalisations of this paradigm include various kinds of kernel discriminant analysis for nonlinear data patterns (Roth and Steinhage 1999; Baudat and Anouar 2000; Mika et al. 1999), and regularised discriminant analysis useful for small samples with high dimensions (Friedman 1989). Discriminant analysis computations can usually be performed quite efficiently even in high dimensions (Cai, He, and Han 2011).

If we do not have any labelled anomalous data, we can instead use principle component analysis (Jolliffe and Cadima 2016). This calculates the eigenvalues and eigenvectors of the covariance matrix for the whole dataset rather than dividing it into classes. The eigenvectors determine the principal components, and the eigenvalues determine their magnitude, indicating the amount of variance captured by each principal component. The first few principle components capture most of the variability in the dataset and we can choose them as our dimensions of interest. However, since anomalies are rare, we have no guarantee that they capture the variability relevant to separating the anomalies, and we may need to use more components than we would want to for a classification problem with more equally balanced classes.

When using dimension reduction techniques such as discriminant analysis or principle component analysis, the output coordinates can be difficult to interpret, as they are linear or even nonlinear combinations of the original features. This can obscure the underlying meaning of the data. Additionally, both kind of methods are sensitive to the scaling of the data, requiring coordinate normalization or standardization beforehand to avoid coordinates with large ranges with no underlying meaning to them biasing the method.

2.5.3 Types of anomaly detection methods for multivariate data

When working with multivariate data, any anomaly that is an outlier in at least one coordinate can be found by simply running a univariate anomaly detection method over each coordinate individually. It is also possible to simply combine information across coordinates, for example by adding up anomaly scores from each coordinate to give a total anomaly score for that point. This may help identify outliers that occur in several coordinates.

However, anomalies in multivariate data may not be outliers. While detecting outliers is fairly easy, detecting inliers requires careful consideration of the dynamics of the whole of the dataset. The challenge of anomaly detection becomes a challenge of learning the normal structure of the data, where anomalies are classed as points that do not follow this structure.

Many different methods of detecting multivariate inliers exist within the literature, a few of which are covered in this section. These methods often face the primary challenge of maintaining a good computational complexity on large datasets.

When considering the computational complexity of a multivariate anomaly detection method, its scaling is measured with both

1. the number of points n in the dataset, and
2. the number of dimensions, or coordinates, p of each point.

An anomaly detection method is usually considered too slow for practical use on a large

sample if it is $O(n^2)$, representing a need to compare each point in the dataset to every other point in the dataset. Methods that aim to be faster than this often suffer a curse of dimensionality problem, leading to algorithms that are $O(\alpha^p)$ for some $\alpha > 1$ and therefore infeasible for use on high-dimensionality datasets.

We will consider two main ways to constrain the computational cost of an algorithm to below $O(n^2)$: neighbour-based methods, and ensemble methods.

Neighbourhood-based methods

Neighbourhood-based methods consider whether a point X is anomalous by operating only on the points nearby to X. The two main ways of defining a neighbourhood are:

1. The neighbourhood of X is X and its k nearest neighbours.
2. Draw a fixed ball of radius ϵ around X. The neighbourhood of X is the points within that ball.

These both have the advantage of being relatively easy to calculate, using data structures such as kd-trees (Bentley 1975) for lower-dimensional spaces or ball trees (Omohundro 1989) for high-dimensional ones, to compute neighbourhoods for all points in the dataset more efficiently than computing each neighbourhood individually.

Often, the anomaly score for X directly represents some estimate of sparsity: number of points per area, or average distance between points, in the neighbourhood of X. The basic method is useful for detecting anomalies whose anomalous nature meets a globally set threshold: they lie in some sparse bit of the dataset for a uniform concept of sparsity.

Two-pass methods can capture more complex concepts of an anomaly by first computing some intermediate variable for all points, and then calculate an anomaly score for X as a function of the variables in the neighbourhood of X. These methods include:

1. The Local Outlier Factor (Breunig et al. 2000) method, a k-nearest neighbours method whose anomaly score for X represents whether the neighbourhood of X is sparser than the neighbourhoods of its neighbours. This allows LOF to identify anomalies in the presence of different clusters of various densities.

2. The Density-Based Noise (DBSCAN) method (Ester et al. 1996), an ϵ-neighbour method where points are designated core points if their neighbourhood has more than a minimum threshold of points, and then designated outliers if their neighbourhood contains no core points. DBSCAN is suitable for finding irregularly-shaped clusters, and can be formulated as a linear-time method (Gan and Tao 2015).

3. The k-NNN Nearest Neighbours of Neighbours (Nizan and Tal 2024) method, where the first pass computes a set of eigenvalues and eigenvectors for each point based on its k-nearest neighbours. Eigenvectors with small eigenvalues are assumed to be anomalous directions because the data does not spread out in those directions. A point's anomaly score is based upon how anomalous the direction pointing to it from each of its k neighbours is. k-NNN is suitable for working with datasets where non-anomalous data is roughly following a lower-dimensional manifold.

Neighbourhood methods don't use any information from points far enough away from X. This means they are good at handling irregular structure in data, but if the data structure is regular then they have less detection power than other methods.

Neighbourhood methods are independent of the coordinate system used to describe the data. Rotating our coordinates makes no difference to a point's neighbourhood. This means they are useful for applications where the choice of coordinate system is arbitrary - for example, spatial data using latitude and longitude. However, the computational methods used to calculate a point's neighbourhood often do not scale well with very large dimensions p (Liu, Moore, and Gray 2006).

Ensemble and isolation methods

Rather than comparing a point to the points in its neighbourhoods, isolation-based methods consider anomalies as the points that are easy to separate out from the other points in the dataset.

The first such method developed, iForest (Liu, Ting, and Zhou 2008) uses repeated axis-parallel subdivisions to isolate a point at the end of decision trees called isolation trees. Anomalous points are considered to be those points that are easier to isolate away from the other points of the dataset, and have a small average length of isolation tree. This leads to an execution time that is linear in the number of points in the dataset, which is very attractive computationally. However, its definition of an anomaly is a global definition, so it fails to find local anomalies.

Isolation-based Nearest Neighbour Ensemble (iNNE) (Bandaragoda et al. 2014) builds on this by using nearest neighbour ideas in order to construct an algorithm able to identify local anomalies, as well as having other advantages such as finding anomalies where the number of anomalous dimensions is low.

Both iForest and iNNE are examples of ensemble methods. Ensemble methods are based on drawing subsets from a large data sample to force an $O(n)$ complexity with respect to the size of the dataset. They use the idea that it is possible to construct multiple models by running the same anomaly detection method on different subsets of the data, and that different models make different errors of judgment which can be reduced by averaging their results.

For instance, if a method in its usual instance is $O(n^2)$ (a complexity that is unsuitable for large datasets), an ensemble version can be constructed as follows:

1. Draw a data subset of size $m << n$.

2. Run the method on this subset, requiring $O(m^2)$ operations, and constructing an approximate anomaly score.

3. Improve on this approximation by repeating this draw approximately n/m times and taking the average of the anomaly scores for each draw.

4. This constructs an $O(mn)$ method, which can be regarded as linear if $m << n$ and m doesn't need to grow with dataset size.

The extent to which this can produce a good method depends on how much of a drop-off

there is in approximation quality when you're only using m points rather than n, and whether this approximation quality is sufficiently restored by repeating the procedure m/n times.

High-dimensional outlier detection methods

In very high-dimensional datasets, standard multivariate anomaly detection methods can run into problems. Anomaly detection methods often have a complexity that scales poorly with the number of dimensions. High-dimensional datasets require more computational resources and time, making many traditional methods infeasible. In high-dimensional spaces, the Euclidean distance between all points becomes increasingly similar. This makes it difficult to distinguish between anomalies and non-anomalous data because the notion of proximity loses its meaning. High-dimensional data also tends to be sparse, making it hard to identify meaningful patterns. The density of points drops, leading to difficulties in density-based outlier detection methods.

Angle-based outlier detection (ABOD) (Kriegel, Schubert, and Zimek 2008) addresses the problems with distance and density by considering a point's anomaly score as a function of the angle at the point to a set of pairs of other points in the dataset. Outlier points to non-anomalous points will always give rise to small angles, whereas angles at non-anomalous points will be much more varied. ABOD as a method uses k-nearest neighbours as choices for other points, so scales with dataset size as any algorithm based on k-nearest neighbours. However, it scales linearly with dimension and is suitable for 100 or more dimensions.

Copula-based outlier detection (COPOD) (Li, Zhao, et al. 2020) is a fast method that involves estimating the multidimensional cumulative distribution function and then calculating a point's anomaly score as the probability of observing a point as or more extreme than the observation. This is done by constructing one-dimensional empirical cumulative distributions, as well as an empirical copula that captures the distributional dependency between the different dimensions. COPOD scales linearly with number of points n and number of dimensions d, and is suitable for very high numbers of

dimensions (1000 or more).

2.5.4 Multivariate collective anomaly detection

Working with data that is both multivariate and a time series can very quickly introduce large amounts of complexity to the anomaly detection problem. This is because multivariate methods tend to scale in complexity with the number of coordinates, and time series methods tend to scale with dataset size. These complexity scalings both multiply with each other directly, and can also further interact. In cases where either the time series or the multivariate aspect is not relevant - the data could be considered as either a multivariate dataset that happens to occur in time order or a collection of unrelated univariate time series - it is usually best to choose anomaly detection methods that consider only the relevant aspects of the problem in order to minimise complexity.

One common example class of multivariate time series detection problems is finding collective anomalies that are present in some number of coordinates at once and start and end at approximately the same time in all coordinates in which they appear. These are presumed to represent some underlying real-world process that is being picked up by one or more sensors. Here, the time series and multivariate aspects of the problem are inseparable.

A number of different methods can be employed to tackle this scenario. Mei (2010) contains details of simple and scalable monitoring methods for large numbers of data streams. Here, statistics such as the Page-CUSUM statistic (see Section 2.3.8) are calculated for each data stream and then either the sum or the maximum of all such statistics is used as a global monitor depending on if one is more interested in anomalies occurring in fewer or more data streams. Chen, Wang, and Samworth (2022) and Yang, Eckley, and Fearnhead (2024) both utilise local thresholding to attempt to identify changes present in more than one data stream while reducing the noise introduced by monitoring a large number of data streams. All these methods are approximate methods but they avoid iterating over the intervals in the signal. Fisch, Eckley, and Fearnhead (2021) also uses local thresholding for its penalties to tackle multivariate

aspects of the problem but is slower than other methods listed here due to the need to iterate over all possible anomaly lengths (see Section 2.3.6 for a description of the underlying computational issue in the univariate setting).

Tackling all these challenges simultaneously remains an open area of research. Chapter 6 gives contributions to this problem by adapting the FOCuS algorithm from a univariate to a multivariate setting and considering the problems that arise when this is done.

2.5.5 Summary

Multivariate anomaly detection methods vary widely in their approach and suitability for different types of data. The choice of method depends on factors such as data dimensionality, data structure, computational constraints, and the nature of anomalies. The best choice of anomaly detection method to use very much depends on your specific problem.

2.6 Assessing anomaly detection methods

Assessing an anomaly detection method can be challenging due to several inherent issues. First of all, you must have data to test it on where you already know something about the anomalies present in the data. Second, you must choose a scoring method that makes sense in the context of anomaly detection rather than other kinds of classification problems where your classes are more balanced. Finally, there are also specific scoring issues that arise in the context of working with time series such as how to score collective anomaly detections that contain overlap, and how to penalise for delayed detections in the online setting.

2.6.1 Dataset problems

Anomaly detection methods must be chosen with knowledge of your specific use case, because the concept of what should be classed as anomalous or not varies wildly between

different applications. Therefore, ideally in order to assess an anomaly detection method on your use case, you would want to have a test dataset where real anomalies are present and labelled. Doing this by hand on your own data can be time-consuming, because anomalies are sufficiently rare that identifying even a small number of them requires looking through a large amount of data. In addition to this, the person doing the labelling may need to have quite a lot of domain expertise in order to understand what features of the data are and aren't anomalous. This makes the labelling process very expensive.

An alternative is to use a publicly available benchmark dataset that comes with anomalies pre-labelled, such as the Numenta Anomaly Benchmark (Lavin and Ahmad 2015), the UCR Time Series Anomaly Datasets (Wu and Keogh 2023) or ADBench (Han et al. 2022). These have the advantage of a wide variety of different datasets and data characteristics, some of which are likely to provide a reasonable match for your problem. They have also pre-run and assessed large numbers of anomaly detection methods so you have a good starting point to identify what methods are best for your use case. For example, ADBench tested 30 different algorithms on 57 benchmark datasets. Problems with the benchmark approach include that with so many algorithms tested on so many datasets, algorithms are usually tested using default parameters (Soenen et al. 2021), and not enough care has been taken tailoring each algorithm to each dataset to generate the best performance. You might need to repeat the analysis yourself more carefully for the particular dataset best aligned to your use case. You also need to choose the detection threshold you are using and any other parameters of your method, which in a practical setting often involves such things as trend and seasonality estimation as well as the anomaly detection method itself. In addition to this, the scoring methodology used by various benchmarks has been criticised for overly favouring the anomaly detection methods developed by the researchers who created those benchmarks (Wu and Keogh 2023).

Lastly, you could simulate a dataset to test your anomaly detection methods on. Often, this consists of your own real-world dataset to which known simulated anomalies

	Actual positive	Actual negative
Predicted positive	True Positive	False Positive
Predicted negative	False Negative	True Negative

Table 2.6.2: Definitions of the confusion matrix: the four basic metrics from a binary classification problem. Blue represents correct classifications, and red incorrect ones.

are added, avoiding the labelling problems. These simulated anomalies can be of various different forms: point anomalies could be contextual or global, and collective anomalies could be trend, seasonal, or shape-based (Lai et al. 2021). This has the advantage of being able to generate data points to be exactly what you think an anomaly should look like, in order to score methods in detecting them. However, lack of real-world anomalies can limit your certainty that your method will perform well in practice.

2.6.2 Scoring metrics tailored to the rarity of anomalies

The confusion matrix

Anomaly detection methods can be regarded as binary classification algorithms that classify data as either anomalous or non-anomalous. Tharwat (2021) provides a general review of metrics for binary classification problems. The four basic metrics that can be collected from running a binary classification algorithm on a non-temporal dataset are the numbers of true positives TP, false positives FP, false negatives FN, and true negatives TN, defined as in Table 2.6.2. This matrix is called the confusion matrix (Stehman 1997). From these base metrics, different ways of evaluating and scoring an algorithm exist, some of which are more useful in anomaly detection than others.

We will first consider the problems with standard scoring methods when applied to anomaly detection. One way of evaluating classification in the machine learning sphere is to look at the proportion of data classified correctly:

$$\text{proportion correctly classified} = \frac{TP + TN}{TP + FP + TN + FN}.$$

However, if anomalies are assumed to be $< 0.1\%$ of the data, a classifier that scores nothing anomalous would have a $> 99.9\%$ correct proportion of classification. A method

which found all n anomalies but had $2n$ false positives would score worse on the proportion correctly classified measure, despite clearly being a far more useful method.

Problems with anomaly rarity

Anomaly detection problems have a class imbalance where non-anomalous data massively outnumbers anomalous instances. The number of true negatives (non-anomalous points not classified as anomalous) TN cannot be compared well with any of the other three metrics because it is always very large due to the rare nature of anomalies. This means that any evaluation metric using TN in a sum tends to be skewed and not a good measure. For example, the standard diagnostic plot of the receiver operator characteristic curve (true positive rate versus false positive rate as you raise the detection threshold) for classifiers is not a good visual plot for anomaly detection problems. True positive rate and false positive rate are defined as follows:

$$\text{true positive rate} = \frac{TP}{TP + FN},$$

$$\text{false positive rate} = \frac{FP}{FP + TN}.$$

This is because true positive rate and false positive rate cannot be compared to each other very meaningfully due to the use of true negatives in calculating the false positive rate. A difference between an 0.1% and an 0.2% false positive rate has huge implications for the viability of an anomaly detection method. This is very hard to see visually, and is not captured well by metrics such as the area under the receiver operator characteristic (ROC) curve, that is mostly contributed to by very low threshold levels where the false positive rate is high, as is shown in Figure 2.6.16. Therefore, standard metrics such as the area under the ROC curve do not capture the important information for anomaly detection problems.

True positive rate alone is not sufficient as a single metric, as a method classifying everything as an anomaly would have perfect true positive rate.

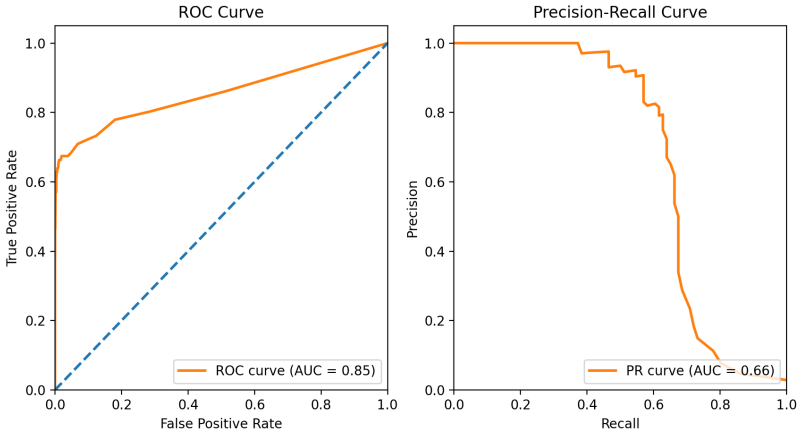


Figure 2.6.16: A comparison of the ROC curve and precision-recall curve for an anomaly detection method run on a dataset consisting of 2% anomalous data. Most of the important information happens on the extreme left-hand side of the ROC curve.

Precision, recall, and F-scores

Instead, we use the concepts of precision and recall as useful measurements of an anomaly detection method's performance.

$$\text{precision} = \frac{TP}{TP + FP},$$

$$\text{recall} = \frac{TP}{TP + FN}.$$

Precision is the proportion of detections that correspond to anomalies, whereas recall is the proportion of anomalies detected (and is the same as the true positive rate). We may wish to combine them into a single global measure, in which case a popular choice would be the F_1-score, the harmonic mean of precision and recall.

$$\mathrm{F}_1 \text{ score} = \frac{2}{\text{precision}^{-1} + \text{recall}^{-1}} = \frac{2TP}{2TP + FP + FN}.$$

For different applications, precision or recall may be more valuable. In this case, if we weight precision as β times as important as recall, we would use the F_β score (Van Rijsbergen 1979). Methods implemented in code often use a general F-score, with

the default settings as F_1 score but with β able to be specified if desired.

$$F_\beta \text{ score} = \frac{1+\beta^2}{\text{precision}^{-1} + \beta^2 \text{recall}^{-1}} = \frac{(1-\beta)^2 TP}{(1+\beta)^2 TP + FP + \beta^2 FN}.$$

To see our results visually, and for a range of different thresholds for classifying data as anomalous, we would use a diagnostic plot called a precision-recall curve (see Figure 2.6.16). The area under the precision-recall curve is classification threshold invariant. It considers all possible weightings of precision versus recall. This makes it a good measure for a researcher wanting to report the overall performance of an anomaly detection method. However, it's less useful for a practitioner, who is interested in the method's performance for a particular problem, and often wants to choose thresholds caring about either a specific ratio of precision to recall, or seeks to maximise recall for a fixed precision (or vice versa). (Garg et al. 2022)

2.6.3 Scoring metrics tailored to time series

Metrics like precision, recall, and F-score do not capture the temporal aspects of anomalies. There are additional aspects of scoring both point and collective anomaly detection methods that must be taken into account to properly evaluate a method, such as how close to the actual anomaly the detection estimated the anomaly to be. Additionally, when working in the online setting it is important to consider delay to detection as a separate matter from estimation error.

Scoring point anomalies

Anomalies found by an algorithm in a time series dataset may not exactly overlap with true anomaly locations, but instead be close by. Therefore, when evaluating an anomaly detection algorithm, it is required to determine how close a match will be recorded as a true positive. Rather than having sharp thresholds, most time series anomly detection scoring algorithms will weight points as true positives smoothly into false positives depending on distance. Once the weighted constructions of TP, FP, FN

have been created, modified versions of precision, recall and F-score are derived from these in order to score the algorithm.

Often there is a scoring function that operates on a sliding scale, with detections further away from the true anomaly being scored worse. This opens up the problem of what to do when multiple detections happen near a single true anomaly. Should additional detections be scored as true, disregarded, or scored as false?

Figure 2.6.17 shows the approach used by the Numenta Anomaly Benchmark (Lavin and Ahmad 2015), where a scoring window and sigmoid drop-off is proposed for each anomaly based on the ground truth. Detections near the beginning of the scoring window are given nearly full credit as a true positive, additional detections after this within the window are ignored, and detections only near the end of the window are still counted as true positive but they are penalised. For false detections, those slightly after the window are scored as false positives but not as strongly, whereas those a long time after the window are scored fully as false positives. Missing a window entirely records as a false negative.

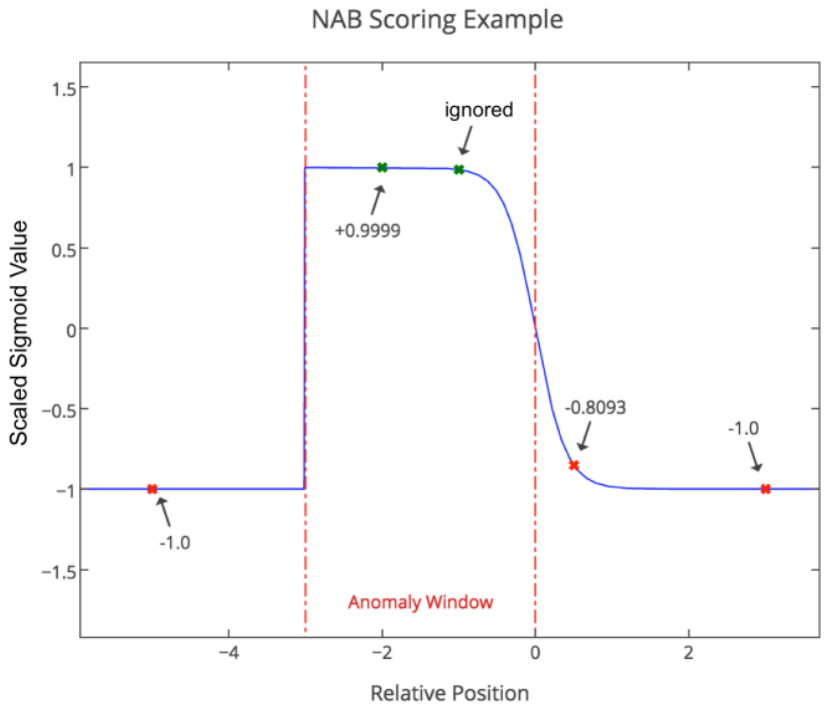


Figure 2.6.17: The scoring window for anomalies used by the Numenta Anomaly Benchmark

Here, the size of the scoring window affects the performance of a method, with wider scoring windows being more lenient to late detections. The scoring window can be a user-specified parameter. The Numenta Anomaly Benchmark sets the length of the scoring window as equal in length for each anomaly, and inversely proportional to the number of real anomalies in the dataset.

Scoring collective anomalies

Collective anomalies have many of the same problems as point anomalies, but also require evaluation methods that account for duration and sequence. Here, a detected interval may not exactly overlap with the true anomalous interval. Should the area of overlap be scored, or should each anomalous interval be given a separate score that is not in proportion to its area? If this is the case, how will the scoring handle where a detector flags a lot of short intervals, but the true anomaly is a large interval containing them all? A variety of different methods have been proposed.

One method for scoring collective anomaly detections, used by Hundman et al. (2018), is the eventwise F-score. Here, we only care about overlap and do not adjust for distance: a detection is recorded as a true positive if it overlaps at all with one or more anomalies, a false positive if not, and additional detections of the same anomaly are disregarded. Anomalies that do not overlap with any detections are recorded as false negatives. However, it incentivises methods that detect very large intervals as anomalous, as it is likely they overlap some real anomaly somewhere: a method flagging the whole dataset as one very long detection would have a maximum eventwise F-score.

A different method, known as the pointwise-adjusted F-score (Xu et al. 2018), mitigates this by scoring points individually rather than events. However, an adjustment is made: points within an anomaly that overlaps with a detection are all scored as true positives, regardless of whether they were within the overlap. This mitigates the issue of collective anomalies often containing small sections that look non-anomalous in isolation. However, it has the opposite problem to the eventwise F score: especially on datasets where anomalies are quite long, it favours anomaly detection methods that

detect very short intervals, as long detections can be heavily penalised. (Audibert et al. 2020)

The composite F-score (Garg et al. 2022) uses the harmonic mean of the (non-adjusted) pointwise precision and the eventwise recall. By using pointwise precision, the method penalises long false positives and avoids problems with the eventwise F-score. By using eventwise recall, it avoids the need for an adjustment to deal with collective anomalies containing sections that appear non-anomalous in isolation.

Each of these three methods has the advantage of being relatively simple, and not requiring weighting on distance-based drop-off parameters as is used for point anomalies. In addition to this, no difference in scoring is made based on anomaly intensity: we may care more about detecting some anomalies than others.

2.6.4 Scoring metrics for online data streams

When working with streaming data, it is necessary to detect anomalies as soon as possible after they occur. We may also care about how much of a collective anomaly is needed to be observed before a detection was made, which is different from the estimated startpoint of the anomaly.

A different approach to scoring a collective anomaly detection method is by reporting its average run length and detection delay, similar to the in-control average run length and out-of-control average run length initially proposed by Page (1954).

Average run length

The run length is the time until returning a false positive when a detection method is run on a signal containing no anomalies. The average run length is an average of this over many different runs, often the mean, but median average run lengths and distributional boxplots (see Figure 2.6.18) can also be used as the distribution of run lengths tends to be positively skewed. (Lee and Khoo 2006)

Average run length is a replacement for precision, and can roughly be related to the false positive rate as follows:

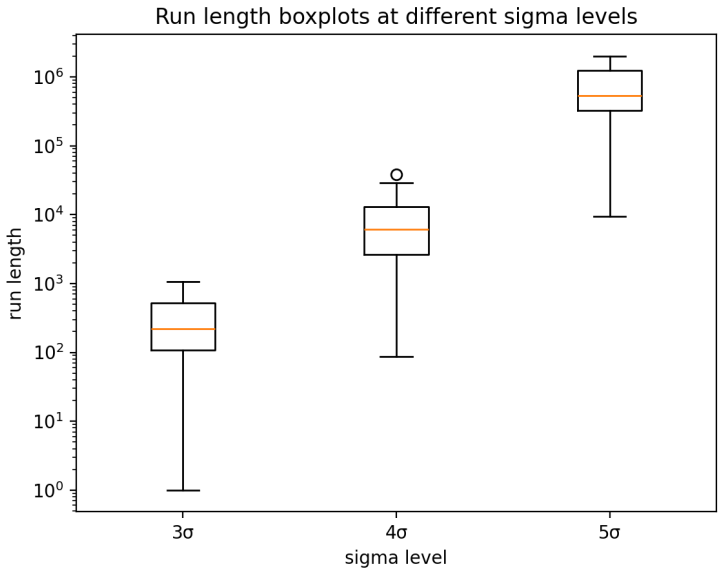


Figure 2.6.18: Boxplots for a method's run length over different sigma thresholds. Run lengths are plotted on a log scale for sensible comparison, showing that the distribution of run lengths is positively skewed.

$$\text{average run length} \approx \frac{1}{\text{false positive rate}} \approx \frac{TN}{FP}.$$

However, false positives will often occur nearby each other in temporal data where the method is building an anomaly score contributed to by nearby points. For this reason, when measuring runs the algorithm's memory is cleared out between runs, and some time may be built in for the method to reset after a false detection before starting another run. This makes the average run length more lenient than just the false positive rate, and more useful for the live online setting. For example, if a signal returns a detection and a human checks in on it, they may watch it for some time to see how it develops regardless of what it then does. Lots of false positives clustered near each other are less costly than evenly spaced false positives in the currency of human attention.

For data of a given distribution, it may be possible to calculate the average run length exactly as a function only of the threshold used by the anomaly detection method, either exactly by using integral equations (Petcharat, Sukparungsee, and Areepong 2015) or

by simulating data to give an empirical estimate. If anomaly scores are available rather than over/under threshold, plots of the average run length over many different threshold levels can be easily generated without having to re-run the algorithm for each threshold level, as in Figure 2.6.19.

Mean average run length is a generally well-accepted metric, and for data of a fixed distribution its existence is well-defined and unique (Areepong and Peerajit 2022). Some criticisms of mean average run length include the existence of schemes with expanding thresholds where the mean average run length is infinite but that return a false detection with probability 1 (Mei 2008), in which case median average run length can be used for comparisons.

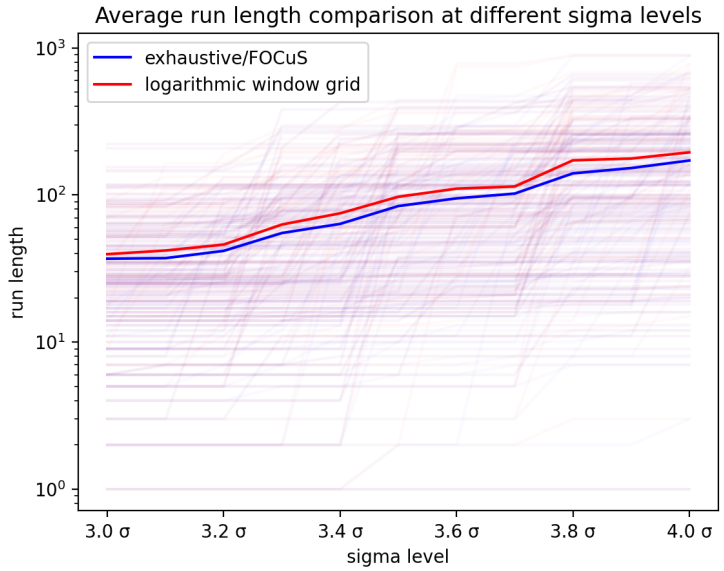


Figure 2.6.19: Average run length with run lengths over different σ threshold levels for two methods, used to calculate a threshold adjustment between methods of about 0.1σ in order to equalise the average run lengths.

Detection delay

The detection delay is the time until returning a true positive when a detection method is run over an anomaly. Detection delay is a replacement for recall, however it accounts for how long an anomaly took to detect rather than just if it was detected or not.

This means that detection delay is useful for evaluating an algorithm's performance on longer less intense anomalies, or on anomalies that start less intense but become more intense over time.

Detection delay will vary by the kind of anomaly as well as by the threshold level. Broadly, larger anomalies and lower thresholds will have less detection delay, as shown in Figure 2.6.20. Lower thresholds will also lead to lower average run lengths, giving a tradeoff between average run length and detection delay.

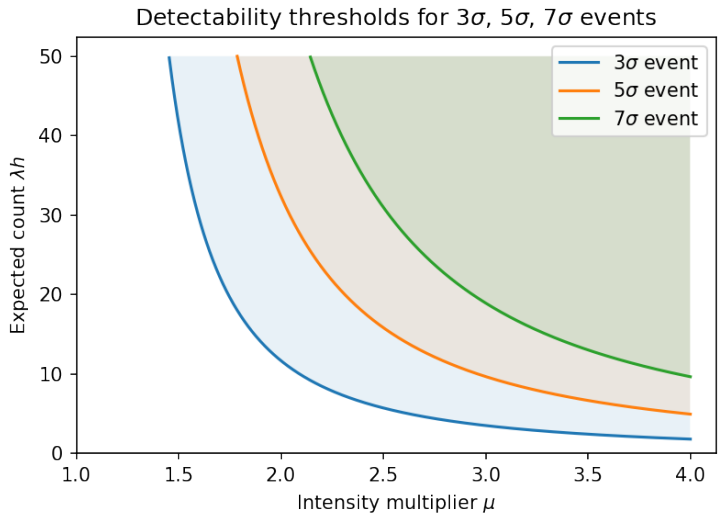


Figure 2.6.20: Graph showing detection delays for various anomaly intensities μ at three different sigma thresholds using a likelihood ratio test method over all intervals.

Detection delay does not always equal the out-of-control average run length, and cannot always be measured simply by starting an algorithm off running over an anomaly. For example, anomaly detection methods that accumulate evidence against an anomaly when run over non-anomalous data that must be overcome will have very high detection delays but perform well when started on an anomaly (Lorden 1971).

Detection delay can also be contributed to by the part of the anomaly detection method that is estimating or tracking the background rate. Many anomaly detection methods will use centred estimates, such as an estimation of the background mean μ_T at time T by using the mean over a sliding window centred on T:

$$\mu_T = \frac{1}{2w+1} \sum_{t=T-w}^{t=T+w} x_t$$

This means that the estimate μ_T is only available at time $T + w$, introducing an additional detection delay w proportional to the length of the sliding window. For long windows that are used to generate precise estimates, this can be a substantial addition to the detection delay.

For these two reasons, measuring detection delay should ideally take place in a setting containing both anomalous and non-anomalous data.

Unlike the F-score that combines precision and recall, there are no accepted metrics for combining average run length and detection delays into a single score. For evaluating a specific application a fixed average run length is chosen based on the specifics of the application, and thresholds are adjusted across methods to equalise the average run length, as shown in Figure 2.6.19. Given that constraint the detection delays on various kinds of anomalies can be compared against each other.

2.6.5 Summary

Assessing an anomaly detection method presents a range of challenges due to the need for appropriately labeled datasets, suitable scoring methods, and the specific nuances involved in time series data. The scoring of an anomaly detection method must be informed by the application, as the definition of anomalies can vary significantly across different domains. Public benchmark datasets offer valuable resources but require further fine-tuning to match your use case.

Scoring metrics tailored to the rarity of anomalies, such as precision, recall, and F-scores, are more appropriate to anomaly detection than traditional binary classification metrics, because anomalies are rare. These can be adapted for the time series setting, where they must also account for the proximity and overlap between detected anomalies and true anomalies. However in online time series settings, average run length and detection delay provide important information that are not captured by other metrics.

Ultimately, the choice of how to score a method will depend on your application. In the next section, we look at some application domains for anomaly detection that are relevant to the methods developed in this thesis.

2.7 Some Applications of Anomaly Detection Methods

Here, we review some possible application areas for real-time high-frequency time series collective anomaly detection methods.

2.7.1 Astrostatistics

Astrostatistics is an application field involving the use of mathematical or statistical techniques to extract information from observations of the sky. These observations can come from satellites or ground-based telescopes. They are usually based on the electromagnetic spectrum (light), though other signals such as gravitational waves (Bailes 2021) and neutrinos (Guépin, Kotera, and Oikonomou 2022) are possible. A light curve is a time series of light intensities, and a short-lived astronomical event that may produce a collective anomaly in this light curve is called a transient. Transients can brighten the light curve, such as a gamma-ray burst (see Chapter 3), or they can dim the light curve, such as an exoplanet crossing in front of a star (Borucki et al. 2010).

Measuring high-energy light such as X-rays and gamma rays often involves counting each photon, whereas measuring low-energy light such as radio and infrared involves taking exposure photographs of a region of the sky over a few seconds or minutes. A light curve may also be multivariate consisting of different spectral intensity bands, and a transient may be expected to show up in more than one band at once.

There is a huge volume of data produced by modern astronomical instruments, and analysing it all for transients can be computationally intensive. Even classifying the transients that have been found can take a lot of computing power. For example, the Rubin Observatory Legacy Survey of Space and Time observes millions of transient alerts each night (*LSST: From Science Drivers to Reference Design and Anticipated*

Data Products 2014), and must classify them in order to identify interesting ones suitable for visual inspection by a human (Muthukrishna et al. 2022).

Time series anomaly detection methods that have been employed in astrostatistics applications include CAPA for detecting exoplanets (Fisch, Eckley, and Fearnhead 2022), and FOCuS for detecting gamma-ray bursts (see Chapter 3).

2.7.2 Radiation detection

Radioactive decay is the process by which unstable atoms break down into stable ones, emitting various kinds of particles. Of these, gamma particles (high-energy photons) are the most suitable for detecting radiation sources in the environment, because they travel a few metres through air before decaying and can pass through many solid objects, being stopped only by large masses like lead and concrete.

Radiation detection is used to detect the presence of radioactive material in the environment near the sensor. Radioactive material can occur naturally or due to human activity, which happens for many reasons mostly related to mining, extracting or burning oil or gas (al-Nabhani, Khan, and Yang 2016), nuclear power, and specifically created radioactive sources used for sterilisation or in medical applications (International Atomic Energy Agency 2024). Very high levels of radiation are hazardous to health.

The spectral signature of a material is the electromagnetic wavelength/energy bands at which it emits gamma particles. The signature of the decays from an individual longer-lived radioactive isotope can be expected to occur together with the signatures of its shorter-lived decay products, giving an overall signature for the presence of that longer-lived isotope in a material. In equilibrium, each successive isotope in the decay chain is present in direct proportion to its half-life. Since its activity is inversely proportional to its half-life, each isotope in the decay chain contributes the same rate of decay. Only some of these decays will be measurable as gamma radiation, defining the gamma radioactive detection signature of the longer-lived isotope fairly precisely (Bateman 1910).

Borehole monitoring is one possible area for radiation time series anomaly detection (Elísio et al. 2023). Here, a radiation detector is lowered slowly down a deep, narrow hole in the ground. The radiation signatures found by the detector provide a map of the radioactive material at different depths within the borehole. This is then combined with other data taken from the borehole such as groundwater samples, and can be used to estimate the percentages of different mineral types within the bedrock.

Handheld monitoring is another area. Here, an anomaly detection method must pick up and classify sources as the detector is moved around. This may be used for scanning shipping containers at border ports (Connolly, Connor, and Martin 2023). Computationally scaleable methods are particularly preferred in handheld applications because they lead to longer battery lives for handheld units and the amount of data collected this way is large (Russell-Pavier et al. 2023).

2.7.3 Telecommunications monitoring

Anomaly detection may be used to identify unusual patterns in the traffic flows through a telecommunications network. Data in internet telecommunications networks consists of individual internet packets (IP), which are routed through the network based on their source and destination IP addresses. Packets that discretise a live video or audio signal do so very frequently in order to reduce delays. For example, the time interval in live audio data that is assigned to each internet packet varies from 5-20ms depending on the standard used (Zurawski 2004b). This means there is a high velocity of packets passing through a router, so sensors usually monitor summaries of them rather than working on the raw data of each individual packet.

Anomaly detection and resolution plays an important part of quality of service monitoring. Often, a telecommunications provider will have service-level agreements that specify what service they must provide, and have to pay fines if these agreements are not met. There are several metrics of quality that might feature in a service-level agreement (Zurawski 2004a). These include:

1. Latency: the average packet travel time. If latency is high this will cause delay in

real-time communication.

2. Jitter: the variance in packet travel times. If jitter is high packets will arrive out of order and this reduces the smoothness of video or audio.

3. Packet loss: the proportion of packets that do not arrive at their destination and must be resent, causing delays.

4. Overall availability of the network. Availability is often expressed in terms of "nines" (Merzbacher and Patterson 2003). A four nine system is expected to be available 99.99% of the time whereas a five nine system is expected to be available 99.999% of the time, equating to only a few minutes of downtime per year.

Anomaly detection methods are used to identify developing problems within the network before they reach a point where they impact the quality of the service the network provides. In particular, identifying periods of potential downtime before they happen means that action can be taken quickly to resolve the issue, preserving the availability of the system (Chumash 2006). Collective anomaly detection methods are helpful because they can identify an anomaly before any individual signal point becomes unusual.

Telecommunications data is very seasonal, with clear daily and weekly patterns of how people like to use the internet. Anomalies within a telecommunications network may indicate changes in human behaviour rather than network problems. For example, a breaking news story or sports match may cause a surge in internet usage (Wang and Kim 2019). However, problems that can cause an anomaly in a telecommunications network that may need detecting immediately include:

1. A network intrusion, security threat or cyberattack.

2. A hardware fault that needs a engineer to be called out to fix it.

3. Excess congestion on the network, which may be caused by hardware faults in other areas causing traffic to be rerouted through this line.

In Internet of Things (IoT) applications, lots of small sensors monitor data in real-time. Each sensor decides when to transmit data to be processed centrally. Rather than

transmitting all data or updates at regular intervals, we may decide to only send data that looks anomalous in order to reduce transmission cost. This means we must use anomaly detection methods locally on the small sensor. The sensor may have restricted computing power or battery life, and a high velocity of data being collected.

2.8 Summary

In this review, we have defined the specific concepts related to anomaly detection that we will be using throughout the rest of the thesis. We have looked at several areas of the relevant anomaly detection literature, including working with time series, both as context and as a computational interval search problem. We have looked at various point anomaly detection methods for multivariate data. We have also explored how to assess an anomaly detection method using metrics such as precision and recall, and how these can be adapted to work with time series both online and offline. Finally, we have briefly explored three key application areas: astrostatistics, nuclear radiation monitoring, and telecommunications. These application areas form the basis of the collaborations behind the work done within this thesis.

Chapter 3

Poisson-FOCuS for detecting gamma ray bursts

This chapter is a reproduction of the Ward, Dilillo, et al. (2023) paper published in the Journal of the American Statistical Association. It details novel work to adapt the FOCuS algorithm (Romano, Eckley, Fearnhead, and Rigaill 2023), originally developed for the Gaussian setting, to the Poisson setting in order to handle count data. Minor edits have been made to render the paper suitable for a thesis chapter.

3.1 Introduction

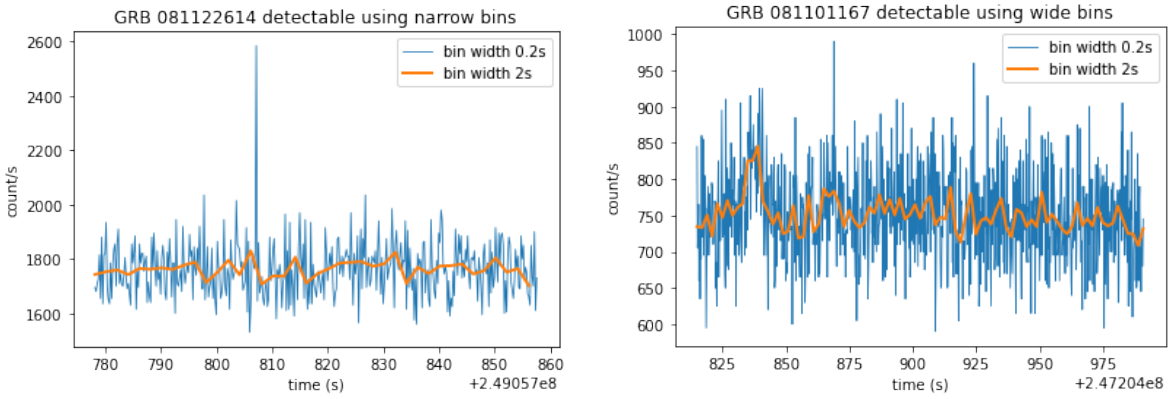
This work is motivated by the challenge of designing an efficient algorithm for detecting gamma ray bursts (GRBs) for cube satellites, such as the HERMES scientific pathfinder mission (Fiore et al. 2020). GRBs are short-lived bursts of gamma ray light caused by the catastrophic accretion of matter into newly formed black holes. Long GRBs are associated with the formation of black holes in the collapse of massive, rapidly rotating stars, whereas short GRBs are associated with coalescence events in neutron star binary systems. These bursts were first detected by satellites in the late 1960s (Klebesadel, Strong, et al. 1973). At the time of writing there is considerable interest in detecting gamma ray bursts due to their association with gravitational wave events (Luongo

and Muccino 2021). For example, during 17 August 2017, the combined detection of the short gamma ray burst GRB170817 and the gravitational wave event GW170817 constrained the source's location to a region of about 1100 deg^2, roughly the size of the Ursa Major constellation in the night sky (Abbott, Abbott, et al. 2017). This limited the number of candidate host galaxies to a pool small enough to identify an optical counterpart. A large broadband observation campaign started soon after, leading to insights into many aspects of gravity and astrophysics (Miller 2017).

Instruments detecting GRBs must operate in space, as gamma light wavelengths are absorbed by nitrogen and oxygen in the upper layers of the atmosphere. Because of this, multiple cube satellite missions are being deployed to study them in the coming years (Bloser et al. 2022). Cube satellites are compact and therefore relatively cheap to launch into space, but have limited computational power on-board. One of these missions is HERMES (High Energy Rapid Modular Ensemble of Satellites), a mission whose first six units will be launched in near-equatorial orbits during 2023, aiming to build an all-sky monitor for GRBs and other high-energy astronomical transients (Fiore, Burderi, et al. 2020). The HERMES main scientific goal is to monitor the whole sky for GRBs and locate their source directions.

Raw data from a satellite consists of a data stream of photons impacting a detector. The time of a photon impact is recorded in units of microseconds since satellite launch. New photon impacts are recorded on the order of approximately one every 500 microseconds (Campana et al. 2020). A GRB is indicated by a short period of time with an unusually high incidence of photons impacting the detector. Ideally the satellite would detect each GRB, and for each burst it detects it then transmits the associated data to earth.

There are a number of statistical challenges associated with detecting GRBs. First, bursts can come from close or far away sources with different intrinsic luminosities, and can therefore be either very bright and obvious to observe or very dim and hard to pick out from other background sources. They can also impact the detector over a variety of timescales. Figure 3.1.1 shows two GRBs, one short and intense lasting a fraction of a



(a) a short, intense burst

(b) a long, less intense burst

Figure 3.1.1: Plots of two recorded gamma ray bursts from the FERMI catalogue, with photon counts binned into 0.2s and 2s intervals.

second, one longer and less intense lasting about ten seconds. These bursts were taken from the FERMI catalogue (Axelsson, Bissaldi, Omodei, et al. 2019). Bursts ranging from a fraction of a second to a few minutes are possible (Kumar and Zhang 2015). Secondly, less than one GRB is recorded per 24 hours on average (Von Kienlin, Meegan, Paciesas, Bhat, et al. 2020), which is relatively rare in comparison to the velocity of the signal. The background rate at which the satellite detects photons also varies over time. This is due to both rotations of the spacecraft and features of the near-Earth radiation environment at different points in orbit, which leads to irregularly cyclic patterns. This variation is on timescales much larger (many minutes to days) than those on which bursts occur (milliseconds to seconds), and thus is able to be estimated separately from the bursts. Figure 3.1.2 gives an example of a background signal.

Finally, there are also computational challenges. For example, there is limited computational hardware on board the satellite, and additional constraints arise on the use of these due to battery life and lack of heat dissipation (Fenimore et al. 2003). There is also a substantial computational and energy cost to transmitting data to earth, so only promising data should be sent. This means we require any detection system to have a very low false positive rate.

At a fundamental level, algorithmic techniques for detecting GRBs have gone unchanged through different generations of space-born GRB monitor experiments (Mee-

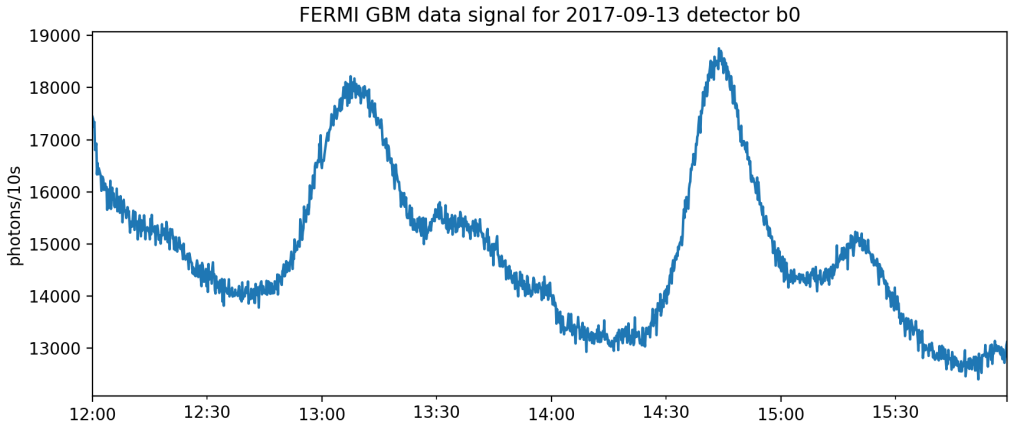


Figure 3.1.2: Example 4 hours of background data from one detector, grouped into 10 second bins to show background rate fluctuations.

gan, Lichti, et al. 2009; Fenimore et al. 2003; Paciesas, Meegan, et al. 1999; Feroci, Frontera, et al. 1997). As they reach a detector, high-energy photons are counted over a fundamental time interval and in different energy bands. Count rates are then compared against a background estimate over a number of pre-defined timescales (Li and Ma 1983). To minimize the chance of missing a burst due to a mismatch between the event activity and the length of the tested timescales, multiple different timescales are simultaneously evaluated. Whenever the significance of the excess count is found to exceed a threshold value for that timescale, a trigger is issued.

Figure 3.1.3 gives a simplified overview of such a detection system. As data arrives we need to both detect whether a gamma ray burst is happening, and update our estimates of the background photon arrival rate. Because of the high computational cost of transmitting data to earth after a detection, if an algorithm detects a potential gamma ray burst there is an additional quality assurance step to determine whether it should be transmitted. This step often includes checking that a gamma ray burst has been detected at two or more detectors on the satellite. The detection algorithm needs to be run at a resolution at which all gamma ray bursts are detectable. By comparison background re-estimation is only required once every second, and the quality assurance algorithm is only needed every time a potential gamma ray burst is detected. Thus

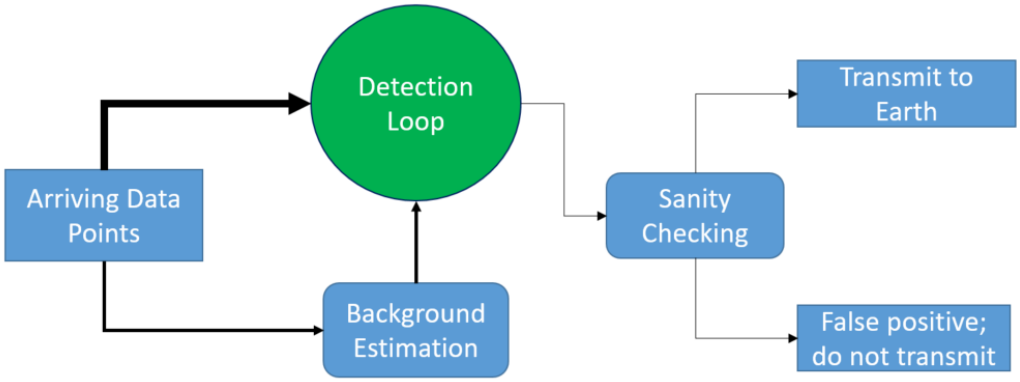


Figure 3.1.3: A schematic of the detection system, with the arrow thickness corresponding to the relative velocities of data flows. Most of the computing requirements of the trigger algorithm are within the detection loop, highlighted in green.

the majority of the computational effort is for the detection algorithm – and how to construct a statistically efficient detection algorithm with low computation is the focus of this paper.

As mentioned, current practice for detecting a GRB is to compare observed photon counts with expected counts across a given bin width (Paciesas et al. 2012). However, the choice of bin width affects the ease of discovery of different sizes of burst. Figure 3.1.1 shows an example. The short burst in Figure 3.1.1a is easily detectable with a bin width of 0.2s, but lost to smoothing at a 2s bin width. In contrast, the burst in Figure 3.1.1b has a signal too small relative to the noise to be detectable at a 0.2s bin width, with the largest observation on the plot being part of the noise rather than the gamma ray burst. Only when smoothed to a bin width of $2s$ does the burst become visually apparent. Therefore, the bin width is first chosen small enough to pick up short bursts, and geometrically spaced windows of size 1, 2, 4, 8, ... times the bin width, up to a maximum window size, are run over the data in order to pick up longer bursts.

This paper develops an improved approach to detecting GRBs. First we show that using the Page-CUSUM statistic (Page 1954; Page 1955), and its extension to count data (Lucas 1985) is uniformly better than using a window-based procedure. These schemes require specifying both the pre-change and post-change behaviour of the data - in our case, this equates to specifying the background rate of photon arrivals and the

rate during the gamma ray burst. While it is reasonable to assume, for our application, that good estimates of the background photon arrival rate are available, specifying the photon arrival rate for the gamma ray burst is difficult due to their heterogeneity in terms of intensity. For detecting changes in mean in Gaussian data, Romano, Eckley, Fearnhead, and Rigaill (2023) show how one can implement the sequential scheme of Page (1955) simultaneously for all possible post-change means, and call the resulting algorithm FOCuS. Our detection algorithm involves the non-trivial extension of this approach to the setting of detecting changes in the rate of events for count data. It is based on modelling the arrival of photons on the detector as a Poisson process, and we thus call our detection algorithm Poisson-FOCuS.

Our algorithm is equivalent to checking windows of any length, with a modified version equivalent to checking windows of any length up to a maximum size. This makes it advantageous for detecting bursts near the chosen statistical threshold whose length is not well described by a geometrically spaced window. In addition, the algorithm we develop has a computational cost lower than the geometric spacing approach, resulting in a uniform improvement on the methods already used for this application with no required trade-off. These advantages mean that the Poisson-FOCuS algorithm is currently planned to be used as part of the trigger algorithm of the HERMES satellites.

Our improvement of existing window based methods addresses the aspect of trigger algorithms that has been shown to be most important for increasing power of detecting GRBs. As the computational resource on-board a satellite have increased, trigger algorithms have grown to support an increasing number of criteria, and it has been seen that the most important aspect of any detection procedure is the timescale over which the data is analyzed (McLean et al. 2004). This is in contrast to other possible aspects such as testing data accumulated over multiple, fine-grained energy bands outside the standard 50-300 keV energy band. This did not result in more GRBs being detected by Fermi-GBM, and was eventually turned off to ease the computational burden on the on-board computer (Paciesas et al. 2012).

While early software, such as Compton-BATSE (Gehrels et al. 1993), operated only

a few different trigger criteria, a total of 120 are available to the Fermi-GBM (Meegan, Lichti, et al. 2009) and more than 800 to the Swift-BAT (McLean et al. 2004) flight software. Whilst in many cases, this growth in algorithm complexity did not result in additional GRB detection, better coverage of different timescales for GRBs did (Paciesas et al. 2012). During the first four years of Fermi-GBM operations, 135 out of 953 GRBs were detected only over timescales not represented by BATSE algorithms, most of which were over timescales larger than the maximum value tested by BATSE (1.024 s) (Von Kienlin, Meegan, Paciesas, Bhat, et al. 2014).

In Section 3.2 we define the mathematical setup of the problem. Section 3.3 introduces the functional pruning approach, leading to an algorithm and computational implementation specified in Section 3.4. In Section 3.5 we give an evaluation of our method on various simulated data, and real data taken from the FERMI catalogue.

3.2 Modelling framework

The data we consider take the form of a time-series of arrival times of photons. We can model the generating process for these points as a Poisson process with background parameter $\lambda(t)$, defining anomalies as periods of time which see an increase in the arrival rate over background level. By identifying anomalies, we hope to detect the GRBs that may cause them.

Changes in the background rate, $\lambda(t)$, over time may be due to rotation of the spacecraft or its orbit around the earth. They exist on a greater timescale (minutes to days) than the region of time over which a GRB could occur (seconds). We assume that a good estimate of the current background rate $\lambda(t)$ is available. To ease exposition we will first assume this rate is known and constant, and denote it as λ, before generalising to the non-constant background rate in Section 3.4.1. We discuss accounting for error in estimating the background rate in Section 3.5.2. No autocorrelation is present in our data when this change in background rate has been accounted for, see the Supplementary Material.

The standard approach to analysing our data is to choose a small time interval, w, which is smaller than the shortest GRB that we want to detect. Then the data can be summarised by the number of photon arrivals in time bins of length w. We denote the data by $x_1, x_2, \ldots$, with x_i denoting the number of arrivals in the ith time window. We use the notation $x_{t+1:t+h} = (x_{t+1}, \ldots, x_{t+h})$ to denote the vector of observations between the $(t+1)$th and $(t+h)$th time window, and

$$\bar{x}_{t+1:t+h} = \frac{1}{h} \sum_{i=t+1}^{t+h} x_i,$$

the mean of these observations. Under our model, if there is no gamma ray burst then each x_i is a realisation of X_i, an independent Poisson random variable with parameter λ. If there is a gamma ray burst then the number of photon arrivals will be Poisson distributed with a rate larger than λ. We make the assumption that a gamma ray burst can be characterised by a width, h, and an intensity $\mu > 1$ such that if the gamma ray burst starts at time $t+1$ then $x_{t+1}, \ldots, x_{t+h}$ are realisations of independent Poisson random variables $X_{t+1}, \ldots, X_{t+h}$ with mean $\mu\lambda$. See Figure 3.2.4 for a visualisation of an anomaly simulated directly from this model.

Our algorithm is primarily interested in reducing the computational requirements of constant signal monitoring. Therefore our model considers a gamma ray burst as a uniform increase in intensity over its length, which does not take into account the unknown shape of a gamma ray burst. If a possible burst is found, an additional round of shape-based sanity checking requiring more computational resources can easily be performed prior to transmission back to Earth.

3.2.1 Window-based methods and detectability

If we assumed we knew the width of the gamma ray burst, h, then detecting it would correspond to testing, for each start time t, between the following two hypotheses:

- $\mathbf{H}_0$: $X_{t+1}, \ldots, X_{t+h} \sim \text{Poisson}(\lambda)$.

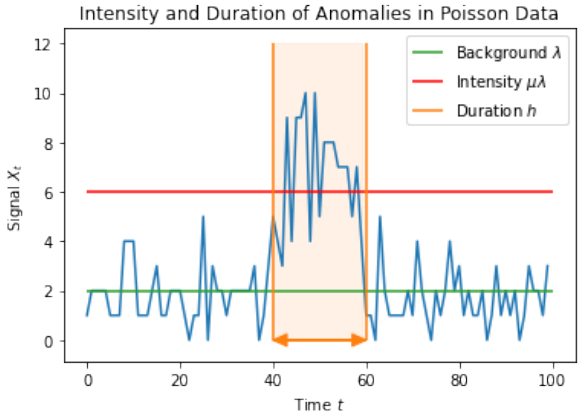


Figure 3.2.4: A simulated example anomaly with intensity multiplier $\mu = 3$ and duration $h = 20$ against a background $\lambda = 2$.

- $\mathbf{H}_1$: $X_{t+1}, \ldots, X_{t+h} \sim \text{Poisson}(\mu\lambda)$, for some $\mu > 1$.

We can perform a likelihood ratio test for this hypothesis. Let $\ell(x_{t+1:t+h}; \mu)$ denote the log-likelihood for the data $x_{t+1:t+h}$ under our Poisson model with rate $\mu\lambda$. Then the standard (log) likelihood ratio statistic is

$$LR = 2\left\{\max_{\mu>1} \ell(x_{t+1:t+h}; \mu) - \ell(x_{t+1:t+h}; 1)\right\}.$$

This is 0 if $\bar{x}_{t+1:t+h} \leq \lambda$, otherwise

$$LR = 2h\lambda\left\{\frac{\bar{x}_{t+1:t+h}}{\lambda}\log\left(\frac{\bar{x}_{t+1:t+h}}{\lambda}\right) - \left(\frac{\bar{x}_{t+1:t+h}}{\lambda} - 1\right)\right\}.$$

The LR statistic is a function only of the expected count $h\lambda$ and the fitted intensity $\hat{\mu}_{t+1:t+h} := \bar{x}_{t+1:t+h}/\lambda$ of the interval $[t+1, t+h]$. Alternatively, it can be written as a function only of the expected count $h\lambda$ and the actual count $h\bar{x}_{t+1:t+h}$, which forms the fundamental basis for our algorithm.

In our application, thresholds for gamma ray burst detection are often set based on k-sigma events: values that are as extreme as observing a Gaussian observation that is k standard deviations above its mean. For our one-sided test, the distribution of the likelihood ratio statistic under the null is approximately a mixture of a point mass at 0 and a χ^2_1 distribution, each with probability 1/2 (Wilks 1938). We will call a k-sigma

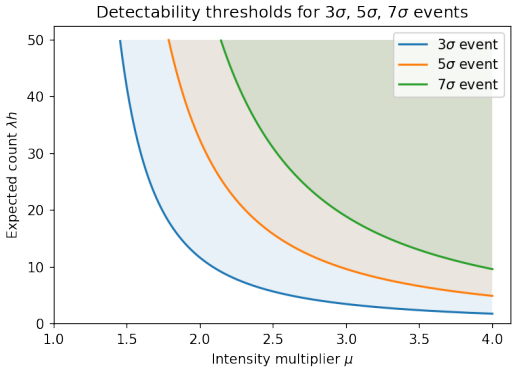


Figure 3.2.5: Detectability of GRBs at different k-sigma levels. Shaded regions show values of $h\lambda$ and $\hat{\mu}_{t+1:t+h}$ where the likelihood ratio exceeds k-sigma event thresholds for $k = 3$ (blue region), $k = 5$ (orange region) and $k = 7$ (green region) for a test that uses the correct value of h.

event one where the likelihood-ratio statistic exceeds a threshold of k^2.

The definition of a k-sigma event is based on a test for a single anomaly with a specific start and end point. In practice, detection of gamma ray bursts is achieved through performing multiple tests, allowing for different start and end points of the anomalies. Furthermore, different detection methods may perform more or fewer tests – which can make their statistical performance for the same k-sigma level different. We return to this in Section 3.5 where we present results that relate the k-sigma level for different methods to average run length, a common measure of type-I error rate in sequential testing.

Gamma ray burst detection and anomaly detection for other astrophysical events often works with a threshold of $k \approx 5$, a significantly higher statistical threshold than the $k = 3$ used in other areas of anomaly detection or the $k \approx 2$ used to reject a null hypothesis at the 5% level. This elevation of statistical threshold is required due to the multiple hypothesis testing problem outlined above. Where a specific threshold choice is required, we have presented the graphs and method assuming $k = 5$ in order to show impacts on our algorithm that may occur when working with a statistical threshold appropriate to the domain.

Gamma ray bursts with a combination of high intensity $\hat{\mu}_{t+1:t+h} := \frac{\bar{x}_{t+1:t+h}}{\lambda}$ and long length, as quantified by the expected count $h\lambda$, will have higher associated likelihood

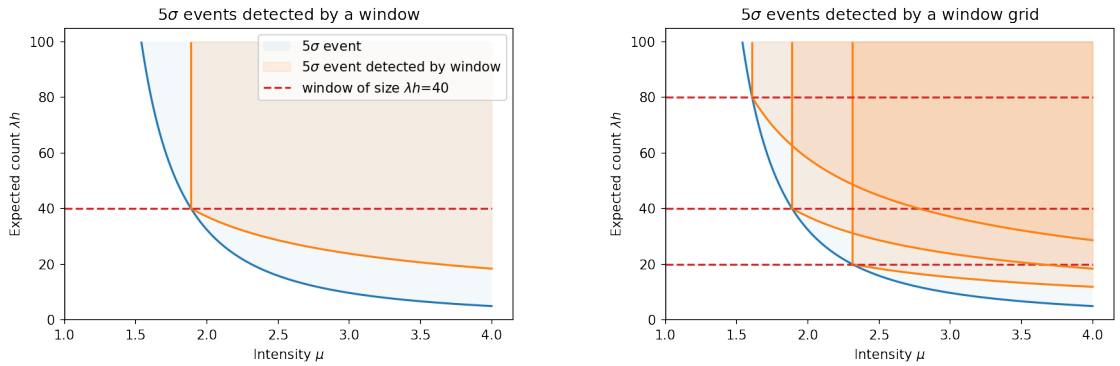


Figure 3.2.6: Detectability of GRBs by the window method, for one window (left) or a grid of three windows (right). The orange shaded area shows the values of $h\lambda$ and $\hat{\mu}_{t+1:t+h}$ where the likelihood ratio exceeds a 5-sigma threshold, and the blue shaded area shows the detectability region form Figure 3.2.5. Dashed lines show expected count $h\lambda$ over the window.

ratio statistics and thus be easier to detect. Figure 3.2.5 shows regions in this two-dimensional space that correspond to detectable GRBs at different k-sigma levels.

Checking every interval is not computationally possible in this setting due to the high signal velocity. For example, most GRBs last between 0.1s and 5 minutes (Kouveliotou et al. 1993). In order to find GRBs of 0.1s in length we may wish to use a fundamental bin width of 20ms, giving 15,000 intervals to check each time the signal updates and 750,000 intervals to check each second. This also means that more complex anomaly detection methods than that presented here that also iterate over all intervals up to a certain length, such as CAPA (Fisch, Bardwell, and Eckley 2022), are also computationally impossible in this setting.

Figure 3.2.6 shows what happens when we set a fixed threshold and for computational reasons only check intervals of certain lengths. We rely on the fact that a slightly brighter burst will also trigger detection on a longer or shorter interval than optimal. This is the type of approach that current window-based methods take (Paciesas et al. 2012). We can see that, even with a grid of window sizes, we lose detectability if the true width of the GRB does not match one of the window sizes.

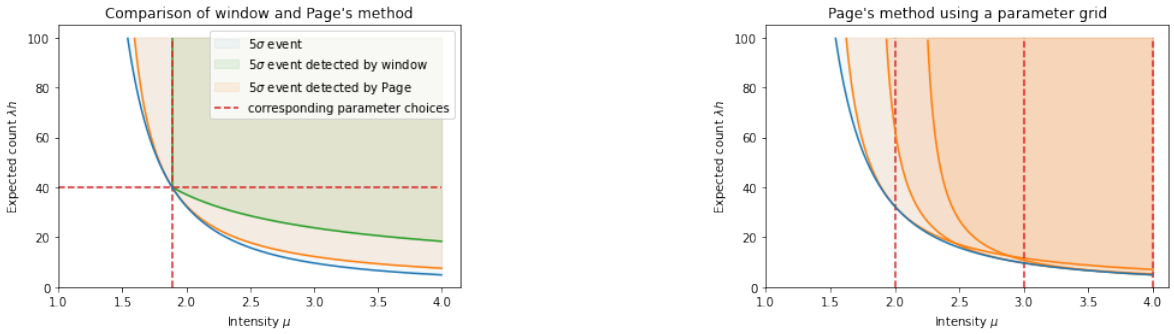


Figure 3.2.7: Detectability of GRBs by Page-CUSUM for a single μ value (left) and a grid of three μ values (right). Orange shaded area shows the values of $h\lambda$ and $\hat{\mu}_{t+1:t+h}$ where the likelihood ratio exceeds a 5-sigma threshold; the green shaded area shows the detectability region for the corresponding window test as defined by Proposition 3.2.2 (left-hand plot); and the blue shaded area shows the detectability region from Figure 3.2.5.

3.2.2 Page-CUSUM for Poisson data

As a foundation for our detection algorithm, consider the CUSUM (cumulative sum) approach of Page (1955) that was adapted for the Poisson setting by Lucas (1985). These methods can search for gamma ray bursts of unknown width but known size μ, differing from a window method that searches for gamma ray bursts of known width h and unknown size. To run our methods online it is useful to characterise a possible anomaly by its start point τ. We have our hypotheses for the signal at time T:

- $\mathbf{H}_0$: There have been no anomalies, i.e. $X_1, ..., X_T \sim \text{Poisson}(\lambda)$.
- $\mathbf{H}_1$: There has been one anomaly, beginning at some unknown time τ, with known intensity multiplier $\mu > 1$, i.e. $X_1, ..., X_{\tau-1} \sim \text{Poisson}(\lambda)$ and $X_\tau, ..., X_T \sim \text{Poisson}(\mu\lambda)$.

Our LR statistic for this test is 0 if $\bar{x}_{\tau:T} \leq \lambda \frac{\mu-1}{\log(\mu)}$ for all τ, otherwise

$$LR = \max_{1 \leq \tau \leq T} \left[2(T - \tau + 1)\lambda \left\{ \frac{\bar{x}_{\tau:T}}{\lambda} \log(\mu) - (\mu - 1) \right\} \right].$$

We work with a test statistic, S_T, that is half the likelihood ratio statistic for this test, and compare to a k-sigma threshold of $k^2/2$. S_T can be rewritten in the following

form:

$$S_T = \left[\max_{1 \leq \tau \leq T} \sum_{t=\tau}^{T} (x_t \log(\mu) - \lambda(\mu - 1)) \right]^+,$$

where we use the notation $[\cdot]^+$ to denote the maximum of the term $\cdot$ and 0. As shown in Lucas (1985), S_T can be updated recursively as

$$S_0 = 0, \quad S_{T+1} = [S_T + x_{T+1} \log \mu - \lambda(\mu - 1)]^+.$$

It is helpful to compare the detectability of GRBs using S_T with their detectability using a window method. To this end, we introduce the following propositions (see related results for Normally distributed data in Basseville and Nikiforov 1993; Romano, Eckley, Fearnhead, and Rigaill 2023).

Proposition 3.2.1. *For some choice of μ against a background rate of λ, let S_T be significant at the k-sigma level. Then there exists some interval $[\tau, T]$ with associated likelihood ratio statistic that is significant at the k-sigma level.*

Proposition 3.2.2. *For any k, λ and h there exists a μ and corresponding test statistic, S_T, that relates directly to a window test of length h, and background rate λ as follows: if, for any t, the data $x_{t+1:t+h}$ is significant at the k-sigma level then S_{t+h} will also be significant at the k-sigma level.*

Proofs can be found in Appendix A.2.

Together these results show that Page's method is at least as powerful as the window method for detecting a GRB at a fixed k-sigma level. Rather than implementing the window method with a given window size, we can implement Page's method with the appropriate μ value (as defined by Proposition 3.2.2) such that any GRB detected by the window method would be detected by Page's method. However Page's method may detect additional GRBs and these would be detected by the window method with some window size (by Proposition 3.2.1). In practice, as shown in Figure 3.2.7, Page's method provides better coverage of the search space.

Whilst the Page-CUSUM approach is more powerful than a window-based approach,

to cover the space completely it still requires specifying a grid of values for the intensity of the gamma ray burst. If the actual intensity lies far from our grid values we will lose power at detecting the burst.

3.3 Functional pruning

To look for an anomalous excess of count of any intensity and width without having to pick a parameter grid, we consider computing the Page-CUSUM statistic simultaneously for all $\mu \in [1, \infty)$. This can be achieved by considering the test statistic as a function of μ, $S_T(\mu)$. That is, for each T, $S_T(\mu)$ is defined for $\mu \in [1, \infty)$, and for a given μ is equal to the value of the Page-CUSUM statistic for that μ.

By definition, $S_T(\mu)$ is a pointwise maximum of curves representing all possible anomaly start points τ:

$$S_T(\mu) := \left[\max_{1 \leq \tau \leq T} \sum_{t=\tau}^{T} [x_t \log(\mu) - \lambda(\mu - 1)] \right]^+$$

We can view this as $S_T(\mu) = [\max_{1 \leq \tau \leq T} C_\tau^{(T)}(\mu)]^+$, where each curve, $C_\tau^{(T)}(\mu)$, corresponds to half the likelihood ratio statistic for a gamma ray burst of intensity μ starting at τ,

$$C_\tau^{(T)}(\mu) := \sum_{t=\tau}^{T} [x_t \log(\mu) - \lambda(\mu - 1)].$$

Each curve is parameterised by two quantities, as

$$C_\tau^{(T)}(\mu) := a_\tau^{(T)} \log(\mu) - b_\tau^{(T)}(\mu - 1),$$

where $a_\tau^{(T)} = \sum_{t=\tau}^{T} x_t$ is the actual observed count and $b_\tau^{(T)} = \sum_{t=\tau}^{T} \lambda = \lambda(T - \tau + 1)$ is the expected count on the interval $[\tau, T]$. As we move from time T to time $T + 1$ there is a simple recursion to update these coefficients: $a_\tau^{(T+1)} = a_\tau^{(T)} + x_{T+1}$, $b_\tau^{(T+1)} = b_\tau^{(T)} + \lambda$. These are linear and do not depend on τ, so the differences between any two curves are preserved with time updates.

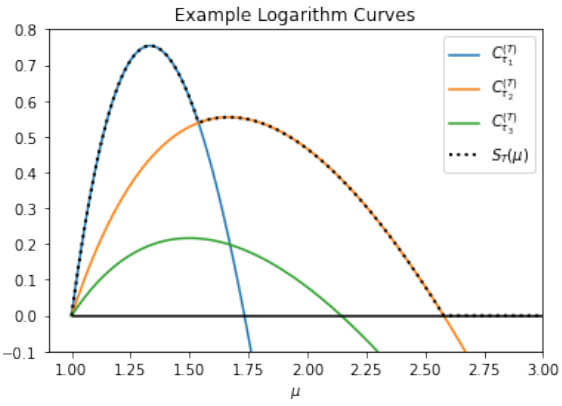


Figure 3.3.8: Three example logarithmic curves. The statistic $S_T(\mu)$ is defined as the maximum of all logarithmic curves and the 0 line.

We call $C_\tau^{(T)}(\mu)$ a logarithmic curve. Figure 3.3.8 shows examples of these logarithmic curves. The maximum of $C_\tau^{(T)}$ is located at $\mu = a_\tau^{(T)}/b_\tau^{(T)}$, representing the likelihood ratio for a post-change mean $\mu\lambda = \bar{x}_{\tau:T}$. If and only if $a_\tau^{(T)} > b_\tau^{(T)}$ then the logarithmic curve will be positive for some $\mu > 1$. In this case we will define the root of the curve to be the unique $\mu^* > 1$ such that $C_\tau^{(T)}(\mu^*) = 0$.

3.3.1 Adding and pruning curves

For any two curves $C_{\tau_i}^{(T)}$ and $C_{\tau_j}^{(T)}$ at a given present time T, we will say that $C_{\tau_i}^{(T)}$ dominates $C_{\tau_j}^{(T)}$ if

$$[C_{\tau_i}^{(T)}(\mu)]^+ \geq [C_{\tau_j}^{(T)}(\mu)]^+, \quad \forall \mu \in [1, \infty).$$

This is equivalent to saying that there is no value of μ such that the interval $[\tau_j, T]$ provides better evidence for an anomaly with intensity μ than $[\tau_i, T]$. As the difference between curves is unchanged as we observe more data, this in turn means that for any future point $T_F \geq T$, the interval $[\tau_j, T_F]$ will not provide better evidence than $[\tau_i, T_F]$. Therefore, the curve associated with τ_j can be pruned, i.e., removed from our computational characterisation of $S_T(\mu)$.

The following gives necessary and sufficient conditions for one curve to be dominated by another.

Proposition 3.3.1. *Let $C_{\tau_i}^{(T)}$ and $C_{\tau_j}^{(T)}$ be curves that are positive somewhere on $\mu \in$*

$[1, \infty)$, *where* $\tau_i < \tau_j$ *and* $C_{\tau_i}^{(\tau_j - 1)}$ *is also positive somewhere on* $\mu \in [1, \infty)$. *Then* $C_{\tau_i}^{(T)}$ *dominates* $C_{\tau_j}^{(T)}$ *if and only if* $a_{\tau_j}^{(T)} / b_{\tau_j}^{(T)} \leq a_{\tau_i}^{(\tau_j - 1)} / b_{\tau_i}^{(\tau_j - 1)}$ *or equivalently* $a_{\tau_j}^{(T)} / b_{\tau_j}^{(T)} \leq a_{\tau_i}^{(T)} / b_{\tau_i}^{(T)}$. *Additionally, it cannot be the case that* $C_{\tau_j}^{(T)}$ *dominates* $C_{\tau_i}^{(T)}$.

A formal proof is given in Appendix A.2.2, but we see why this result holds by looking at Figure 3.3.8. There we see that $C_{\tau_i}^{(T)}$ dominates $C_{\tau_j}^{(T)}$ precisely when $C_{\tau_i}^{(T)}$ has both a greater slope at $\mu = 1$ (which occurs when $C_{\tau_i}^{(\tau_j - 1)}$ is positive) and a greater root than $C_{\tau_j}^{(T)}$, where (as shown in the proof) the root of a curve $C_{\tau}^{(T)}$ is an increasing function of $a_{\tau}^{(T)} / b_{\tau}^{(T)}$.

3.4 Algorithm and theoretical evaluation

Using Proposition 3.3.1 we obtain the Poisson-FOCuS algorithm, described in Algorithm 1. This algorithm stores a list of curves in time order by storing their associated a and b parameters, as well as their times of creation τ, which for the constant λ case can be computed as $T + 1 - b/\lambda$.

On receiving a new observation at time T, these parameters are updated. If the observed count exceeds the expected count predicted by the most recent curve we also add a new curve which corresponds to a GRB that starts at time T. Otherwise we check to see if we can prune the most recent curve. This pruning step uses Proposition 3.3.1, which shows that if any currently stored curve can be pruned, the most recently stored curve will be able to be pruned. (Our pruning check does not does not need to be repeated for additional curves, as on average less than one curve is pruned at each timestep.)

The final part of the algorithm is to find the maximum of each curve, and check if the maximum of these is greater than the threshold. If it is, then we have detected a GRB. The start of the detected GRB is given by the time that the curve with the largest maximum value was added.

Algorithm 1: Poisson-FOCuS for constant λ

Result: Startpoint, endpoint and test-statistic value for anomaly detected at a k-sigma threshold.

set threshold $k^2/2$;
initialise empty curve list;
while *anomaly not yet found* **do**
  $T \leftarrow T + 1$;
  get actual count X_T;
  get expected count λ;
  `// update curves:`
  for *curve* $C_{\tau_i}^{(T-1)}$ *in curve list* $[C_{\tau_1}^{(T-1)}, ..., C_{\tau_n}^{(T-1)}]$ **do**
    $a_{\tau_i}^{(T)} \leftarrow a_{\tau_i}^{(T-1)} + X_T$; $b_{\tau_i}^{(T)} \leftarrow b_{\tau_i}^{(T-1)} + \lambda$;
  end
  `// add or prune curve:`
  if $X_T/\lambda > \max[a_{\tau_n}^{(T)}/b_{\tau_n}^{(T)}, 1]$ **then**
    add $C_T^{(T)} : a_T^{(T)} = X_T, b_T^{(T)} = \lambda, \tau = T$ to curve list;
  else if $a_{\tau_n}^{(T)}/b_{\tau_n}^{(T)} < \max[a_{\tau_{n-1}}^{(T)}/b_{\tau_{n-1}}^{(T)}, 1]$ **then**
    remove $C_{\tau_n}^{(T)}$ from curve list;
  end
  `// calculate maximum` M`:`
  for *curve* $C_{\tau_i}^{(T)}$ *in curve list* **do**
    if $\max(C_{\tau_i}^{(T)}) > M$ **then**
      $M \leftarrow \max(C_{\tau_i}^{(T)})$;
      $\tau^* \leftarrow \tau_i$
    end
  end
  if $M > k^2/2$ **then**
    anomaly found on interval $[\tau^*, T]$ with $\sqrt{2M} > k$;
  end
end

3.4.1 Dealing with varying background rate

Algorithm 1 deals with the constant λ case. If $\lambda = \lambda(t)$ is not constant, but an estimate λ_T of $\lambda(T)$ is available at each timestep T, we can apply the same principle but with a change in the definition of $b_\tau^{(T)}$. We now have $b_\tau^{(T)} := \sum_{t=\tau}^{T} \lambda_t$, the total expected count over the interval $[\tau, T]$. For the algorithm, this only impacts how the coefficients are updated, with the new updates being: $a_\tau^{(T+1)} \leftarrow a_\tau^{(T)} + X_{T+1}$, $b_\tau^{(T+1)} \leftarrow b_\tau^{(T)} + \lambda_{T+1}$. If we work with a non-homogeneous Poisson process in this way, it is impossible to recover τ from the coefficients $a_\tau^{(T)}$ and $b_\tau^{(T)}$, so $C_\tau^{(T)}$ must be stored as the triplet $(\tau, a_\tau^{(T)}, b_\tau^{(T)})$.

The Poisson-FOCuS algorithm gives us an estimate of the start point of a GRB by reporting the interval $[\tau^*, T]$ over which an anomaly is identified. In our application, if the additional sanity checking indicates a GRB is present, the whole signal starting some time before τ^* to some time after T is then transmitted from the satellite to Earth. After this has occurred, Poisson-FOCuS can restart immediately provided that a good background rate estimate is available.

3.4.2 Minimum anomaly intensity

For our application there is an upper limit on the length of a gamma ray burst, and it makes sense to ensure we do not detect gamma ray bursts that are longer than this. To do so, we set an appropriate $\mu_{\min}$, and additionally prune curves which only contribute to $S_T(\mu)$ on $1 < \mu < \mu_{\min}$, by removing, or not adding, curves $C_\tau^{(T)}$ to the list if $C_\tau^{(T)}(\mu_{\min}) \leq 0$, i.e.

$$\frac{a_\tau^{(T)}}{b_\tau^{(T)}} \leq \frac{\mu_{\min} - 1}{\log \mu_{\min}}.$$

We can choose $\mu_{\min}$ according to our threshold and the maximum expected time period we are interested in searching for bursts over, using the proof of Proposition 3.2.2 about detectability for the window-based method, as follows:

$$(h\lambda)_{\max} = \frac{k^2}{2[\mu_{\min} \log(\mu_{\min}) - (\mu_{\min} - 1)]}.$$

For a 5-sigma threshold, assuming a background rate of one photon every 500μs, a maximum length of 1 minute for a GRB would correspond to $\mu_{\min} = 1.015$.

3.4.3 Using time-to-arrival data

Rather than taking as data the number of photons observed in each time window, we can take as our data the time between each observation. In this case our data is $U_1, U_2, \ldots$ where U_i is the time between the $(i+1)$th and ith photons. Under the assumption the data follows a Poisson process, we have that the U_i are independently Exponentially distributed.

The Poisson-FOCuS algorithm still works in the Exponential case, with the only difference being how we update the coefficients of the curves.

$$a_\tau^{(T+1)} = a_\tau^{(T)} + 1, \quad b_\tau^{(T+1)} = b_\tau^{(T)} + \lambda_{T+1} U_{T+1},$$

where λ_T is the estimate of the background rate at the time of the Tth photon arrival.

In our application there is a high velocity of photon arrivals, and a GRB will consist of a large (> 50) number of photons. The additional computation required to process individual photons and the false positive rate introduced by considering very short time intervals render this method not as computationally or statistically as effective as binning the data. However, it may be useful in applications where anomalies can consist of smaller numbers of counts.

3.4.4 Computational cost comparisons

Using a window method, the computational cost per window consists of: adding x_T and λ_T to the window; removing x_{T-h} and λ_{t-h} from the window; calculating the test statistic and comparing to the threshold. Using Poisson-FOCuS, our computational cost per curve consists of: adding x_T to $a_\tau^{(T)}$; adding λ_T to $b_\tau^{(T)}$; calculating the maximum of the curve and comparing to the threshold. The computational cost per curve is therefore roughly equal to the computational cost per window. Thus when evaluating

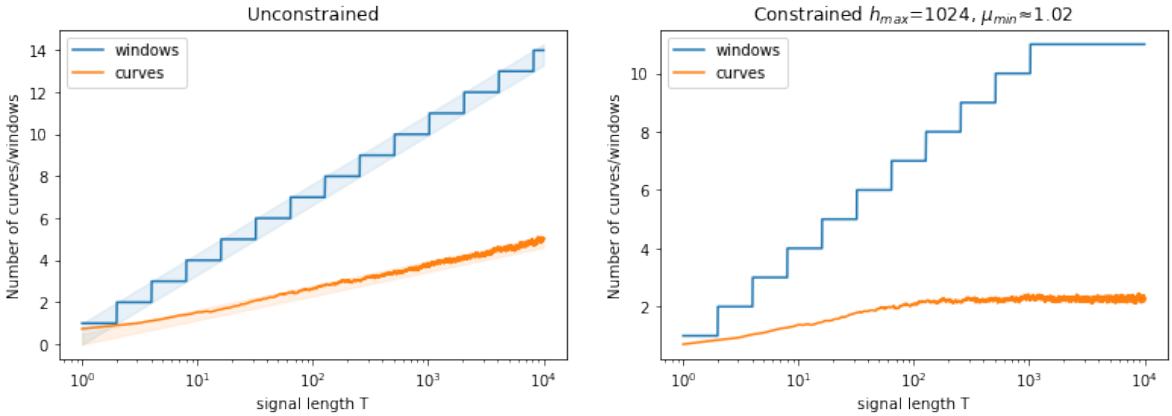


Figure 3.4.9: Comparisons of the number of windows and expected number of curves (average over 1000 runs) kept by FOCuS running over a signal with base rate $\lambda = 100$ using a 5-sigma threshold, with on constraint on length of GRB (left) and with $h_{\max} = 1024$, corresponding to $\mu_{\min} = 1.02$ (right).

the relative computational cost of Poisson-FOCuS versus a window method we need only calculate the expected number of curves kept by the algorithm at each timestep, and compare against the number of windows used. We now give mathematical bounds on this quantity, as follows:

Proposition 3.4.1. *The expected number of curves kept by Poisson-FOCuS without* μ_{min} *at each timestep* T *is* $\in \left[\frac{\log(T)}{2}, \frac{\log(T)+1}{2}\right]$.

Proposition 3.4.2. *The expected number of curves kept by Poisson-FOCuS using some* $\mu_{min} > 1$ *at each timestep is bounded.*

Proofs for both propositions are in Appendix A.2.4. Chapter 6 Section 6.2.1 also contains further work beyond the scope of this chapter to compute the bound given by Proposition 3.4.2 at both each timestep and overall.

For geometrically spaced windows, over an infinite horizon the number of windows used at each timestep T is $\in [\log_2(T), \log_2(T)+1]$, and if a $h_{\max}$ is implemented then this will be bounded after a certain point. Figure 3.4.9 gives a comparison of the number of windows and expected number of curves, showing that although the bound from Proposition 3.4.2 is difficult to calculate, it is substantially below the corresponding bound on the number of windows. Therefore, Poisson-FOCuS provides the statistical

advantages of an exhaustive window search at under half the computational cost of a geometrically spaced one.

3.5 Empirical evaluation

3.5.1 Simulations of GRBs and average run length comparison

We have previously worked under the assumption that a threshold level of 5 sigma will be used. In practice, the threshold is a tuneable parameter and will be chosen to give a trade-off between detection sensitivity and number of false detections. We can make a direct comparison between the sigma threshold and average run length, a standard metric in anomaly detection literature. The average run length is the expected time until we have a false detection if we simulate data under the null hypothesis. This can be calculated for both FOCuS and logarithmically spaced windows, letting us make comparisons between the two by choosing a slightly different sigma threshold based on equal average run lengths. The average run length for each algorithm does not easily translate into an average run length for detecting GRBs, as it does not account for the sanity checking step of the detection algorithm, which can require a GRB to be detected by multiple detectors; nor do the results we present account for any error in estimating the background rate.

The results are shown in Figure 3.5.10. The run length is given in terms of number of observations, so would need to be multiplied by the choice of fundamental bin width, w, to be converted to time. The main message from these results is that for the same average run length we would need to have a threshold that is 0.1σ higher for FOCuS than for the methods that uses logarithmically spaced windows.

We now compare Poisson-FOCuS with a window based method on synthetic data that has been simulated to mimic known GRBs, but allowing for different intensities of burst. To simulate the data for a chosen known GRB at a range of different brightnesses, the photon stream of the GRB was converted into a random variable via density estimation. One draw from this random variable would give a photon impact time, and

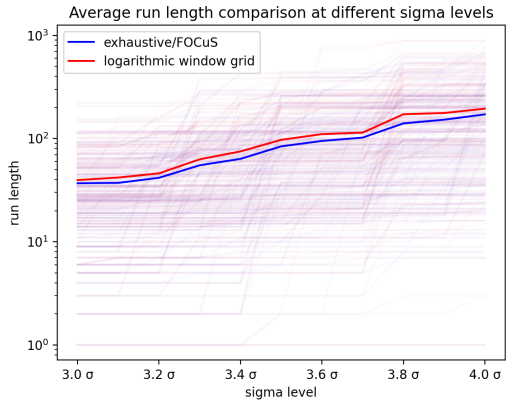


Figure 3.5.10: Comparison between FOCuS and logarithmic window method showing the average run length at different sigma levels.

n independent draws sorted into time order would give a stream of photon impact times that well approximate the shape of the burst. These were then overlaid on a background photon stream to form a signal that was binned into fundamental time widths of 50ms, which was fed into either Poisson-FOCuS or a geometrically spaced window search. The maximum sigma-level that was recorded when passing over the signal with each method is then plotted for various different brightnesses n. To stabilise any randomness introduced by the use of a random variable for GRB shape, this was repeated 10 times with different random seeds common to both methods and the average sigma-level is plotted.

The extent to which Poisson-FOCuS provides an improvement in detection power depends on the size and shape of the burst, and in particular whether the most promising interval in the burst lines up well with the geometrically spaced window grid. For example, the burst illustrated in Figure 3.5.11c does not line up with this grid, and Figure 3.5.11d shows how Poisson-FOCuS provides an improvement in detection power of approximately 0.5σ for this shape of burst at various different brightnesses, far higher than the approximately 0.1σ increase in threshold required to give a similar average run length (see Figure 3.5.10). However, the shorter burst in Figure 3.5.11e clearly has a most promising interval of size 1 for this binning choice, which is covered exactly by the logarithmic window grid. Therefore, Poisson-FOCuS provides no improvement over

the window grid.

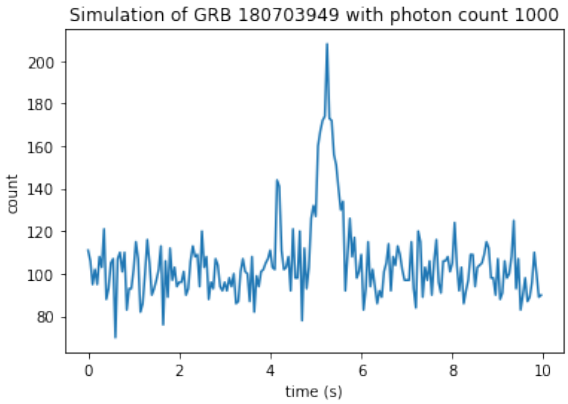
3.5.2 Bias from estimating background rate

When using Poisson-FOCuS on a dataset requiring background rate estimation, particular care needs to be taken with the choice of the estimator for the background rate. Small, consistent under-estimation of the background rate will be recorded as an anomaly over a long timescale. The ability of any online algorithm to give immediate detections requires a background estimate for time T to be available using only data from $t \leq T$, and this can cause challenges with avoiding underestimating the background rate in periods where it is increasing. Furthermore, the presence of a GRB within the data being used to estimate background rate could destabilise the estimation, so robust methods are needed.

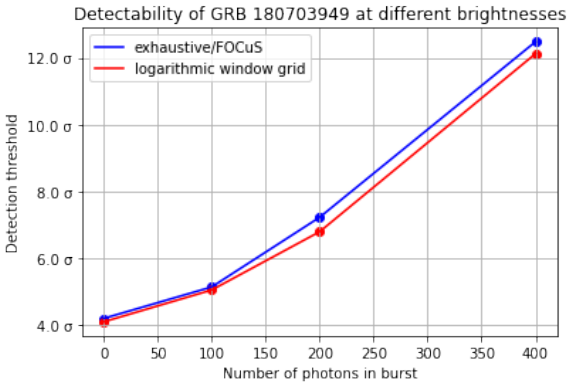
To show this effect, Figure 3.5.12 shows a portion of the same data as Figure 3.1.2 at a higher resolution. This hour of data was chosen because it contains a smooth rise in background rate, which can give rise to anomalies when using a biased background rate estimation method.

Figure 3.5.13 shows the sigma thresholds recorded by Poisson-FOCuS running over this hour's worth of higher-resolution data using a background estimate of 3 minute centered moving-average window (Figure 3.5.13a), and a 3 minute uncentered moving-average window such that background estimates at time T use only data from $t \leq T$ (Figure 3.5.13b). The uncentered method has a large peak in the recorded statistical threshold as it passes over the signal. This is caused by the upward change in background rate, which results in a consistent underestimation of the background rate at time T when using data from just the period prior to T. This bias in background estimation is then interpreted as a very small anomaly over a very long time period.

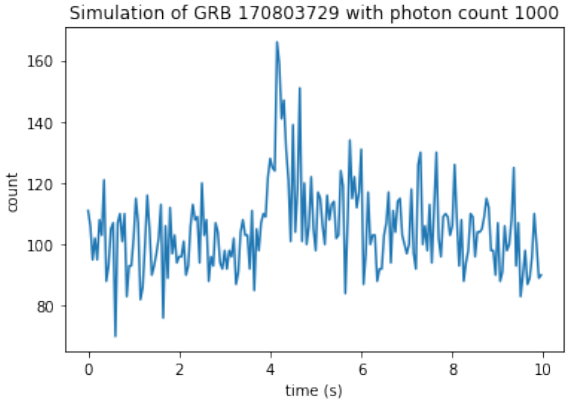
While the effect of a biased background estimation method can be somewhat countered by setting a value for $\mu_{\min} > 1$ as in Figure 3.5.13c, careful consideration should be given to de-biasing the background estimation method in order to avoid false detections when the background rate is rising (see e.g. Crupi, Ward, et al. 2023). For



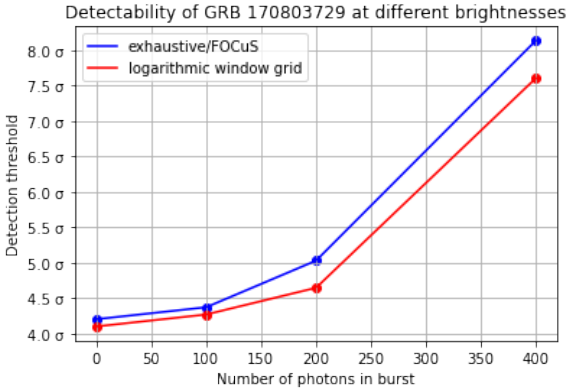
(a)



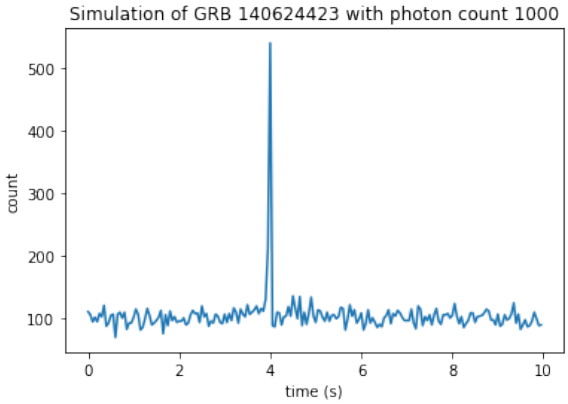
(b)



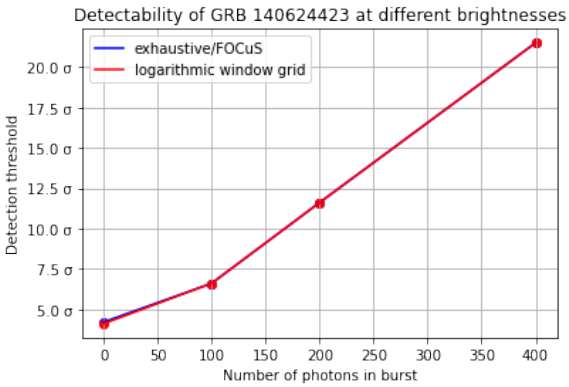
(c)



(d)



(e)



(f)

Figure 3.5.11: Plots of runs of FOCuS over simulated GRB copies of different brightnesses

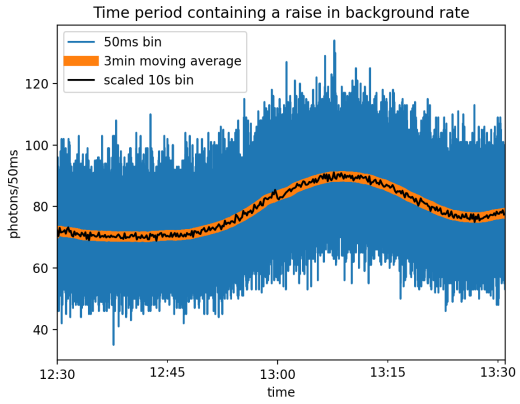
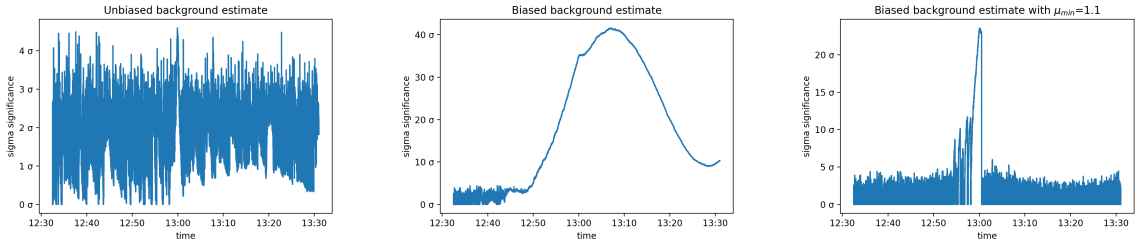


Figure 3.5.12: An hour's portion of the same data from Figure 3.1.2 at higher resolution of 50ms (blue). In black is the data at 10s resolution identical to that from Figure 3.1.2 but rescaled by 0.005x to fit the graph. In orange is a centered 3 minute moving-average background estimate (linewidth increased for visual clarity).



(a) (b) (c)

Figure 3.5.13: Plots of a run of FOCuS over the data using various background estimation methods and parameters.

example, we show in the next section that reliable detection of GRBs can be obtained using an exponential smoother to estimate the background rate.

3.5.3 Application to FERMI data

In the context of HERMES, Poisson-FOCuS is currently being employed for two different purposes. First, a trigger algorithm built using Poisson-FOCuS is being developed for on-board, online GRB detection. To date, a dummy implementation has been developed and preliminary testings performed on the HERMES payload data handling unit computer. Second, Poisson-FOCuS is being employed in a software framework intended to serve as the foundation for the HERMES offline data analysis pipeline (Crupi, Ward, et al. 2023). In this framework, background reference estimates are provided by

a neural network as a function of the satellite's current location and orientation.

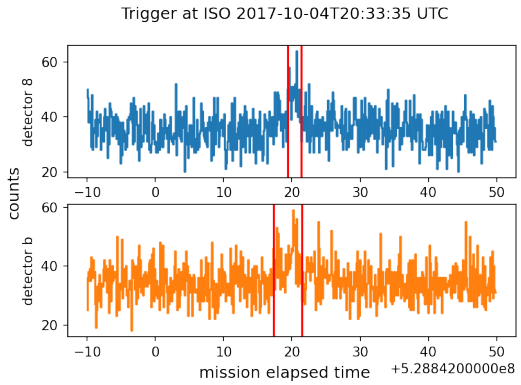
Since no HERMES cube satellites have been launched yet, testing has taken place over Fermi gamma ray burst monitor (GBM) archival data, looking for events which may have evaded the on-board trigger algorithm. The data used for the analysis were drawn from the Fermi GBM daily data, Fermi GBM trigger catalogue, and Fermi GBM untriggered burst candidates catalogue, all of which are publically available at NASA's High Energy Astrophysics Science Archive Research Center (HEASARC 2022a; HEASARC 2022b; Bhat 2021).

The algorithm was run over eight days of data, from 00:00:00 2017/10/02 to 23:59:59 2017/10/09 UTC time. This particular time frame was selected because, according to the untriggered GBM Short GRB candidates catalog, it hosts two highly reliable short GRB candidates which defied the Fermi-GBM online trigger algorithm. During this week the Fermi GBM algorithm was triggered by 11 different events. Six of these were classified as GRBs, three as terrestrial gamma ray flashes, one as a local particle event and one as an uncertain event. The Poisson-FOCuS algorithm was run over data streams from 12 sodium iodide GBM detectors in the energy range of $50 - 300$ kiloelectron volts, which is most relevant to GRB detection but excludes the bismuth germanate detectors and higher energy ranges designed to find terrestrial gamma ray flashes.

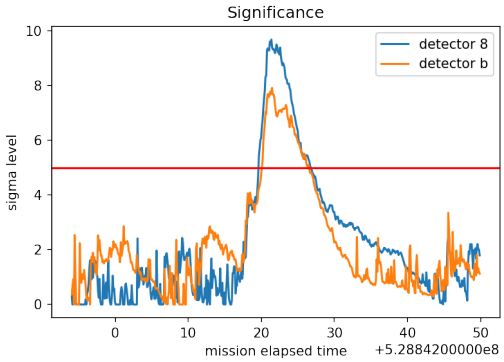
The data was binned at 100ms. Background count-rates were assessed by exponential smoothing of past observations, excluding the most recent 4s, and any curves corresponding to start points older than 4s were automatically removed from the curve lists. The returning condition used was the same used by Fermi-GBM: a trigger is issued whenever at least two detectors are simultaneously above threshold. After a trigger, the algorithm was kept idle for five minutes and then restarted.

At a 5-sigma threshold, Poisson-FOCuS was able to identify all the six GRBs which also triggered the Fermi-GBM algorithm, one of which is shown in Figures 3.5.14a and 3.5.14b. We also observed a trigger compatible with an event in the untriggered GBM Short GRB candidates catalog (Bhat 2021), which is shown in Figures 3.5.14c and

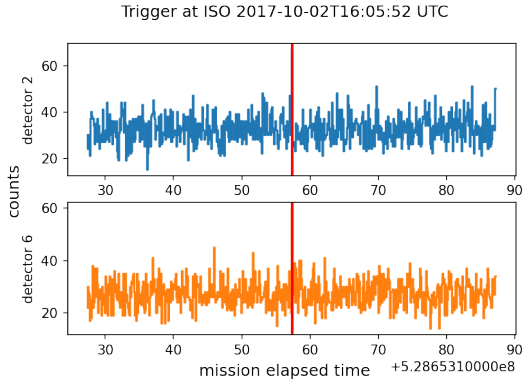
3.5.14d. An uncertain event not in either catalogue is shown in Figures 3.5.14e and 3.5.14f, which may indicate a GRB that had been missed by earlier searches or may be a false positive.



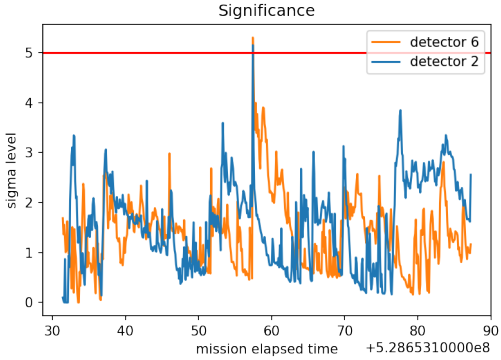
(a) known GRB from Fermi catalogue



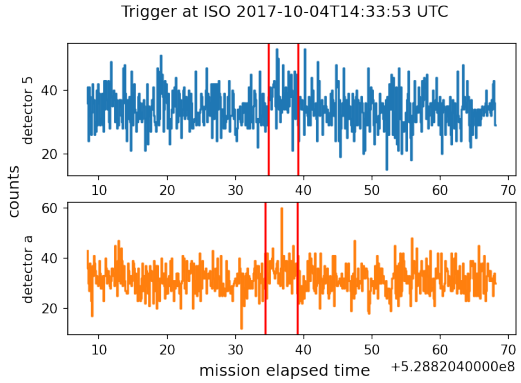
(b) significance from Poisson-FOCuS



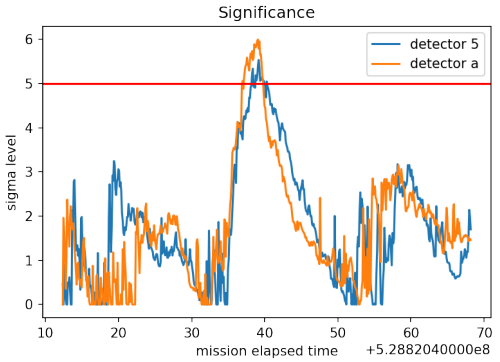
(c) GRB in short candidates catalogue



(d) significance from Poisson-FOCuS



(e) possible GRB not in either catalogue



(f) significance from Poisson-FOCuS

Figure 3.5.14: Three of the triggers found in the FERMI daily data. Left-hand column shows data from the two detectors that give a trigger, and the right-hand column shows the corresponding output from the Poisson-FOCuS algorithm.

3.6 Discussion

The main purpose of this paper was to present a GRB detection algorithm that is mathematically equivalent to searching all possible window lengths, while requiring less computational power than the grid of windows approach. This algorithm is suitable for use on the HERMES satellites, where it has lead to a reduction of required computations in a very computationally constrained setting, as well as a simplification of parameter choices by practitioners as values for window lengths in a grid no longer need to be specified.

There is increasing interest in detecting anomalies in other low-compute settings, for example Internet of Things sensors which must continuously monitor a signal (Dey et al. 2018). These may have a limited battery life or limited electricity generation from sensor-mounted solar panels. Therefore, the algorithm we have developed may be of use more widely.

Much of the mathematical work presented in this paper is also applicable to the $\mu \in [0, 1]$ case that searches for an anomalous lack of count in a signal. When adapting Poisson-FOCuS to this setting, it is important to make sure the algorithm functions well in situations where the counts are small, as these are precisely the locations of anomalies. Combining these two cases would give a general algorithm for detection of anomalies on $\mu \in [0, \infty)$.

Code for Poisson-FOCuS and the analysis for this paper is available at the GitHub repository `https://github.com/kesward/FOCuS`

3.7 Impact from this research

After six years of development and qualification, the six HERMES scientific pathfinder nanosatellites launched into low-Earth orbit in February 2025. Poisson-FOCuS was not the on-board algorithm, but will be the centrepiece of the HERMES offline data pipeline for burst search. This software, developed jointly by the Italian Space Agency-Space Science Data Center and the Italian National Astrophysics Institute, will search

for gamma ray bursts and other astrophysical transients as data are downlinked from spacecrafts to ground stations. The pipeline is based on the Poisson-FOCuS implementation described in Dilillo, Ward, et al. (2024), adapted to the HERMES instrument design. We hope the successful application of this pipeline will demonstrate the capabilities of Poisson-FOCuS and open a path for its implementation onboard future iterations of the HERMES constellation and other high-energy astrophysical transient detection instruments.

Chapter 4

Poisson-FOCuS for nuclear radiation monitoring

This chapter is a reproduction of a technical report developed for the Nuclear Security Science Network. In this report, we describe the use of Poisson-FOCuS for detecting changes in radiation counts in the SIGMA dataset. The report is written for the point of view of a practitioner interested in use of the FOCuS algorithm. It is presented in this thesis as an example of industrial engagement and the impact it is possible to achieve by adapting existing methodological research to novel application settings.

4.1 Introduction

The Poisson Functional Online Cumulative Sum (Poisson-FOCuS) method is a method for solving the likelihood ratio test of Poisson(λ) null against Poisson($\mu\lambda$) alternative where $\mu > 1$, i.e. searching for an increase in count. This can be thought of as equivalent to testing all possible anomaly start points $\tau \leq T$ at each timestep T, giving a computationally efficient way to analyse count anomalies that occur over intervals of time. We run the Poisson-FOCuS method on SIGMA data used for nuclear radiation monitoring, with an additional adjustment to remove anomaly tail traces, and report the results.

4.2 Data description

The SIGMA data consists of gamma radiation (high energy photon impacts) on radiation detectors placed at different locations around London, UK. We want to design a system to monitor this data in real-time and search for threat profiles, for example a person smuggling a backpack of material with a Uranium-235 signature. At the same time, we want to be able to identify and discount anomalies in the data that are not threat profiles, such as a patient leaving a hospital after a radionuclide thyroid scan with an iodine-123 signature. The SIGMA data is from multiple different sensors, however as they are located far from each other it is assumed that any radiation threat profile would only show up in one sensor at a time.

The data exists in 4096 energy band bins. The file containing each day's worth of data is separated into approximately 55800 time bins, giving a sample rate of 1 bin per second. There are approximately four months' worth of data, from 2018-08-06 until 2018-12-18 (135 days). This gives a total of approximately 31 billion total data bins (time by energy band) per sensor. In our analysis we will primarily consider data from one sensor. All data plots are taken from the 6th and 14th August, 2018.

4.3 Problem setup

Our data signal $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_T, ...)$ is a multivariate signal evolving through time. Each $\vec{x}_t := (x_t^1, ..., x_t^p)$ is a p-dimensional object, which represents the energy spectrum (for our data $p = 4096$).

We denote by T the present time, such that at time T only the signal $\vec{x}_t : t \leq T$ has been observed. We are interested in algorithms that perform well when $T \to \infty$, i.e. we have been observing a signal for a long time, or the signal is high-velocity. In the data we have, we assume batch processing each day's worth of data independently with $T \to 86400$ and this computation repeated up to 135 times to process the whole available dataset.

An anomaly with start time τ affecting some subset $P \subset \{1, ..., p\}$ of coordinates

is such that for $t > \tau, i \in P$ there has been a change in the underlying process used to produce the measurements. We want to identify τ and P at the soonest possible point $T > \tau$ we are able to observe sufficient evidence. We are also only interested in anomalies where their length $h := T - \tau + 1$ is relatively short, e.g. $h \leq 300$ (up to five minutes). We are only interested in anomalous increases in count, rather than decreases.

We will consider each x_t^i to be the realisation of a random variable X_t^i. As radiation counts can be modelled well as a Poisson process, we will say that under our null hypothesis of no anomaly each $X_t^i \sim \text{Poisson}(\lambda^i)$. We can estimate the λ_t^i well using previous data, noting that in the absence of anomalies our data stream does not change much over time.

By additivity of Poisson processes, we have that

$$\sum_{i \in P} X_t^i \sim \text{Poisson}\left(\sum_{i \in P} \lambda^i\right).$$

Initially we will work with $P := \{1, ..., 4096\}$ the whole signal trace and define $x_t := \sum_{i=1}^{4096} x_t^i$ and $\lambda := \sum_{i=1}^{4096} \lambda^i$, noting that we can estimate $\lambda \approx 28$ although there are a few mild fluctuations in the data. In section 4.8, we will outline ways of defining subsets that may be useful for detecting the radiation signatures of different isotopes.

4.4 Theory and method

4.4.1 Likelihood ratio testing

In general our significance (how surprised we are) is a function only of what we expect to see, and what we actually see.

$$\text{significance} = f\ (\text{expected count}, \text{actual count})$$

When working with count data, we use Poisson random variables. Denoting the actual count x_t and the expected count λ, we have

$$\text{significance p-value} = \mathbb{P}(\text{Poisson}(\lambda) \geq x_t)$$

However, this can be computationally inefficient. Therefore we use Wilks theorem (Wilks 1938) to approximate twice the Poisson log-likelihood ratio by a χ^2_1 random variable. This is a very accurate approximation for any appreciable value of λ, certainly so for our problem. We have that

$$\frac{(\text{significance sigma-value})^2}{2} = x_t \log\left(\frac{x_t}{\lambda}\right) + \lambda\left(\frac{x_t}{\lambda} - 1\right)$$

We use thresholds of 4.5 for a 3-sigma event, 12.5 for a 5-sigma event, etc. In general, a k-sigma event needs a threshold of $k^2/2$.

To find expected and actual counts for an interval $[\tau, T]$, you just add up the actual counts $\sum_{t=\tau}^{T} x_t$ and expected counts $\lambda(T - \tau + 1)$ for each time point in the interval, and use these in the above method to calculate your significance statistic. However, a signal of length T generates $T+1$ new intervals when a new point is added. Even if we only check the final h intervals, this can be computationally costly.

4.4.2 Page and FOCuS

The Functional Online Cumulative Sum (FOCuS) method (Ward, Dilillo, et al. 2023) is a quick method for solving the likelihood ratio test of Poisson(λ) null against Poisson($\mu\lambda$) alternative where $\mu > 1$, i.e. searching for an increase in count. This can be thought of as equivalent to testing all possible anomaly start points $\tau \leq T$ at each timestep T. Imposing a constraint of maximum length $h_{\max}$ of anomaly is equivalent to imposing a constraint on minimum intensity $\mu_{\min}$, as less intense anomalies are only detectable over longer timescales. This is linked to the sigma significance k and the background rate λ as follows:

$$\mu_{\min} \log(\mu_{\min}) - (\mu_{\min} - 1) = \frac{k^2}{2h_{\max}\lambda}.$$

For a statistical threshold of $k = 5$ (a "five-sigma event"), a background rate $\lambda \approx 28$ and $h_{\max} = 300$ (five minutes) this solves to give $\mu_{\min} \approx 1.055$, i.e. an anomaly of average magnitude less than 5.5% of the background level is statistically undetectable on this timescale. Anomalies that only occur over shorter timescales will have to be of greater magnitude in order to be detectable: for a rough estimate see Table 4.4.1. In particular, an anomaly present in just one time bin would have to more than equal (108.4%) the background radiation rate in order to be statistically detectable. By considering intervals rather than points, we are able to substantially improve on this power and detect the presence of less intense anomalies.

maximum time	$h_{\max}$	$\mu_{\min}$	relative magnitude	absolute magnitude
5 minutes	300	1.055	5.5%	1.54 counts/sec
1 minute	60	1.124	12.4%	3.48 counts/sec
10 seconds	10	1.313	31.3%	8.77 counts/sec
1 second	1	2.084	108.4%	30.36 counts/sec

Table 4.4.1: How large an anomaly needs to be, as a relative proportion of the background signal and as an absolute size assuming $\lambda = 28$, to be detected over different timescales at a 5-sigma threshold.

In order to search for anomalies with a length exactly $h_{\max}$, we could use an iterated form of the Page-CUSUM statistic (Page 1955; Lucas 1985) for Poisson data $S_T(\mu_{\min})$, defined as follows:

$$S_0(\mu_{\min}) = 0, \quad S_T(\mu_{\min}) = [S_{T-1}(\mu_{\min}) + x_T \log(\mu_{\min}) - \lambda(\mu_{\min} - 1)]^+$$

Here, the notation $[]^+$ is used to denote the maximum of the term in brackets and zero. The last time $\tau \leq T$ that $S_{\tau-1}(\mu_{\min})$ was zero is the estimated start point for any anomaly. When $S_T(\mu_{\min})$ resets to zero, this indicates that it is more likely that no anomaly is present than an anomaly of intensity $\mu_{\min}$ is. Under a null hypothesis of no anomaly present, this should occur frequently, moreso the larger $\mu_{\min}$ is. Values of $S_T(\mu_{\min}) \geq k^2/2$ indicate a sigma significance k over the interval $[\tau, T]$.

All anomalies that would be picked up by running a window of size $h_{\max}$ over the data will be picked up by using $S_T(\mu_{\min})$. One advantage of using $S_T(\mu_{\min})$ over the use

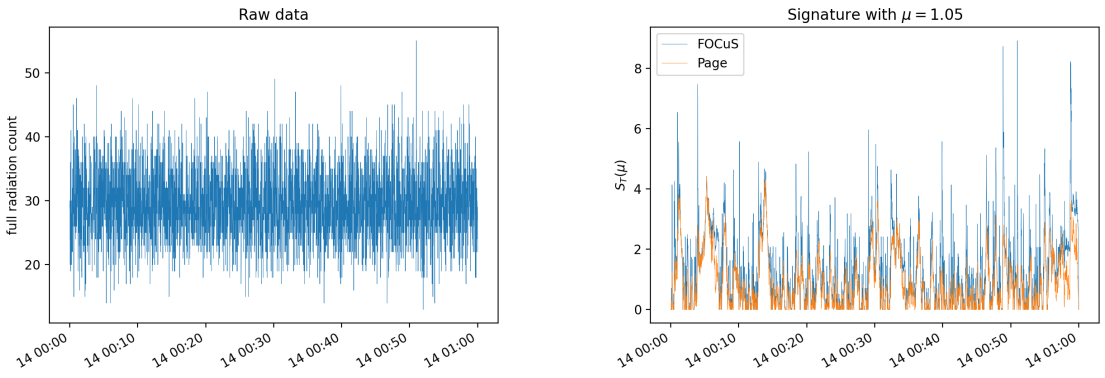


Figure 4.4.1: The first hour of data on the 14th August 2018, as raw data and as siginifance trace.

of a window of size $h_{\max}$ is that the method is one-scan: it does not need the separate storage of points inside the window that is required to remove the $T - h_{\max}$th point at time T. This makes it fast. One disadvantage of this algorithm is that it is not well-targeted for the detection of anomalies of $\mu > \mu_{\min}$, i.e. shorter, more intense anomalies.

The FOCuS method calculates $\max_\mu S_T(\mu) : \mu \geq \mu_{\min}$ in a similar timescale and is therefore able to efficiently detect anomalies of all sizes $h \leq h_{\max}$.

For both Page's method and FOCuS, anomalous *intervals* $[\tau, T]$ in the raw signal correspond to anomalous *points* in the significance trace $S_T(\mu)$. This means that it's a lot easier to identify interesting intervals in the significance trace than in the raw signal. For an example see Figure 4.4.1.

Because FOCuS can be thought of as an expanded form of Page's method, a similar set of intuitions apply, for example:

1. For a given $\mu_{\min}$, the FOCuS trace for that signal must be at least the Page trace for that signal.

2. The FOCuS trace tends to be fuzzier at low significance levels as it captures the normal fluctuations associated with individual time bins $h = 1$ and smaller intervals $h << h_{\max}$.

3. Reading from the signature traces, the estimated start point for an anomaly giving a FOCuS trace at a high significance is approximately where the FOCuS trace started climbing out of this fuzzy state. (The actual start point is available within the algorithm).

4.5 Dealing with background fluctuations

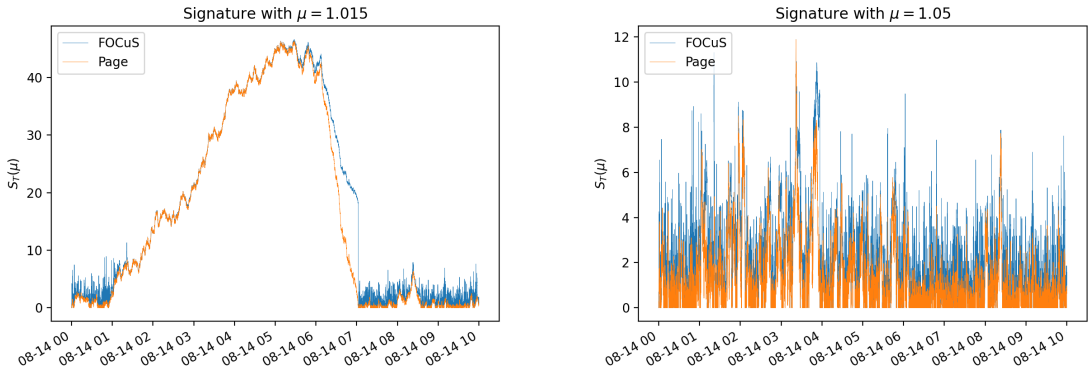


Figure 4.5.2: Comparing two different values of $\mu_{\min}$ for their ability to filter out small, long background fluctuations.

The first six hours of data of 14th August, 2018 contain a slight deviation above baseline background rate, but not enough to be considered anomalous. This is probably caused by a larger level of radon release from rock in rainy weather conditions. This demonstrates the necessity of choosing an appropriate $\mu_{\min} > 1$. Too small, and the background fluctuation is captured in the signature trace, as in Figure 4.5.2. Here, a choice of $\mu_{\min} = 1.015$ means that the slight deviation above baseline is recorded as a six hour long anomaly.

Even with $\mu_{\min} = 1.05$ we can see that the signature we receive differs from what we would expect from a simulation of independent Poisson random variables (shown in the right of Figure 4.5.3). There are periods of up to half an hour where there is a slight deviation above baseline. However because it no longer builds over long stretches of time, the statistical significance of this is not strong enough to trouble us.

It should be noted that an alternate way to handle this problem would be assuming

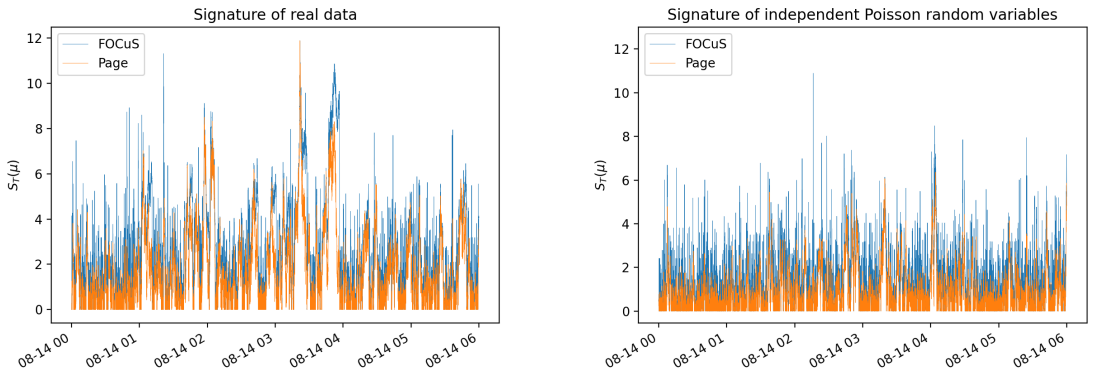


Figure 4.5.3: The difference between the data containing a small upwards fluctuation and independent Poisson random variables.

a nonconstant λ and performing a rolling estimate of the background rate λ_t, using e.g. a sliding window or exponential smoothing. This may be needed if we were interested in detecting longer anomalies and distinguishing them from background. However bias in the background rate estimate can be difficult to remove and in either case a choice of $\mu_{\min} > 1$ will help to mitigate this bias.

4.6 Resetting after large anomalies

From 11:10 to 11:15 on 14th August 2018, there is a large radiation anomaly clearly visible in the raw signal, as shown in Figure 4.6.4.

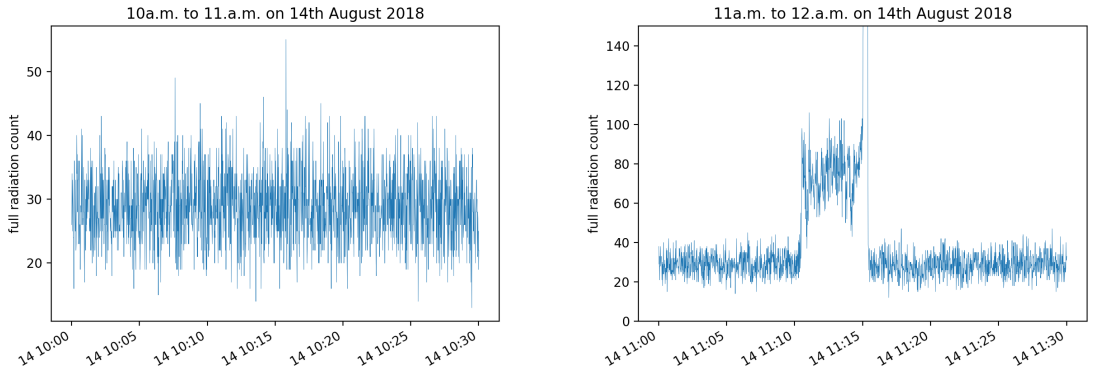


Figure 4.6.4: Two half-hours of data from 14th August 2018.

Large anomalies can leave traces in our significance trace long after they have ended. See Figure 4.6.5 for the effect of this large anomaly on our significance traces. Page's method tends to drop linearly and FOCuS quadratically (from a higher starting point), but both leave the same length of tail, which is in this case approximately five hours.

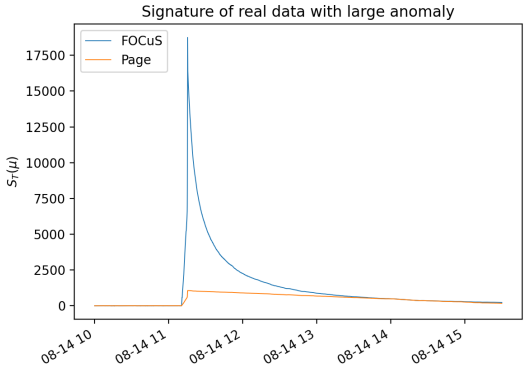


Figure 4.6.5: The signature of the large anomaly shown in Figure 4.6.4.

In order to get around the tail trailing behaviour, we institute a parameter h_{clear} that will remove any start points further in the past than h_{clear} if in the interval $[T-h_{\text{clear}}, T]$ contains no possible start points for an anomaly of size at least $\mu_{\min}$. This says that whatever anomaly is present is deemed to have ended, and should no longer be recorded in the current signal.

There is no positive evidence for an anomaly of intensity $\mu_{\min}$ beginning anywhere in the interval $[T - h_{\text{clear}}, T]$ and ending at T precisely when

$$S_T(\mu_{\min}) = \min_{t \in [T-h_{\text{clear}}, T]} S_t(\mu_{\min})$$

This can be calculated quickly using the ascending minima algorithm (Harter 2009) with computational cost not dependent on h_{clear}.

Because we know that positive evidence for an anomaly of size $\mu > \mu_{\min}$ on an interval necessitates positive evidence for an anomaly of size $\mu = \mu_{\min}$ on that interval, we can calculate Page's statistic and use it to reset FOCuS using the same condition. However, it can be more advantageous to reset FOCuS using the signal output from

FOCuS directly, that is if

$$\max_{\mu>\mu_{\min}} S_T(\mu) = \min_{t\in[T-h_{\text{clear}},T]} \max_{\mu>\mu_{\min}} S_t(\mu).$$

In most cases this will be very similar to resetting using Page's statistic. Where it differs is when passing over a large anomaly, the FOCuS statistic will drop more quickly and will continue to drop even if a small anomaly continues to be present. However, because the FOCuS algorithm would either way not store the start point corresponding to the small anomaly, resetting this way can be preferable.

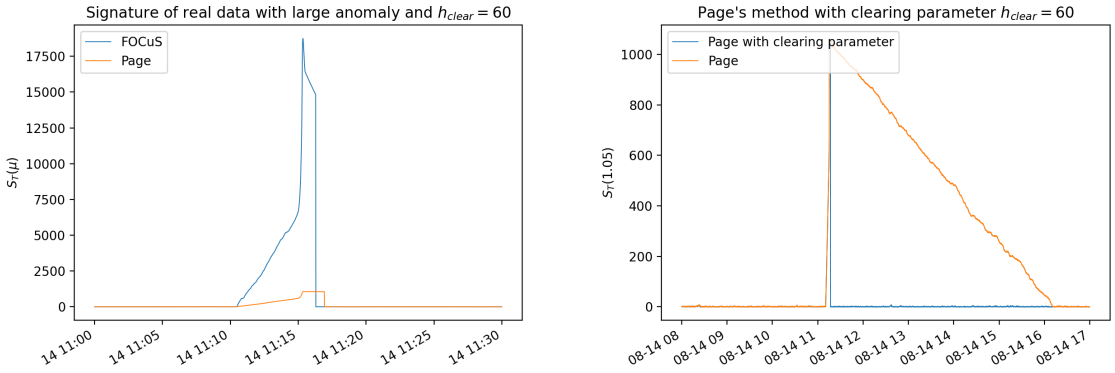


Figure 4.6.6: By using a clearing parameter h_{clear}, we can reset our method after large anomalies have ended.

Figure 4.6.6 shows the effect of this resetting strategy. Arbitrarily choosing $h_{\text{clear}} = 60$, we find that after passing over a large anomaly the algorithm resets within about a minute. Page takes a little longer than FOCuS as it takes a little longer to be sure of no evidence at all within the last minute, but even Page resets quickly compared to the five-hour tail of without a clearing parameter (see Figure 4.6.6).

4.7 Finding a threat

Figure 4.7.7 gives an artificially generated example of the kind of threat profile we would like our algorithm to be able to detect. A threat source approaches the detector, stops, and then leaves, over the total course of three minutes. With a radiation count mean of

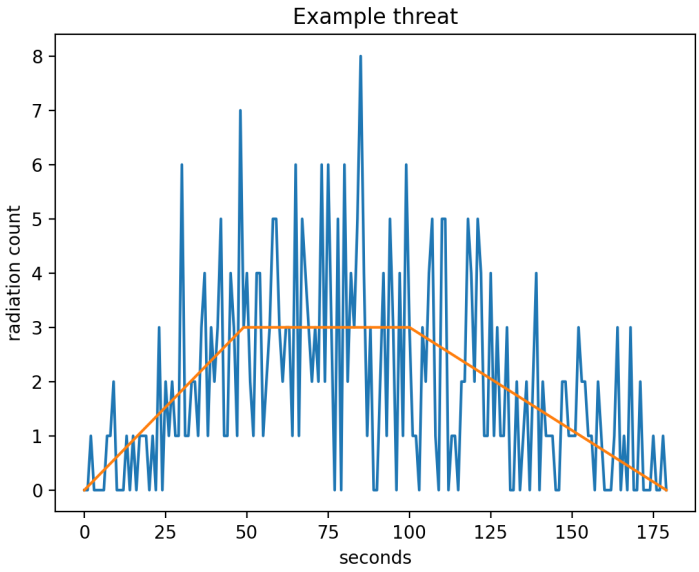


Figure 4.7.7: An example threat with intensity $\mu \approx 1.1$.

up to 3 counts/sec compared to a background count mean of 28 counts/sec, this means it is barely visible to the naked eye when added to the SIGMA data (see Figure 4.7.8). Here our threat is about 10% of the size of the background signal, giving $\mu \approx 1.1$.

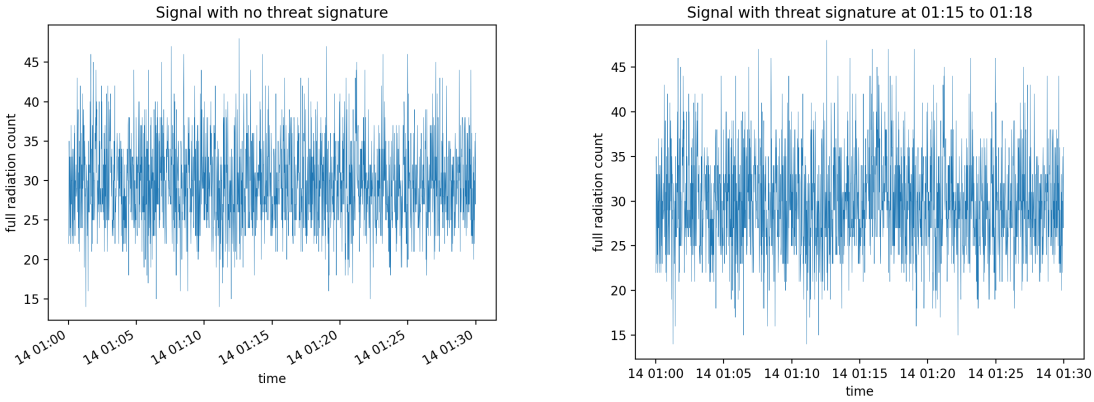


Figure 4.7.8: The example three minute threat incorporated into the SIGMA data at 01:15 to 01:18, barely visible to the naked eye.

The threat is detectable by FOCuS as shown in Figure 4.7.9 and is clearly visible in the significance trace. It crosses the 7-sigma significance line. The advantage over Page's method is also apparent: as FOCuS correctly targets the intensity of the anomaly it records a higher significance. The location of the anomalous interval can be easily

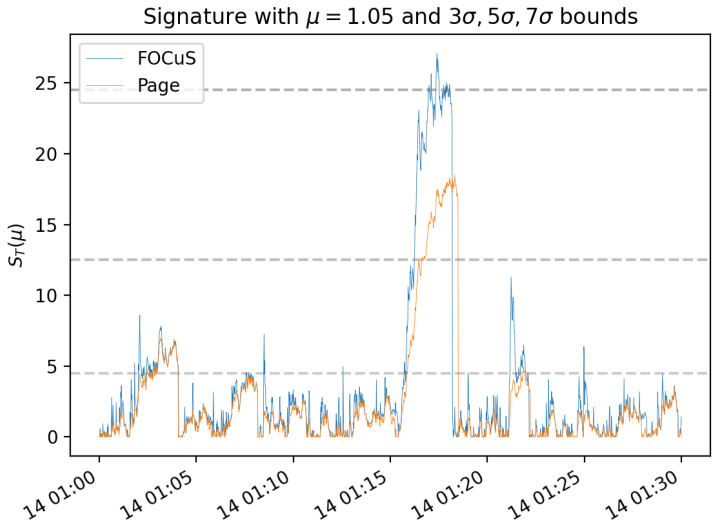


Figure 4.7.9: The signature trace of the threat with dashed lines showing 3σ, 5σ and 7σ significance levels.

read from the graph as from wherever FOCuS began to increase to where it attained its maximum value: here from 01:15 to 01:18.

4.8 Future work

Future work should incorporate the 4096 energy bands. Here, the average radiation count varies greatly by energy band. The raw energy band counts for one hour's worth of data on the 6th August 2018 are shown in Figure 4.8.10. The 4096 bands are represented individually on the left, and are grouped into logarithmic multiples of $2^{1/8}$ on the right to give a spectral graph that's easier to read. Most of the background radiation count occurs in energy band 100 to 1000.

The idea is that each of these energy bands may provide a different utility for identifying a threat, depending on the relative mix of background and anomalous radiation held by the band. The bands with the most utility for finding anomalies would have low background radiation, but a high amount of anomalous radiation if an anomaly was present. For an example of this, see Figure 4.8.11 of the log energy spectrum of a

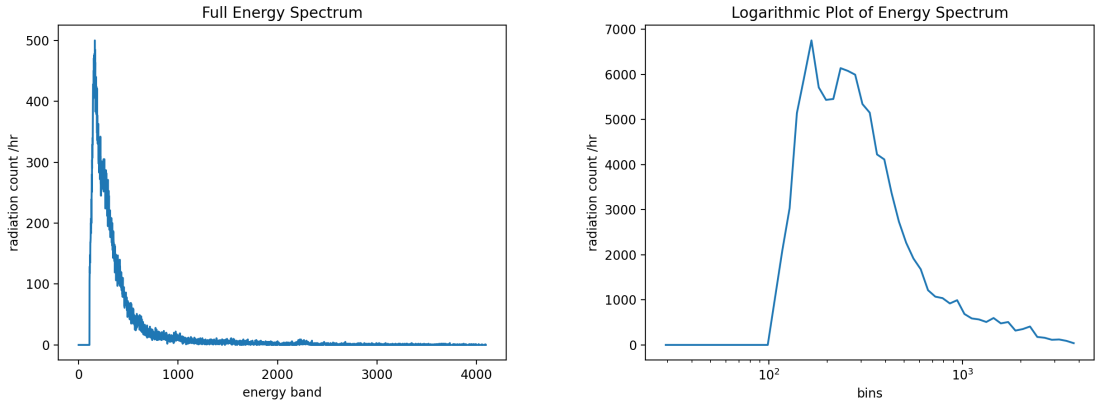


Figure 4.8.10: Energy band counts for one hour's worth of data

large anomaly on the 6th August 2018 plotted against nearby background traces. Note that although the anomaly is numerically greatest in the centre, it has a greater ratio of anomaly to background rate in the spike off to the right.

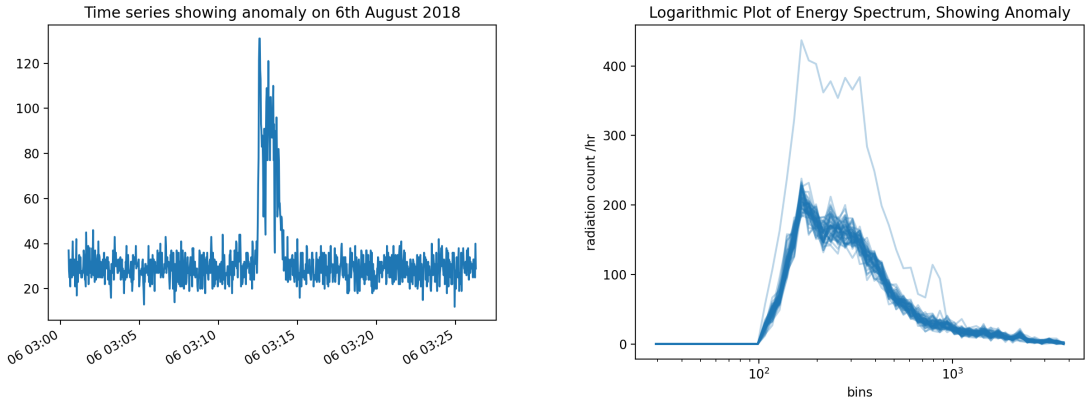


Figure 4.8.11: Graph showing the time and spectral structure of a large anomaly present in the SIGMA data on 6th August 2018.

Previously we have been using all the bands and combining them into one (by summing up counts) before calculating significance. Other options include:

1. Only use one subset of the bands with the most utility and combine before calculating significance.

2. Use multiple distinct subsets of bands, adding significances across bands.

3. Use multiple increasing subsets of bands that contain each other, taking the max-

imum significance across bands.

4.8.1 Utility criteria for subset selection

Suppose that we have a subset of the signal included in our anomaly detection algorithm and we are looking into whether it would make sense to expand it.

Already included elements have in total anomaly amount A and background amount B. We are looking to see if it increases overall significance to add an energy band with anomaly amount a and background amount b. Working on the assumption that a and b are small compared to A and B, it can be shown that this binary yes/no question is a function only of the ratio a/b. For example, consider the alternate question of whether to add a category with anomaly amount $2a$ and background amount $2b$. If we assume the amounts are sufficiently small to be able to discount second-order effects, then the answer to both questions should be the same.

A good ranking measure of utility of a spectral band should be the ratio between the (normalised) rate a of the threat profile in that band and the background rate b in that band. We want to include all and only spectral bands with utilities above a set threshold.

Exactly how this utility threshold should be chosen for different threat profiles is not clear and may vary according to our desired false positive rate, so we may wish to track multiple increasing subsets. However, it is possible to both reliably estimate the background profile of the signal at that time using previous signal points, and also the radiation signature of the threats we are hoping to find. This is because, for example, a sample of Uranium-235 will contain both the signature of U-235 and the signature of the decay products of U-235, where the rate of decays of each isotope present are constant in equilibrium (Bateman 1910). Therefore we cannot use just one isotope's decays when calculating a threat profile, but we can construct a threat profile specific to that isotope.

4.9 Summary

4.9.1 What we have done

We have constructed a fast algorithm to run on the SIGMA data and report possible anomalies for further consideration. To summarize, the detection procedure with specified parameters $\mu_{\min}$, h_{clear}, k is as follows:

1. Set a sensible $\mu_{\min}$ based on your upper time limit for an anomaly that removes long fluctuations from the data (we suggest $\mu_{\min} = 1.05$ here but higher values of $\mu_{\min}$ may be sensible if the data contains more fluctuations)

2. Run FOCuS with the clearing parameter h_{clear} in order to easily reset after passing over large anomalies. The algorithm is not particularly sensitive to exactly what h_{clear} is, but in this report we have arbitrarily chosen $h_{\text{clear}} = 60$ in order to reset the algorithm after a minute.

3. Alert all instances of intervals giving a significance trace greater than $k^2/2$ (which indicates a k-sigma significance level) for further checking as a possible threat.

4.9.2 What we could do next

In order to specifically look for a particular threat profile, we could compute what the threat profile should be based on an isotope mix, and then include only a subset of energy band categories based on the utility ratio, choosing a utility threshold as appropriate. This would likely improve the accuracy of detection of specific known threat profiles.

We could also pair FOCuS as a preliminary method with a more accurate but more computationally expensive method, as follows:

We deliberately choose a low sigma significance level k and therefore high false positive rate in order to ensure that FOCuS when run as preliminary accurately picks up all anomalies of interest. For example, if our overall desired false positive rate is one in eight hours requiring human input checking, we run FOCuS with a false positive

rate of one in every ten minutes (600 seconds) and only feed positives highlighted by FOCuS into our more computationally expensive algorithm. This cuts down the amount of computation needed for the more expensive algorithm by at least 600 times, if not more as FOCuS can accurately estimate the start point of anomalies so each positive only requires the checking of one interval. This approach is summarised in Figure 4.9.12.

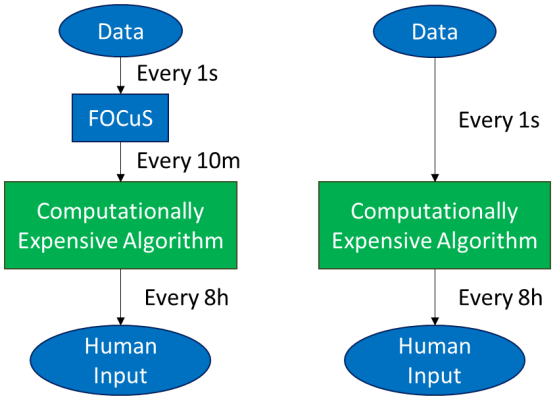


Figure 4.9.12: Flowcharts showing the computational processing comparison for with and without FOCuS as a preliminary method.

Data access statement

The SIGMA Data supporting this chapter are not open-access for security reasons. Please contact the Nuclear Security Science Network SIGMA Data Challenge team at `info@nusec.uk` if you want more information or to request access.

Chapter 5

Linear in time FOCuS for Exponential family models

5.1 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, Dilillo, 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, each involving different trade-offs between statistical efficiency and computational cost. 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, Tasoulis, and Adams 2011; Ross and Adams 2012; Chen and Tian 2010), or specify the distribution of the data after a change (e.g. Page 1955; 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, Eckley, Fearnhead, and Rigaill (2023) proposed a new algorithm called Functional Online Cumulative Sum (FOCuS). This algorithm is able to perform the 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, Dilillo, et al. 2023) and Binomial (Romano, Eckley, and Fearnhead 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.

5.2 Background

5.2.1 Problem statement

Assume we observe a univariate time series signal $x_1, x_2, ...$, 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 that data as being independent realisations from some parametric family with density $f(x \mid \theta)$. 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 base testing for a change using the likelihood-ratio test statistic.

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 Section 5.4.

The log-likelihood for the data $x_{1:T} = (x_1, \ldots, x_T)$, 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

$$LR_T := 2\left\{\max_{\theta_1,\tau} \ell(x_1{:}x_T|\theta_0, \theta_1, \tau) - \ell(x_1{:}x_T|\theta_0, \cdot, T)\right\}.$$

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 Chu, Hornik, and Kaun (e.g. 1995) and Eichinger and Kirch (2018) fix the number K of changepoint times tested, with these being of the form $\tau = T - h_i$ for a suitable choice of $h_1, \ldots, h_K$. Alternatively one can use Page's recursion (Page 1954; Page 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.

5.2.2 FOCuS for Gaussian data

As an alternative to the MOSUM or Page's recursion , Romano, Eckley, Fearnhead, and Rigaill (2023) 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

$$\ell(x_1{:}x_T|\theta_0, \theta_1, \tau) - \ell(x_1{:}x_T|\theta_0, \cdot, T) = \sum_{t=\tau+1}^{T} \left\{\log f(x_t|\theta_1) - \log f(x_t|\theta_0)\right\}.$$

We can then introduce a function

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

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 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

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

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, Eckley, Fearnhead, and Rigaill 2023). 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, Eckley, Fearnhead, and Rigaill (2023) 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 (1955),

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

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

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

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

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

Substituting into Page's recursion we obtain

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

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 Section 5.3 we will present the FOCuS algorithm in a slightly different way to Romano, Eckley, Fearnhead, and Rigaill (2023). Assume that $\mathcal{I}_{T-1} = \{\tau_1, \ldots, \tau_n\}$, with the candidate locations ordered so that $\tau_1 < \tau_2 < \ldots < \tau_n$. We can now define the difference between successive quadratics as

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

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 $\mathcal{C}_{\tau_i}^{(\tau_{i+1})}(\theta_1) \geq 0$. By definition $\mathcal{C}_{\tau_i}^{(\tau_{i+1})}(\theta_0) = 0$. Hence it is readily shown that

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

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

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

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

This suggests Algorithm 2. Note that this algorithm is presented differently from that in Romano, Eckley, Fearnhead, and Rigaill (2023), 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 $\mathcal{C}_{\tau_j}^{(T-1)}$ is smaller than that of $\mathcal{C}_{\tau_{j-1}}^{(\tau_j)}$, 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 $\mathcal{C}_{T-1}^{(T)}$. 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).

Algorithm 2: 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, $\mathcal{C}_{\tau_i}^{(\tau_{i+1})}(\theta_1)$, for $i = 1, \ldots, n$, with $\tau_i < \tau_{i+1}$ and $\tau_{n+1} = T - 1$ such that

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

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

Data: x_T

Set $j = n$;
Set $l_0 = \theta_0$;
while $j > 0$ **do**
  if $l_j \leq l_{j-1}$ **then**
    Update $C_{\tau_{j-1}}^{(T-1)}(\theta_1) = C_{\tau_{j-1}}^{\tau_j}(\theta_1) + C_{\tau_j}^{(T-1)}(\theta_1)$;
    Recalculate l_{j-1}, largest root of $C_{\tau_{j-1}}^{(T-1)}(\theta_1) = 0$;
    Update $\tau_j = T - 1$;
    Update $j = j - 1$;
  end
  Break;
end
Set $\mathcal{C}_{T-1}^{(T)}(\theta_1) = \theta_1(x_T - \theta_1/2)$;
Set $\tau_{j+1} = T - 1$ and $\tau_{j+2} = T$;
Set $l_{j+1} = 2x_T$;
Set $n = j + 1$;
return *The set of n quadratic differences, $\mathcal{C}_{\tau_i}^{(\tau_i+1)}(\theta_1)$ and roots l_i for $i = 1, \ldots, n$.*

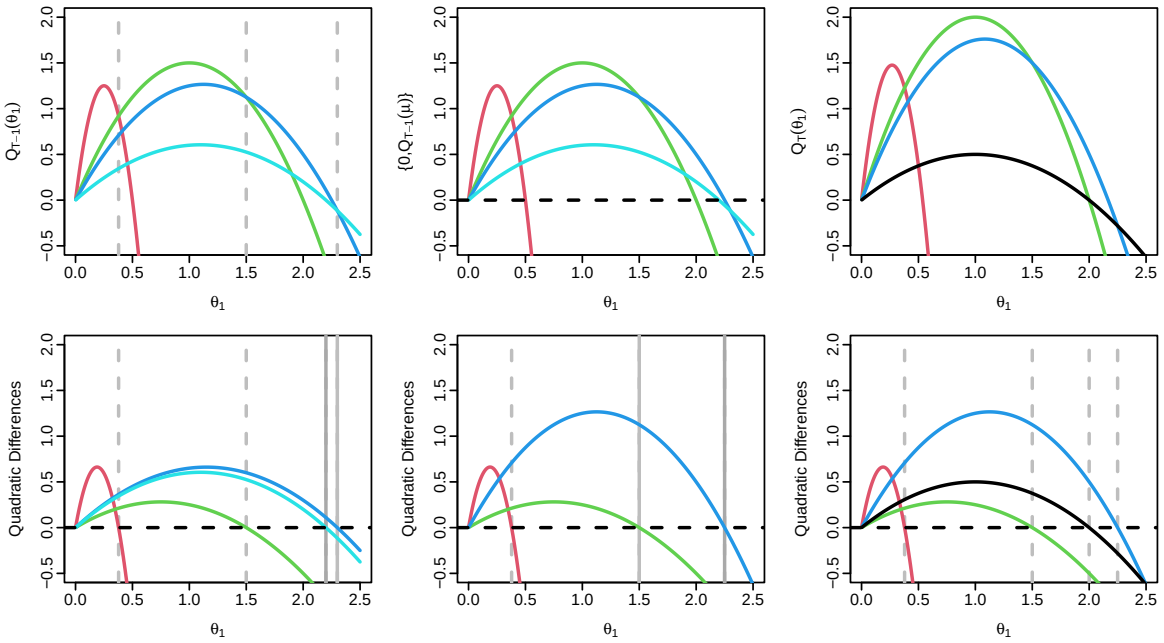


Figure 5.2.1: Example of one iteration of FOCuS. The top row shows the action of FOCuS on the quadratics themselves, whereas the bottom row operates on the corresponding quadratic differences. This form of the algorithm adds the zero line and prunes, before adding the quadratic representing the current point to all quadratics.

A pictorial description of the algorithm is shown in Figure 5.2.1. 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 $2T$.

5.3 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 Equation 5.2.2) 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 2, Ward, Dilillo, et al. (2023) and

Romano, Eckley, and Fearnhead (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

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

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 given in Table 5.3.1 include Gaussian change in mean (with known variance), Gaussian change in variance (with known mean), 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, 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 Binomal(n, θ) data as Poisson$(n\theta)$ for large n and small θ, or utilising the fact that $x \sim N(0, 1)$ 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 Section 5.6 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 Section 5.2.2 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 2 we need to store the differences $C_{\tau_i}^{(\tau_j)}(\theta_1)$ in the log-

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

Table 5.3.1: Examples of one-parameter exponential families and the corresponding forms of $\alpha(\theta)$, $\beta(\theta)$ and $\gamma(x)$. 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 models assumes the number of trials is n; and the Gamma model is for a change in scale parameter with shape parameter k.

likelihood for different choices of the changepoint location. This becomes

$$\ell(x_1{:}x_T|\theta_0,\theta_1,\tau_i) - \ell(x_1{:}x_T|\theta_0,\theta_1,\tau_j) = \\ [\alpha(\theta_1) - \alpha(\theta_0)] \sum_{t=\tau_i+1}^{\tau_j} \gamma(x_t) - [\beta(\theta_1) - \beta(\theta_0)](\tau_j - \tau_i).$$

These curves can 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 2 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 $t = \tau_i + 1, \ldots, \tau_j$, and define $\theta_1^\tau (\neq \theta_0)$ to be the root of

$$\ell(x_1{:}x_T|\theta_0,\theta_1^\tau,\tau) - \ell(x_1{:}x_T|\theta_0,\cdot,T) = 0.$$

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

Proposition 5.3.1. *Suppose that for our choice of θ_0 the function*

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

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 Appendix.

The assumption captures the idea that for the one-parameter exponential family model, higher values of $\gamma(x_t)$ correspond to evidence for an up change $\theta_1 > \theta_0$ whereas lower values of $\gamma(x_t)$ correspond to evidence for a down-change $\theta_1 < \theta_0$. Exponential family models with a defined direction of change can relabel their θ parameter so that this direction applies (for example, Gamma change in rate where a higher mean implies a lower rate can be relabelled as Gamma change in scale). All the exponential family models in this paper have been stated in such a way as for this assumption to hold.

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 2 that compares the roots of two curves with a condition that compares their $\bar{\gamma}$ values. Or equivalently we can implement Algorithm 2 but with $l_i = \bar{\gamma}_{\tau_i:\tau_{i+1}}$ rather than the root of $\mathcal{C}_{\tau_i}^{\tau_{i+1}} = 0$.

An immediate consequence of this result is that one-parameter exponential family models that satisfy the condition of Proposition 5.3.1 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 5.3.2. *The Gaussian (change in mean), Poisson, Binomial, and Gamma variations of the FOCuS algorithm have the same pruning.*

An example of this corollary is shown in Figure 5.3.2

More generally we have the following.

Corollary 5.3.3. *If an exponential family model satisfies the condition of Proposition 5.3.1, then the pruning under this model will be identical to the pruning of FOCuS for the Gaussian change in mean model analysing data $\gamma(x_t)$.*

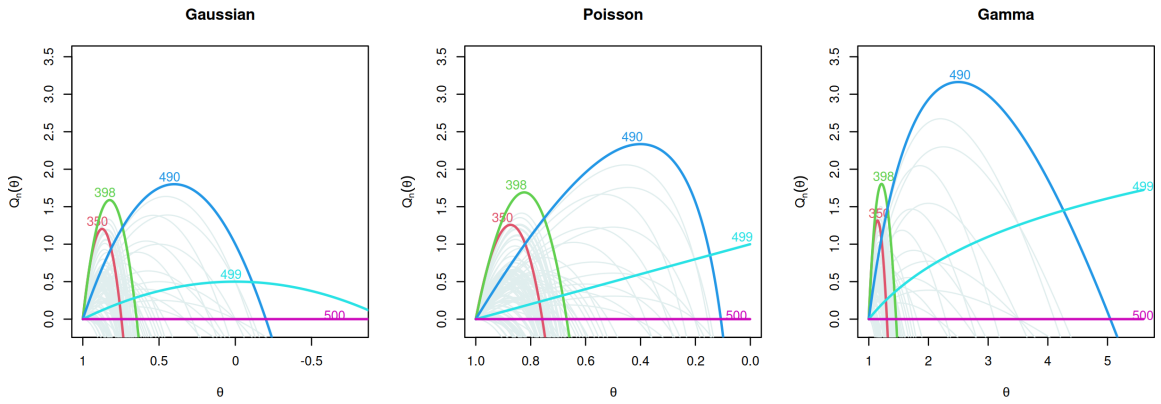


Figure 5.3.2: Comparison of three different cost functions computed from the same realizations $y_1, \ldots, y_{500} \sim \text{Poi}(1)$. The leftmost, center, and rightmost figures show the cost function $Q_n(\theta)$ should we assume respectively a Gaussian, Poisson, or Gamma loss. The floating number refers to the timestep at which each curve was introduced. In gray, the curves that are no longer optimal and hence were pruned.

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 $x_1^2, x_2^2, \ldots$.

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, Eckley, Fearnhead, and Rigaill 2023) applies to these equivalent exponential family models. The results on the expected number of curves kept by FOCuS makes minimal assumptions for the data, namely that the observations are exchangeable. These results imply that on average the number of curves kept at iteration T is $O(\log T)$.

5.4 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

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

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 2 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 2. If we now change θ_0, the roots of the curves will change, but by Proposition 5.3.1 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 2 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 2 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 2 across different exponential family models, that we demonstrated with Corollary 5.3.3, also immediately follows.

5.5 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 5.5.1. *For any $\tau_1 < \tau_2 < ... < \tau_n < T$, we have*

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

Proof: See Appendix. A pictorial explanation of the result is also shown in Figure 5.5.3

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

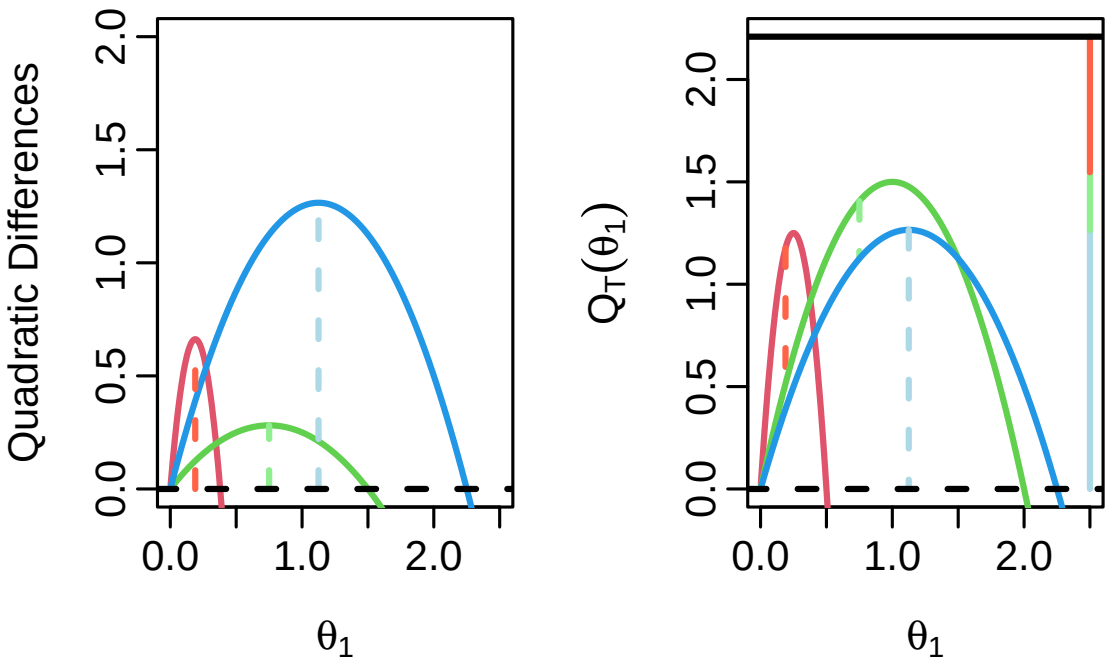


Figure 5.5.3: Example of the bound of Proposition 5.3.1 for the pre-change mean known case. Left-hand plot shows the differences between the three curves that contribute to $Q_T(\theta_1)$. The $m_{\tau_i:\tau_j}$ values correspond to the maximum of these curves (vertical lines). Right-hand plot shows $Q_T(\theta)$, 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 3. 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 curve with the next largest τ_k value and repeat.

Empirical results suggest that for $\tau_1 ... \tau_n \in \mathcal{I}_T$ when searching only for an up-change (or analogously only for a down-change), the upper bound in Proposition 5.5.1 is quite 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 Section 5.6, at the majority of time-steps only one curve needs to be checked before we know that Q_T is less than our threshold.

Algorithm 3: 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.

```
Set k = n;
while k > 0 do
    Calculate m_{τ_k,T};
    if m_{τ_k,T} + M_{τ_k} < Threshold then
        return no change
    else if m_{τ_k,T} ≥ Threshold then
        return change on [τ_k, T]
    end
    k −= 1;
end
return no change
```

5.6 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 (Romano, Eckley, Fearnhead, and Rigaill 2023).

In Figure 5.6.4 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 maxima. We compare three different FOCuS implementations: (i) FOCuS with pruning based on the ordered roots $l_1, \ldots, l_n$, where such roots are found numerically through the Newton-Raphson procedure, (ii) FOCuS with the average value pruning of Section 5.3 and lastly (iii) FOCuS with the average value pruning and the adaptive maxima checking of Section 5.5.

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.

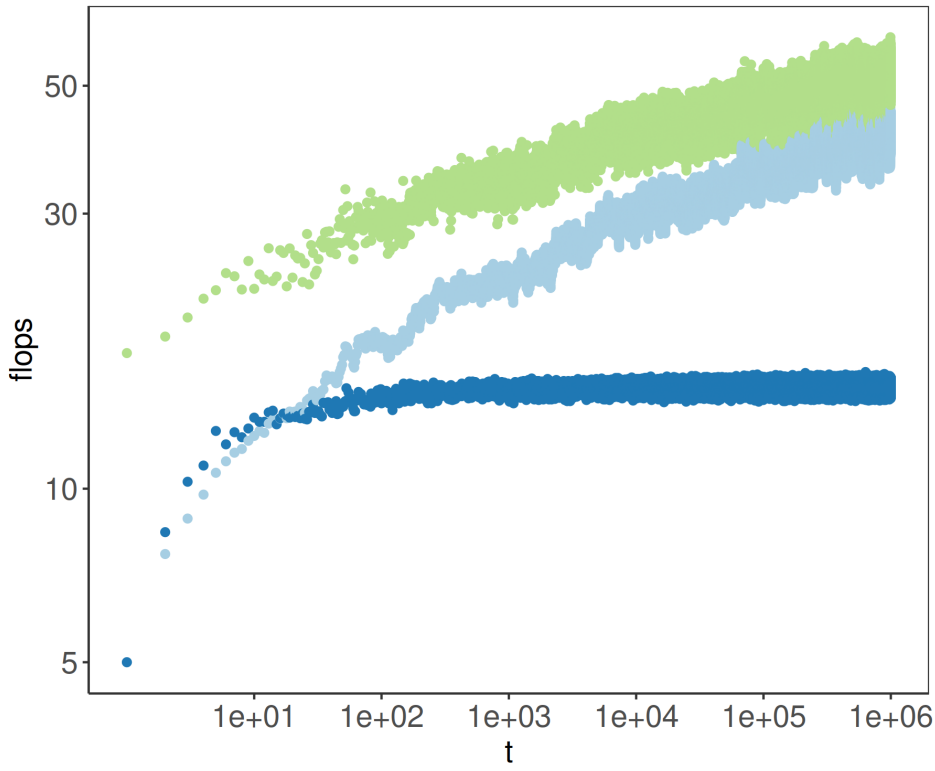


Figure 5.6.4: Flops per iteration in function of time for three FOCuS implementations. In green, the flops for FOCuS with pruning based on calculating the roots $l_1, \ldots, l_n$ numerically. In light blue, FOCuS with the average value pruning. In blue, finally, FOCuS with the average value pruning and the adaptive maxima checking. Log-scale on both axes.

In Figure 5.6.5 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 5.5.1. As it can be seen from the short spike after

the change, this is 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.

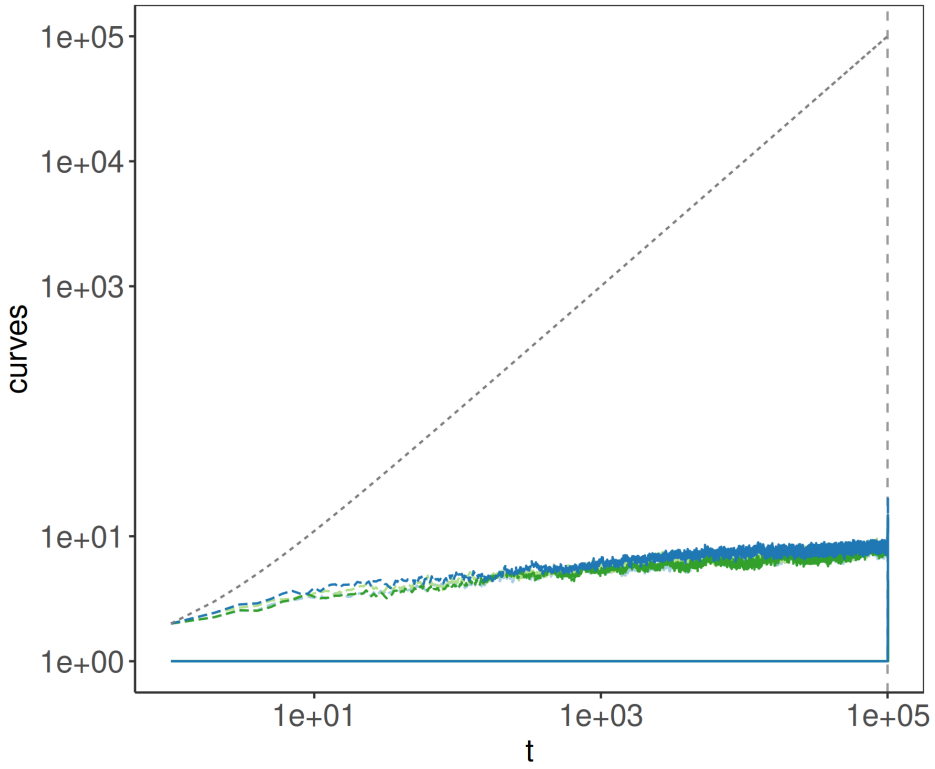


Figure 5.6.5: Number of curves to store and evaluations per iteration as a function of time for FOCuS running over a long non-anomalous signal followed by a short anomaly. The grey dotted line is the naive cost of computing the CUSUM likelihood ratio test. The dashed lines are the number of curves stored by FOCuS over some Gaussian (light-green), Poisson (dark-green), Bernoulli (light-blue) and Gamma (dark-blue) realizations. The solid lines at the bottom are the number of curves that need to be evaluated at each iteration with the adaptive maxima checking: in all three cases always 1 while the signal is non-anomalous. Log-scale on both axes.

To illustrate the advantages of shifting data points from one assumed exponential family to another, we evaluate the performances of FOCuS for a Gaussian change-in-variance by casting the problem into a Gamma change-in-scale (as mentioned in Section 5.3). Let $x_t \sim N(0, \theta_0)$, then, by the simple transformation of the data $y_t = x_t^2$, we

notice that:

$$\log f(x_t, \theta_0) = \log g(y_t, \theta_0) = \frac{1}{2} \log\left(\frac{1}{\theta_0}\right) - \frac{y}{2\theta_0}.$$

We can see that y_t is Gamma distributed with scale parameter $\theta_0/2$ and shape parameter $k = 1/2$. For comparison, we add to the evaluation the naive solution of testing for a Gaussian change-in-mean to the square data (equivalent to the simple CUSUM test). In addition, for both costs we compare the cases for pre-change parameter known and unknown.

For a process distributed under the null as a normal centred 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, Wang, and Samworth (2022). We then introduce a a change at time 1000 and measure performances in terms of detection delay (the difference between the detection time and the real change).

In Figure 5.6.6 we illustrate the scenarios and present results in terms of the proportion of detections within t observations following the change. We note how for a positive change large enough, e.g. for $\theta_1 = 2$, there is no evident advantage in employing the Gamma cost over the Gaussian cost for detecting a change in variance. Arguably, a simple thresholding procedure could perform as well as our statistics in such scenarios. As we however lower the signal-to-noise ratio and shift towards more subtle changes, we can see how the Gamma cost improves significantly on the detection delay. In case of $\theta_1 = 1.25$, we can clearly see that the best performances are achieved by the gamma cost in case of pre-change known and unknown (with little difference between the two). Similarly, for a reduction in variance, we see the Gaussian cost showing significantly slower delays.

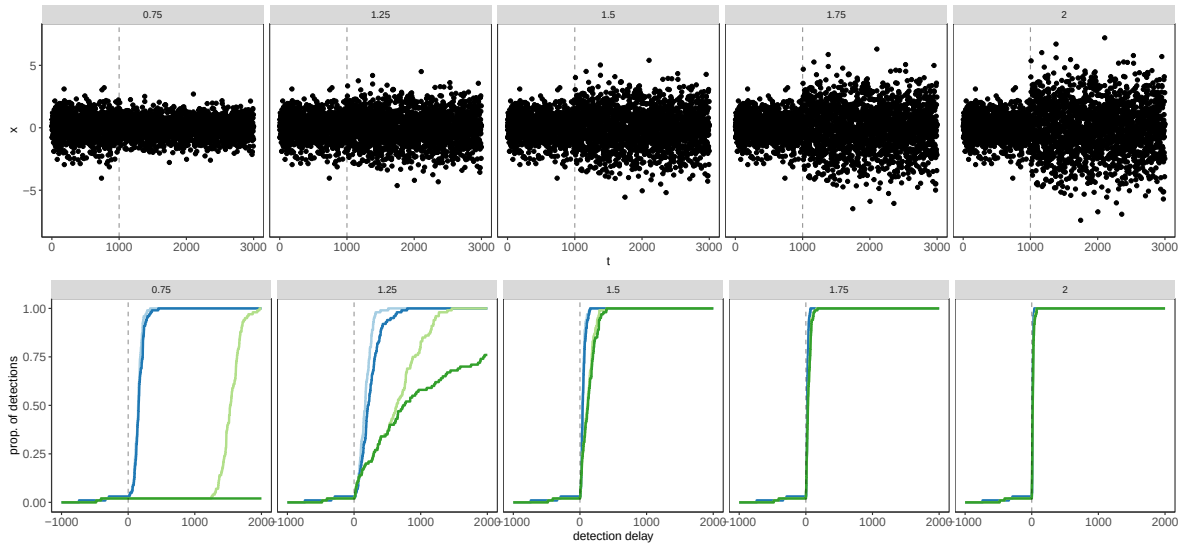


Figure 5.6.6: Empirical evaluation of FOCuS for Gaussian change-in-variance. Top row: example sequences for our simulation scenarios, with labels indicating the post-change parameter θ_1, whilst vertical dotted line refers to the changepoint location τ. Bottom row: proportion of detections in the function of the detection delay for Gamma change-in-scale with pre-change parameter known (light blue), unknown (dark blue), and Gaussian change-in-mean for 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.

5.7 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, Eckley, and Fearnhead 2021).

We do not cover the case where the distribution contains two or more unknown

parameters which may both change, such as Gaussian change in mean and variance simultaneously. We note that testing for two parameters in one data stream can be thought of as a multivariate problem running on multiple copies of the same data stream, as is done in Pishchagina, Romano, Fearnhead, et al. (2023). Similarly, we do not cover the problem of testing only changes in one parameter while another parameter is unchanging but unknown. In practice, if a parameter is assumed not to change, then it is possible to estimate it well enough using previous data to be able to treat it as known, while keeping in mind the robustness of the algorithm to small errors in parameter estimation.

The method we present is robust to these kinds of parameter estimation errors. For example, a test for Gaussian change in mean assuming variance 1 which has been slightly misestimated will slightly raise (if under-estimated) or lower (if over-estimated) the threshold required to detect anomalies at a given significance level, but will not change the most promising intervals found by the algorithm. Because thresholds are usually set in practice by considering factors such as desired false positive rate rather than using a defined significance level, this misestimation turns out to have little practical effect.

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 autocorrelation 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 Romano, Rigaill, et al. (building on ideas in 2022), Cho and Fryzlewicz (2020), and Hallgren, Heard, and Adams (2021) may be better.

Chapter 6

Scaling and multivariate FOCuS

6.1 Introduction

In this chapter, we discuss ways to apply the FOCuS algorithm in a general multivariate setting, with high numbers of coordinates. Specifically, we look at how the framework provided by FOCuS can help us deal with computational complexity issues that arise as the number of data coordinates expands, and the tradeoff of loss of statistical power that occurs when operating only on the output of FOCuS versus the entire data stream. We look at ways of increasing the practicalities of FOCuS in this setting, including:

1. Deriving a tight theoretical bound on the memory cost of FOCuS running on a data stream with a minimum anomaly intensity $\mu_{\min}$. This is particularly useful for multivariate algorithm development as it directly informs required memory allocation to each variate in order to handle signals of any length.

2. Dealing with the additional noise present in a multivariate dataset by local thresholds and sparsity constraints on which data streams can contain anomalies.

3. Combining information from different data streams about where a multivariate anomaly starts.

Our main comparator is the Online Changepoint Detection (OCD) algorithm (Chen, Wang, and Samworth 2022), and to this end our methods are presented mostly in the

Gaussian change in mean setting.

6.2 Proving a good constant bound for FOCuS with a minimum parameter value

Previous work has proved that the number of curves stored by FOCuS without a minimum parameter $\mu_{\min}$ is on the order of $\log(T)/2$ as $T \to \infty$, and that with a minimum parameter the number of curves stored is bounded (see Chapter 3). However, the proof available did not provide a good bound, only proving finiteness, when empirically there seems to be a bound (dependent on the size of $\mu_{\min}$) that is quite small, for example less than ten for any reasonable value of $\mu_{\min}$. We provide such a computable bound in the Gaussian and Poisson cases using recent results about random walk behaviour, and show it matches well with observed data. This has applications in knowing how much memory storage to allocate for practical use of FOCuS methods.

Definition 6.2.1 (Convex minorant)**.** *Given a time series S_t, the convex minorant is the greatest convex time series $\leq S_t$ for all t. Its segments are the intervals between the time points t where S_t is equal to its convex minorant.*

Letting $S_t := \sum_{s=1}^{t} X_s, 1 \leq t \leq T$, we have that for any two points $(t_1, S_{t_1}), (t_2, S_{t_2})$, the gradient of the line connecting those points is $\bar{X}_{t_1:t_2}$ the mean of X_t over the interval $(t_1, t_2]$. Because the startpoints in FOCuS maximally divide the signal into intervals of increasing mean (see for example Chapter 5 Proposition 5.3.1), they maximally divide the random walk sum S_t into intervals of increasing gradient. This means these intervals form exactly the segments of the convex minorant of the random walk S_t, as a function with increasing gradient is a convex function. This fact was utilised in Chapter 3 Proposition 3.4.1 to relate the expected number of startpoints kept by FOCuS to the expected number of segments on the convex minorant of a random walk.

We further expand on this by considering the number of segments on a convex minorant of a random walk with gradient above a defined nonzero threshold value, $\mu_{\min}$. We use the following theoretical result:

Theorem 6.2.2 (Convex minorant construction)**.** *Let $X_t, 1 \leq t \leq T$ be independent identically distributed continuous random variables, and let $S_t := \sum_{s=1}^{t} X_s, 1 \leq t \leq T$ be the corresponding random walk. Let*

$$k, [(L_1, U_1), (L_2, U_2), ...(L_k, U_k)]$$

be the number of segments k on the convex minorant, the L_i the (horizontal) length of each segment starting from the left, and U_i the vertical fall or rise of each segment. It is possible to distributionally construct $(k, (L_1, ..., L_k), (U_1, ..., U_k))$ using the following process:

1. *Generate a random permutation σ of the numbers $1, ..., t$, i.e. a random member of the Symmetric group Sym(t).*
2. *Let k be the number of cycle lengths in σ, and $l_1, ..., l_k$ the corresponding cycle lengths when written in canonical cycle notation.*
3. *Construct $u_1, ..., u_k$ by letting each $u_i \sim \sum_{j=1}^{l_i} X_j$ independently of each other, i.e. progress the random walk for l_i time.*
4. *Sort the pairs (l_i, u_i) by gradient u_i/l_i in ascending order to give the sorted list (L_i, U_i).*

Proof. See Abramson et al. (2011) □

Theorem 6.2.3 (Standard result from abstract algebra)**.** *In a randomly chosen permutation σ of l or more elements, the expected number of cycles of length l in σ is $1/l$.*

6.2.1 Bounds for Gaussian-FOCuS

In order for a curve to contribute to FOCuS, it must correspond to an edge on the convex minorant with gradient $U_i/L_i > \mu_{\min}$. This is only those edges on the right hand side of the convex minorant. However, we don't require the sorting step in order to count

curves, and can work with the l_i and u_i from the cycle random walk construction of Theorem 6.2.2 instead.

Lemma 6.2.4 (Normal rescaling)**.** *Let X_i be standard Normally distributed independent random variables and $l \in \mathbb{N}$, $\mu \geq 0$. Define $u := \sum_{i=1}^{l} X_i$.*

$$\mathbb{P}(u/l > \mu) = \mathbb{P}(Z > \sqrt{l}\mu),$$

where Z is a standard Normal random variable.

Proof is by sums and rescaling of Normals.

Given that we only want to count the i where $u_i/l_i > \mu$, due to independence we can simply add up the expected number of faces of each length multiplied by the probability that a face of that length has our desired gradient.

Lemma 6.2.5. *Let $X_t, 1 \leq t \leq T$ be standard Normally distributed, and let $S_t := \sum_{s=1}^{t} X_s, 1 \leq t \leq T$ be the corresponding random walk. The expected number of faces on the random walk with gradient at least μ is:*

$$\sum_{l=1}^{T} \frac{1}{l} \cdot \mathbb{P}(Z > \sqrt{l}\mu),$$

where Z is a standard Normal random variable.

Proof. We use the construction of Theorem 6.2.2. By Theorem 6.2.3 the expected number of faces on the convex minorant of a random walk of size $T \geq l$ that have length l is $1/l$, and by Lemma 6.2.4 the probability of a face of length l having the required gradient is $\mathbb{P}(Z > \sqrt{l}\mu)$. □

This gives us the expected values in Figure 6.2.1. We can use these to construct a bound on the number of quadratics as $T \to \infty$ and therefore on the required memory allocation to univariate FOCuS, or to each variate in a multivariate FOCuS, that will be appropriate for signals of any size while still being small enough to not crowd our algorithm's workspace.

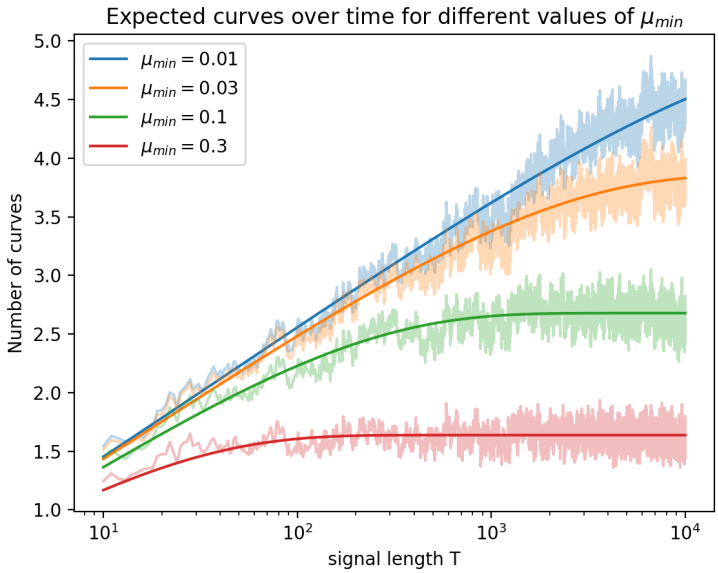


Figure 6.2.1: Mean number of curves in Gaussian-FOCuS using a sample size of 100 signals against the theoretical expected number of curves.

Theorem 6.2.6 (Bound on quadratics as $T \to \infty$)**.**

$$\mathbb{E}[\#\ \textit{quadratics in Gaussian FOCuS}] \leq \sum_{l=1}^{\infty} \frac{1}{l} \cdot \mathbb{P}\left(Z > \frac{\sqrt{l}\mu_{min}}{2}\right),$$

Proof. We have a simple bound for the Normal distribution

$$\mathbb{P}\left(Z \geq z\right) \leq \frac{e^{-z^2/2}}{z\sqrt{2\pi}}, \quad z > 0$$

showing that the series from Lemma 6.2.5 converges as $l \to \infty$ (e.g. due to being sub-geometric).

The extra factor of 2 is because in order to have no evidence for a change at intensity $\mu_{\min}$, the estimated mean of the segment must be less than $\mu_{\min}/2$ (and therefore closer to 0 than to $\mu_{\min}$). □

This bound is just a function of $\mu_{\min}$. It is not too hard to empirically compute with some code. For sanity checking, if we set $\mu_{\min} = 0$ (i.e. no minimum jump size) we have that $\mathbb{P}(Z > 0) = 1/2$ and we recover the harmonic series logarithmic bound.

Corollary 6.2.7 (Memory storage bounds as $h_{\max} \to \infty$)**.** *For a fixed sigma threshold, the expected memory storage required by Gaussian FOCuS to accurately assess anomalies in all intervals up to size h_{max} is $O(\log(h_{max}))$.*

Proof. Consider the effect on Theorem 6.2.6 of dividing $\mu_{\min}$ by 2. We have that for any decreasing sequence A_l where the series convergence occurs,

$$\sum_{l=1}^{\infty} \frac{A_l}{l} = A_1 + \frac{A_2}{2} + \frac{A_3}{3} + \sum_{i=0}^{3} \sum_{l=1}^{\infty} \frac{A_{4l+i}}{4l+i}$$

$$\leq A_1 + \frac{A_2}{2} + \frac{A_3}{3} + \sum_{i=0}^{3} \sum_{l=1}^{\infty} \frac{A_{4l}}{4l}$$

$$= A_1 + \frac{A_2}{2} + \frac{A_3}{3} + \sum_{l=1}^{\infty} \frac{A_{4l}}{l}.$$

Now, let $A_l = \mathbb{P}\left(Z > \frac{\sqrt{l}\mu_{\min}}{4}\right)$ such that $A_{4l} = \mathbb{P}\left(Z > \frac{\sqrt{l}\mu_{\min}}{2}\right)$. We have that $A_1, A_2, A_3 \leq 1/2$ so the first three terms sum to < 1, giving us that

$$\mathbb{E}[\#\text{ quadratics with } \mu_{\min}/2] < 1 + \mathbb{E}[\#\text{ quadratics with } \mu_{\min}]$$

That is to say, halving $\mu_{\min}$ adds at most one expected quadratic.

This means the expected number of quadratics is logarithmic in $1/\mu_{\min}$ or alternatively logarithmic in $h_{\max}$ for a fixed k-sigma threshold $\mu_{\min}^2 h_{\max} = k^2/2$.

Each quadratic is stored in memory as a pair of numbers: an integer $T - \tau + 1$ and a float $\sum_{t=\tau}^{T} x_t$, the size of both of which are probabilistically bounded at well below the limits for single-precision numbers (32 bits memory) for any possible practical application. Therefore the memory allocation required to store one quadratic does not grow with $h_{\max}$.

Putting these together gives that the total expected memory storage for Gaussian FOCuS is logarithmic in $h_{\max}$. □

This compares favourably with a window method, where just running one window

of size h over the data requires $O(h)$ memory allocation. This is because you must represent each point in the window individually, because advancing the window means removing the point furthest in the past.

6.2.2 Bounds for Poisson-FOCuS

Lemma 6.2.8. *Let X_i be standard Poisson(λ) distributed independent random variables and $l \in \mathbb{N}$, $\mu \geq 1$. Define $u := \sum_{i=1}^{l}(X_i - \lambda)$.*

$$\mathbb{P}\left[u/l \geq \lambda(\mu - 1)\right] = \mathbb{P}\left[\sum_{i=1}^{l} X_i \geq \mu l\lambda\right],$$

$$= \mathbb{P}\left[Poisson(l\lambda) \geq \mu l\lambda\right]$$

This means there is also a strict correspondence between the gradients of the unbiased random walk and the increasing means condition.

Theorem 6.2.9. *Let μ_{min} be the minimum parameter at which we are interested in the likelihood ratio test working. Define $\mu^* := \frac{\mu_{min}-1}{\log \mu_{min}}$ i.e. the minimal estimated interval mean we must keep.*

$$\mathbb{E}[\#\ curves\ in\ Poisson\text{-}FOCuS] \leq \sum_{l=1}^{\infty} \frac{1}{l} \cdot \mathbb{P}\left(Poisson(l\lambda) \geq \mu^* l\lambda\right)$$

We get good agreement with this when we look at empirical estimations of expected numbers of curves over time T for different values of $\mu_{\min}$, see Figure 6.2.2:

6.3 Multivariate problem setup

We now turn our attention to FOCuS in the multivariate setting.

6.3.1 Data

Our data signal $(\vec{x}_1, \vec{x}_2, ..., \vec{x}_T, ...)$ is a multivariate signal evolving through time. Each $\vec{x}_t := (x_t^1, ..., x_t^p)$ is a p-dimensional object, which may represent e.g. measurements

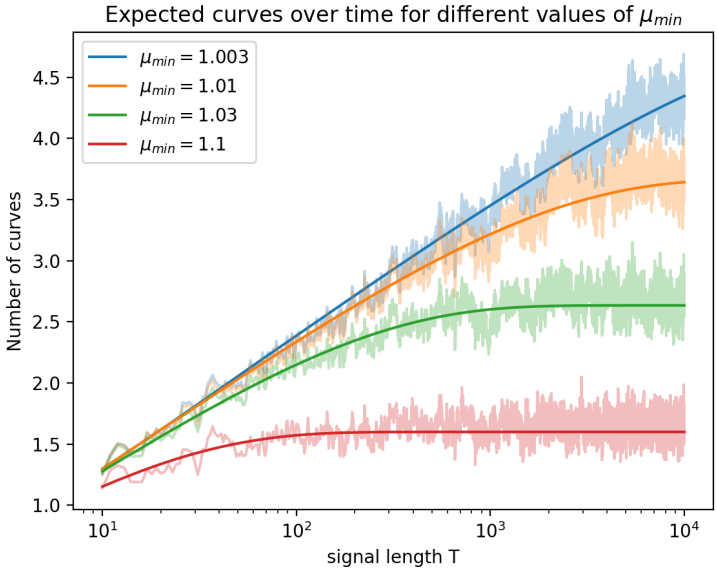


Figure 6.2.2: Mean number of curves in Poisson-FOCuS using a sample size of 100 signals against the theoretical expected number of curves.

from p different sensors at the same point t in time.

We denote by T the present time, such that at time T only the signal $\vec{x}_t : t \leq T$ has been observed. We are interested in algorithms that perform well when $T \to \infty$: we have been observing a signal for a long time, or the signal is high-velocity. We are also interested in algorithms that perform well when p is large: we have a high number of coordinates. In particular, we are very interested in the setting where both T and p are large simultaneously.

A startpoint τ affecting some subset $P \subset \{1, ..., p\}$ of coordinates is such that for $t \geq \tau, i \in P$ there has been a change in the underlying process used to produce the measurements which is now anomalous. We want to identify τ and P at the soonest possible point $T \geq \tau$ we are able to observe sufficient evidence. Defining $h := T - \tau + 1$ the current length of the anomaly at time T, for most practical applications we are interested in the case $h << T$, however as T is large h could also be large with this still being the case.

We will consider each x_t^i to be the realisation of a random variable X_t^i. Working in the Gaussian change in mean setting, we will model these random variables as independent

Normal random variables, that is

$$\vec{X}_t \sim \text{MVN}(0, I), \quad t \leq \tau$$

$$\vec{X}_t \sim \text{MVN}(\vec{\mu}, I), \quad t > \tau$$

where $\vec{\mu} := (\mu^1, ..., \mu^p)$ and $\mu^i = 0$ for $i \neq P$. We do not require the μ^i to be equal across coordinates; a change could affect one coordinate to a much greater extent than another, for example a sensor positioned much closer to the source of the observed anomaly. However, for some applications we may wish to impose a constraint that the μ^i all have the same sign (without loss of generality $\vec{\mu} \geq 0$).

While a pre-change mean of 0 and pre-change variance of 1 can be easily obtained by normalising each data stream in a pre-processing step (see Chapter 5 section 5.2.1 for some discussion of this), this assumption of an identity matrix I for the variance also contains the assumption that data streams are independent across variates. In practice there is often a positive correlation between different data streams, for example if they represent different sensor measurements of the same underlying process. For a known variance matrix Σ, one could reparameterise your dimensions by premultiplication by the matrix $\Sigma^{-1/2}$ to find a set of uncorrelated data streams. This will cause the signatures of anomalies in one data stream to spread out into all other data streams that it is correlated with, something which is in practice undesirable for locating anomalies. Tveten, Eckley, and Fearnhead (2022) describes a more complex way of handling dependence between data streams when the dependence must be estimated even in the possible presence of anomalies.

6.3.2 Test when τ and P are known

If τ and P are known, then we can perform a likelihood ratio test of the null hypothesis $\vec{\mu} = 0$ against the alternate hypothesis that $\mu^i \neq 0, i \in P$. This test has precisely $|P|$ degrees of freedom. Our test statistic, twice the log-likelihood ratio, has the following

form under the null hypothesis:

$$\sum_{i \in P} \frac{1}{T - \tau + 1} \left(\sum_{t=\tau}^{T} X_t^i \right)^2 \sim \chi^2_{|P|}.$$

We would reject the null hypothesis for values of this statistic over some threshold, which we would choose according to our desired probability of a false positive.

Both τ and P are unknown, which is the problem we wish to tackle.

6.4 Previous work

6.4.1 Testing all τ when $|P| = 1$: univariate FOCuS

We will define the statistical significance for one coordinate i and one start time τ as

$$S^i_{\tau:T} := \frac{1}{T - \tau + 1} \left(\sum_{t=\tau}^{T} x_t^i \right)^2 .$$

Under the null hypothesis, we have that each $S^i_{\tau:T} \sim \chi^2_1$, and the $S^i_{\tau:T}$ are independent across different coordinates i. If we only consider one coordinate, we are interested in finding the τ that maximises $S^i_{\tau:T}$.

The Functional Online Cumulative Sum (FOCuS) method (Romano, Eckley, Fearnhead, and Rigaill 2023) is a variant of the Page-CUSUM test that guarantees coverage of all possible μ via likelihood ratio test of $N(0, 1)$ against $N(\mu, 1),\ \mu > 0$. This can be thought of as equivalent to testing all possible startpoints $\tau \leq T$ at each timestep T. FOCuS reports a set of promising startpoints $(\tau_1, ..., \tau_n)$ and associated statistical significances, which is guaranteed to contain the startpoint $\tau \leq T$ with maximum statistical significance among all possible startpoints. The memory complexity is $n \sim O(\log(T))$ and the FOCuS method has been shown to be $O(1)$ in computational cost per iteration in order to evaluate the best possible startpoint. The case $\mu < 0$ can be handled by running FOCuS on the negation of the signal.

For practical applications the number of promising startpoints found by FOCuS can be restricted in two main ways:

1. Implementing a minimum intensity $\mu_{\min} > 0$ for the likelihood ratio test to ensure the removal of startpoints far in the past associated with a relatively constant, nonzero but very small statistical significance. We remove all startpoints τ that do not have intensity at least $\mu_{\min}$ on $[\tau, T]$.

2. Implementing a clearing parameter h_{clear} to ensure the removal of startpoints where there is clear evidence that the signal has since returned to baseline and the anomaly, while previously present, has been passed over. We remove all startpoints $\tau < T - h_{\text{clear}}$ that do not have intensity at least $\mu_{\min}$ (which we may set at 0) for at least one $[t, T], \quad t \geq T - h_{\text{clear}}$.

These two strategies are related to each other, in that they both discourage the retention of startpoints very far in the past. A lower bound on the intensity $\mu_{\min}$ means a linear increase in the amount of total statistical significance required for a startpoint to be maintained over time, requiring high total significance for retention over long periods. A clearing parameter h_{clear} means that as T advances further from τ, the startpoint τ must repeatedly clear independent significance checks over intervals of size h_{clear} (as well as all the intervals that overlap with these). The end result for both strategies is that under the null hypothesis of no change, the mean amount of retained startpoints becomes bounded, rather than rising logarithmically with signal length. For $\mu_{\min} > 0, \quad h_{\text{clear}} = \infty$ bound is given in Section 6.2.1, but is empirically fairly low (e.g. less than 10) for reasonable choices of $\mu_{\min}$ and h_{clear}.

6.4.2 Choosing P using local thresholds and anchoring: OCD

The Online Changepoint Detection (OCD) method (Chen, Wang, and Samworth 2022) is intended as a fast algorithm for the detection of both dense ($|P| \geq \sqrt{p}$) and sparse ($|P| < \sqrt{p}$) changes in multivariate data. It uses Page-CUSUM on a grid to identify promising τ in each coordinate (diagonal statistic), and then uses that τ to test other coordinates on the interval $[\tau, T]$ (dense and sparse off-diagonal statistics). OCD uses diagonal to mean the coordinate that has generated sufficient evidence for an anomaly

on some interval, and off-diagonal to refer to testing other coordinates in the same interval to look for further evidence of this anomaly. The coordinate used to find the interval is known as the anchor coordinate. OCD does not incorporate anchor coordinates into the offdiagonal statistic, which may lead to a loss of statistical power when detecting anomalies affecting small numbers of coordinates.

OCD handles sparsity by setting a lower bound a on the univariate statistical significance of $[\tau, T]$ needed for a coordinate to be considered to belong to P. Where $a = 0$, the regime searches for dense changes. Setting $a \sim \log(p)$ searches for sparse changes.

The OCD algorithm searching for dense changes has a computational cost of $O(p^2)$ in the number of coordinates p. This compares favourably with the number of possible subsets 2^p.

Choosing P when τ is known

Imagine we have a given interval $[\tau, T]$ and we would like to perform a statistical test on all subsets $P \subset \{1, ..., p\}$ of coordinates for this given interval. There are $2^p - 1$ nonempty subsets to check, and scanning them all individually would be computationally infeasible as p gets large.

One quick method to compute the best subset of size i, P_i is to sort each coordinate's significance over $[\tau, T]$ into descending order. Then P_1 is the coodinate with the greatest significance, P_2 is the two top coordinates, and so on until P_p is all coordinates. Sorting algorithms are only $O(p \log p)$ which makes this an attractive procedure.

OCD uses the strategy of using only one subset $P_{>a}$, which is all coordinates with significance over a on the interval $[\tau, T]$, where a is a user-defined local threshold parameter. We must have $P_{>a} = P_i$ for some i, so this is clearly sensible. Using $P_{>a}$ is advantageous if we do not know the size of the anomaly, as if an anomaly is present we would expect to coordinates it affects to collect individual statistical significances, and therefore eventually become included in $P_{>a}$. It also has a computational advantage, as we are not required to sort the significances.

However, OCD has an issue of not varying its global threshold according to the

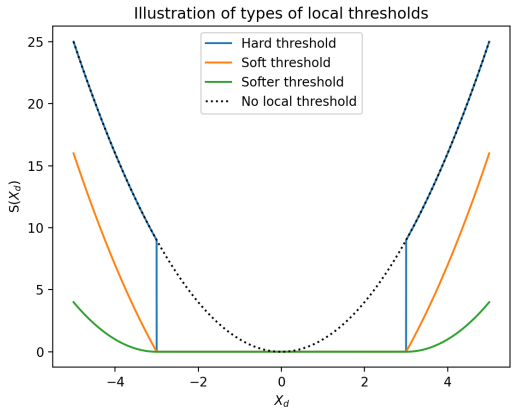


Figure 6.4.3: Three possible ways of applying a local threshold to one coordinate of a data stream.

current size of $P_{>a}$. This means that anomalies affecting small numbers of coordinates will be compared against a global threshold designed to account for the noise in large numbers of coordinates, and results in a loss of statistical power for detection of these anomalies.

Figure 6.4.3 gives an illustration of ways to address this by changing what is passed on after a local threshold is crossed. Subtracting the local threshold from the passed-on significance (orange line) or only starting to collect global significance once over the threshold (green line) are both ways to balance the significance of anomalies occurring in different numbers of data streams while using a constant global threshold.

The choice of a is important, with higher values of a favouring anomalies affecting fewer coordinates.

Local thresholds are particularly important because they limit the amount of transmission of data required between data streams. Often, a multivariate anomaly detection problem represents multiple sensors that are spread out across a physical monitoring network, connected to a central computer, and the cost of one sensor communicating with another is greater than the cost of computations being performed on one sensor because it must first transmit everything required to perform the computation to the central computer. Yang, Eckley, and Fearnhead (2024) gives a more in-depth discussion of this problem.

Choosing a local threshold

Choosing a local threshold a is essentially a coordinate selection procedure as we select those coordinates who have significance statistic above a to contribute to our global sum. We would (trivially) like the following to be true:

1. When the ith coordinate does not contain any anomaly, as often as possible we do not select it. This avoids the introduction of unwanted noise into our detection system.

2. When the ith coordinate does contain an anomaly, as often as possible we do select it.

To motivate some choices of a local threshold a, consider the fact that under the null the likelihood ratio statistic associated with any choice of interval $[\tau, T]$ is chi-squared distributed on one degree of freedom (i.e. the square of a standard normal distribution), as it is the case that

$$\frac{\sum_{t=\tau}^{T} x_t}{\sqrt{T-\tau+1}} \sim N(0,1).$$

The chi-squared distribution on one degree of freedom has mean exactly 1 and median approximately 0.5. Therefore a likelihood ratio statistic with value < 0.5 is showing negative evidence for an anomaly. It makes sense to exclude any coordinates showing negative evidence from our offdiagonal sum, so we can set $a \geq 0.5$.

Our offdiagonal sum across coordinates will look chi-squared on p degrees of freedom under the null hypothesis. The global significance threshold for a small p-value will rise by at least 1 when moving from $p-1$ to p degrees of freedom. Therefore setting $a \geq 1$ means that no coordinates will be included that don't give justification for the global threshold increase.

More generally, we can describe the behaviour of any fixed value of a in terms of quantiles of the χ_1^2 statistic, or alternatively sigma-thresholds. For example, setting a equal to the 98th percentile (approximately 5.41) means that under the null hypothesis

only 2% of coordinates will be included in the sum. This is a useful way of bounding the amount of transmission and centralised computation we do according to given constraints. Chen, Wang, and Samworth (2022) proved that setting $a \sim \log(p)$ would ensure that under the null, the expected amount of transmission would remain bounded even as $p \to \infty$, while also remaining suitable for detecting changes affecting up to $\sqrt{p}$ many coordinates.

Estimating P using a combination of local threshold and minimum parameter

Chen, Wang, and Samworth (2023) take the approach of estimating P by choosing a minimum $\mu_{\min}$ value and then (for up-changes) running the FOCuS procedure on the signal with $\mu_{\min}$ subtracted off, plus a local sparsity constraint (which they call d_1) for this subtracted run that plays the same role as a above. They provide theoretical guarantees that with high probability this estimate of P contains no noise coordinates, while coordinates containing anomalies of size $\mu > \mu_{\min}$ of sufficient duration will be contained in the estimate.

We can compare this with a method that includes coordinates in P if they are both have significance above the local threshold a and have mean $\bar{x} > \mu_{\min}$. Rearranging the significance calculations from the OCD-CI subtraction procedure means that we would place a coordinate in P under

- Our method if $\bar{x} > \max\left[\frac{a_1}{\sqrt{h}}, \mu_{\min}\right]$
- OCD-CI method if $\bar{x} > \frac{a_2}{\sqrt{h}} + \mu_{\min}$

for appropriately chosen a_1 or a_2 that are dependent on our desired amount of transmission. For a given desired level of transmission you would expect $a_1 > a_2$.

Our method has advantages for the correct detection of anomalies of intensities very close to $\mu_{\min}$ as soon as statistically possible. The OCD-CI method has a smoother transition between the significance-based threshold and the mean-based threshold. Both of them work very similarly in practice, as the main advance gained by implementing

them is to treat intervals with small $\bar{x}$ and large h as noise coordinates. Also, if we send $\mu_{\min} \to 0$ the methods become identical.

Choosing this method over the OCD-CI method would give the advantage of being able to pick a slightly higher $\mu_{\min}$ parameter without delaying the detection of anomalies of size above but close to $\mu_{\min}$. This higher $\mu_{\min}$ parameter would then reduce the number of startpoints kept in our algorithm, although not by very much (for example, doubling $\mu_{\min}$ removes less than one startpoint in expectation, as shown in Section 6.2.1).

6.4.3 Constructing comparators

We will use a method designated OCD*, a slightly modified variant of OCD to the one the creators propose incorporating the above ideas. Specifically, we do the three following things:

1. Instead of the Page-on-a-grid method to find startpoints, we implement FOCuS with a minimum $\mu_{\min}$ parameter equal to the smallest point on this grid, essentially constructing an infinitely fine grid with no maximum value rather than one multiplicatively increasing by $\sqrt{2}$ at each grid step to a maximum value. This may find some extra startpoints that were missed by the grid approach. It also allows us to send $\mu_{\min} \to 0$ if we want to without needing to use an 'infinite grid', which was previously computationally impossible.

2. We incorporate our anchor coordinate into its corresponding offdiagonal sum as is implicitly done in the support estimation for the OCD-CI paper rather than leaving it out as the original OCD paper does. This helps us find anomalies that affect more than one but only a few coordinates.

3. We use the maximum instead of the sum boundary in order to reduce detection delays for anomalies with means close to $\mu_{\min}$.

This allows us to bring the OCD algorithm in line with the conventions made by FOCuS-derived algorithms, such that the difference under investigation - how multivariate

coordinates are combined - stands as the only major difference between the algorithms.

We now have three comparator algorithms which are detecting as close as possible the same concept of an anomaly, but are using very different methodologies to combine coordinates.

1. Naive FOCuS: we run univariate FOCuS on each data stream and sum the results across streams.
2. OCD*, as defined above.
3. Geometric FOCuS (Pishchagina, Romano, Fearnhead, et al. 2023), a more complete generalisation of FOCuS intended for lower-dimensional multivariate settings such as for detection of changes in mean and variance.

These algorithms are organised in increasing order of computational complexity with respect to the number of coordinates p. Naive FOCuS is a simple $O(p)$ method. OCD*, the same as OCD itself, is an $O(p^2)$ method. Geometric FOCuS is an $O(\alpha^p)$ method where $\alpha \geq 2$ due to the complexity of the algorithm for its convex hull computation, and is presented here for completeness rather than practicality in the higher-dimensional setting.

6.5 Multivariate FOCuS

We are now able to formulate our question. Can we construct an $O(p)$ method that improves on Naive FOCuS by drawing on insights gained from other methods, while still retaining a low computational cost?

6.5.1 Startpoint selection across coordinates

If we are interested in testing intervals $[\tau, T]$ in a multivariate signal at the present time T, we first have to pick the τ. The best way to do this is not immediately clear, because different start times may be promising in different coordinates. If we label the best startpoint τ_i for each coordinate i, then for different coordinates i and j we may

have τ_i very far from τ_j. We would hope that in the presence of an actual anomaly affecting both i and j at the same time, we would have the τ_i "close to" τ_j. Therefore, checking both $[\tau_i, T]$ and $[\tau_j, T]$ for anomalies should be a good proxy for checking the best time $[\tau, T]$.

The OCD* algorithm uses offdiagonal tail coordinates. That is, we choose the best τ_i using Page's method (Page 1955) for one coordinate i, and then test $[\tau_i, T]$ for the other coordinates as well. This forces the startpoints to line up exactly, but the loop it requires is the source of why OCD* is an $O(p^2)$ method.

When deriving support estimates and confidence intervals for the startpoints for OCD, Chen, Wang, and Samworth (2023) show that, for a sharp change of intensity μ, the length of a good confidence interval for when that change started is $O(1/\mu^2)$. That is, for τ showing evidence for small changes, it is quite hard to tell from the noisy data exactly where the true change started, and we don't get any more evidence about good start points from observing the signal for longer (although we collect more certainty that a change is actually present).

Naive FOCuS would, when given a coordinate subset P, calculate the best τ_i for each i in P and then add up the significances for all the $[\tau_i, T]$ in P. That is, we allow anomalies to start at entirely different times in different coordinates, with no constraint whatsoever. If P really does contain an anomaly that starts in the same time across coordinates, this should appear in our test. However, the rise in the number of tests we are performing means that our false positive rate will be higher than we would like.

6.5.2 Testing for anomalies using summaries of data streams

Consider the following scenario: we have run FOCuS individually on each univariate data stream, giving us lists of promising start points and associated significances. We would like to construct a good test of the hypothesis that there is an anomaly that affects some subset P of coordinates but that starts at (roughly) the same time in each coordinate. To what extent is it possible to do this using not the whole dataset but only the outputs from FOCuS?

For a univariate signal with present T and some past time t, let us define the nearest prior startpoint $\tau_{(\leq t)}$ to t as the greatest $\tau \leq t$ recorded as a startpoint by FOCuS at time T. That is, we are not interested in any startpoints $\tau > t$ as any such anomaly $[\tau, T]$ would not contain t. We can rephrase univariate FOCuS at time T as a backscan for all $t \leq T$ of the significances of the intervals $S[\tau_{(\leq t)}, T]$, where if no such $\tau_{\leq t}$ exists (if t is before the first startpoint) we define the significance to be zero.

This gains us nothing in the univariate case. However, it illustrates one way to proceed in the multivariate case, as follows:

For a multivariate signal with present T and some past time t, let us define the nearest prior startpoint $\tau^i_{(\leq t)}$ for coordinate i similarly, as the greatest $\tau \leq t$ recorded as a startpoint by univariate FOCuS run on coordinate i at time T. We can then do a global backscan for all $t \leq T$ of the sum of the significances of the intervals $S_i[\tau^i_{(\leq t)}, T]$. If we are trying to select some subset $P_{>a}$, then we only sum the significances that are over our local threshold a.

There exists an algorithm to compute this backscan in only $O(p)$ complexity. This is a significant improvement for large p on the $O(p^2)$ complexity for the offdiagonal statistics approach. The algorithm requires us to have a tuple list L of the startpoints in all coordinates that are globally sorted in reverse time order and labelled by coordinate, that we use as our iterator. L consists of tuples (τ, i) containing startpoints and the coordinates that generated them, sorted by τ in descending order. We know that for one coordinate detecting anomalies up to a maximum window size h, the expected number of startpoints kept at each iteration is bounded at all times by $O(\log h)$, and if we have no maximum window size then it is bounded at time T by $O(\log T)$. Therefore the expected size of this list L is $O(p \log h)$ in the maximum window size h case (as is the appropriate comparator for OCD*) and $O(p \log T)$ if there is no maximum window size.

The algorithm is as follows:

1. Construct the sum of the significances $\sum_i S_i[\tau^i_{\leq T}, T]$ using the final startpoint in all coordinates. Test against threshold.

2. Starting with $t = T$, we now construct $\sum_i S_i[\tau^i_{\leq t-1}, T]$ by replacing $S_i[\tau^i_{\leq t}, T]$ by $S_i[\tau^i_{\leq t-1}, T]$ only in the coordinates i where they differ. These i are precisely the coordinates that appear in the elements (t, i) of L. Test against threshold.

3. Decrement t to the next $\tau < t$ that appears in the first elements of the tuples of L.

4. Repeat steps until we are through L.

Constructing L in the first place could be a challenge. However, sorting a list of size n is an $O(n \log n)$ operation, giving us at worst $O(p \log p)$ complexity in p for this algorithm if we were to recreate L from scratch every time. We do not need to do this, as univariate FOCuS never moves its start points, only ever deleting them or adding a startpoint at T. Therefore, as $T \to T + 1$ the elements of L can disappear entirely, but not switch their ordering. We only need to add the relevant (T, i) to the front of L and delete elements while iterating through if they are no longer needed. This means that updating L is only an $O(p)$ operation in p.

This method enforces a notion of nearness between the startpoints tested in different coordinates. In the case of a real anomaly beginning at the same time in all coordinates it affects, this nearness will be satisfied for those coordinates. However in the absence of an anomaly we won't test the most significant startpoints unless they are near each other, cutting down on the noise (see Figure 6.5.4). Here, by using a backscan to constrain where anomalies begin in different data streams, we can still pick up the anomaly well, gaining some statistical power.

In practice, the $\tau^i_{\leq T}$ can still be very far apart in each datastream. This means that our anomaly detection method still contains a large amount of additional noise. We offer two possible means of addressing this: enforcing a hard limit on how far apart the $\tau^i_{\leq T}$ can be in order to contribute to the same anomaly, and computing a distance discount for $\tau^i_{\leq T}$ that are far apart.

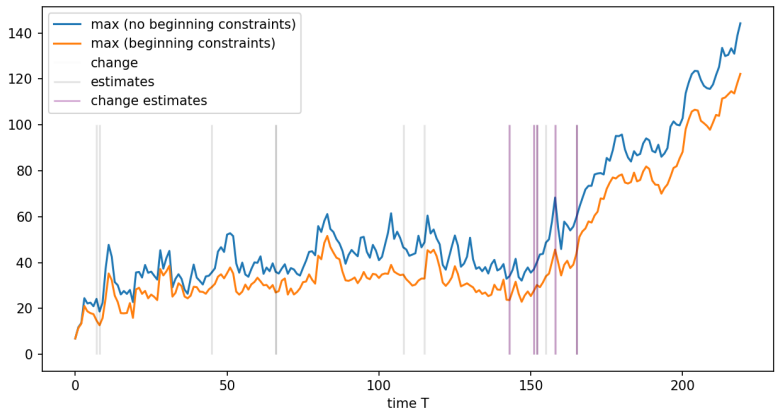


Figure 6.5.4: An anomaly beginning at time 150 is estimated in the coordinates it affects by the vertical purple lines. Estimates for coordinates it does not affect are given in grey.

6.5.3 Enforcing a hard limit on nearness

We may wish to set a hard limit on how far apart we test different startpoints for anomalies. That is, given a fixed overlap W, we only wish to test the startpoint $\tau_{(\leq t)}$ during the backscan at point t if $\tau_{(\leq t)} > t - W$. This requires that as we iterate through L we separate out adding the significances $S_i[\tau^i_{\leq t-1}, T]$ from the removal of the significances $S_i[\tau^i_{\leq t}, T]$ by constructing a separate list of startpoints to be added on a time delay.

One advantage of this way of working is that our parameter W is easily interpretable as the maximum lag between anomaly start times across different coordinates. We can therefore construct a reasonable estimate of a good W based on the kind of anomaly we are looking to search for (size, number of coordinates affected), using Chen, Wang, and Samworth (2023)'s work on global confidence intervals.

Consider the case of an anomaly beginning in the same time point τ across all coordinates. The estimated anomaly startpoint in each coordinate will be somewhere close to τ, but may be slightly before or slightly after. Therefore, if our real data conforms to our model, and we decide to implement a reasonable $\mu_{\min}$ in the jump size of the anomalies we are searching for, use of a constant W will pick up on all real anomalies with high probability.

An alternate method is to linearly increase the length of W as we backscan through the data stream. This is based on the idea that in order for an anomaly of mean μ and length h to be distinguishable from statistical noise at any given significance level, it must have $\mu^2 h$ above a threshold related to the significance level. Therefore the error estimation in the startpoint of an anomaly, of order $O(1/\mu^2)$ for a μ currently detectable, is also of order $O(h)$.

We would not want to use a W if we had reason to suspect that anomalies could legitimately begin at different time points in different data streams. This is the case in many real world applications, such as errors that cascade through a network if not fixed promptly.

6.5.4 Distance discounting

If we have some $\tau_k < \tau < \tau_{k+1}$, where τ_k and τ_{k+1} are adjacent startpoints stored in FOCuS (i.e. τ is not one of these), what is it possible to say about the significance of the interval $[\tau, T]$ given that we only have the summary statistics stored by FOCuS?

We are interested in this in the multivariate setting because we may have evidence from other coordinates that τ is a more correct starting place for an anomaly.

We know that we must have $\mu_{\tau_k:\tau} \geq \mu_{\tau:\tau_{k+1}}$, or else τ would be a startpoint in FOCuS by the ascending means criterion. Therefore we must also have $\mu_{\tau_k:\tau_{k+1}} \geq \mu_{\tau:\tau_{k+1}}$, where $\mu_{\tau_k:\tau_{k+1}}$ is a known quantity as both τ_k and τ_{k+1} are stored in FOCuS. We can use this to create an upper bound on the value of $\mu_{\tau,T}$, namely

$$\mu_{\tau,T} \leq \frac{1}{T-\tau}\left[(T-\tau_{k+1})\mu_{\tau_{k+1:T}} + (\tau_{k+1}-\tau)\mu_{\tau_k:\tau_{k+1}}\right].$$

This then can be used to create an upper bound $S^{\text{upper}}_{\tau:T}$ on the significance $S_{\tau:T} = (T-\tau)\mu^2_{\tau,T}$. This upper bound is shown as the orange line in Figure 6.5.5. Previously we only had the upper bound $S_{\tau:T} \leq \max(S_{\tau_k:T}, S_{\tau_{k+1}:T})$, which is shown in red on Figure 6.5.5. We also have the following proposition:

Proposition 6.5.1 (Linear upper bound)**.** *S^{upper} is piecewise convex between start-*

points, meaning in particular that it is sublinear, and the simpler linear upper bound

$$S_{\tau:T} \leq \frac{(\tau - \tau_k)S_{\tau_k:T} + (\tau_{k+1} - \tau)S_{\tau_{k+1}:T}}{\tau_{k+1} - \tau_k}$$

also applies.

Proof. We define the continuous function $s(\tau)$ on the domain $\tau \leq \tau_{k+1} < T$ as

$$s(\tau) = \frac{\left[(T - \tau_{k+1})\hat{\mu}_{\tau_{k+1:T}} + (\tau_{k+1} - \tau)\hat{\mu}_{\tau_k:\tau_{k+1}}\right]^2}{T - \tau}$$

At each of the integer time points $\tau \in \{\tau_k, \tau_k + 1, ..., \tau_{k+1}\}$, this is the significance $S_{\tau:T}$ at that point. We can represent this as $(a - b\tau)^2/(T - \tau)$ for the appropriate a, b. Differentiating with respect to τ we have that

$$s''(\tau) = \frac{2b^2}{T - \tau} + \frac{-4b(a - b\tau)}{(T - \tau)^2} + \frac{2(a - b\tau)^2}{(T - \tau)^3}$$

$$= \frac{2[(a - b\tau) - b(T - \tau)]^2}{(T - \tau)^3} > 0$$

Therefore $s(\tau)$ is a convex function as its second derivative is non-negative for all relevant τ.

The case $\tau > \tau_n$ more recently than the final quadratic is not covered by this reasoning and must be handled separately, but here we have that $S^{\text{upper}}_{\tau:T}$ is exactly the linear function $(T - \tau)\mu^2_{\tau_n,T}$.

□

Figure 6.5.5 shows the actual interval significances, and various upper bounds on them, from a selection of runs of univariate FOCuS on a randomly generated signal of size 500 without anomalies. In blue, we see the actual interval significance that would be reported by a method such as OCD* that kept and tested all relevant data, whereas the purple line refers to the Naive FOCuS method of considering the most significant startpoint to be relevant for all start times in that coordinate.

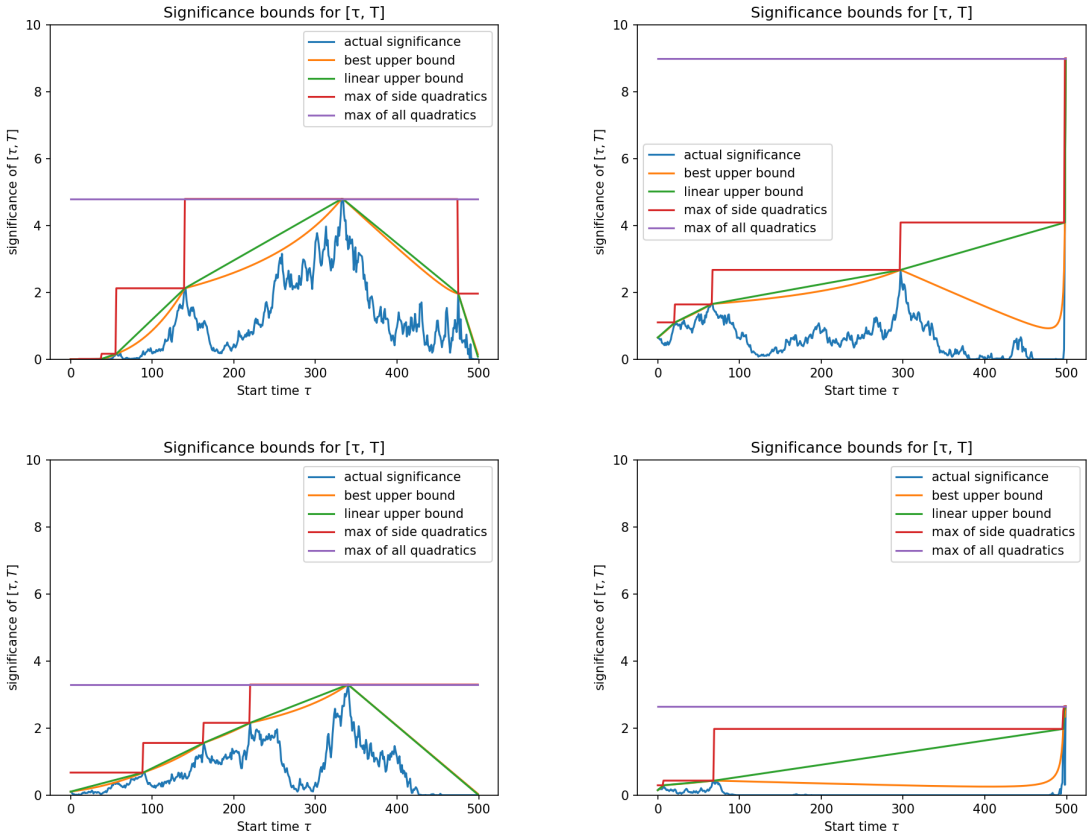


Figure 6.5.5: Significances and significance bounds for four runs of univariate FOCuS on a signal of size 500.

By performing a multivariate backscan as in section 6.5.2, we are able to improve from the purple bound to the red one by only considering the significance of nearby startpoints. Note that our backscan method of considering the $\tau^i_{(\leq t)}$ in each coordinate i is not aiming to test the significance of $[t, T]$, just to ensure that the $\tau^i_{(\leq t)}$ are all close to each other, and therefore there is some $t^* \leq t$ where we are testing the significance of $[t^*, T]$.

If we are prepared to accept a higher computational cost to discount by distance, for example if we have already performed appropriate coordinate selection using a backscan or a hard limit W, then the best possible upper bound on the significance would be given by the orange line S^{upper}. We could also use the linear upper bound in green if this became the computational bottleneck in our algorithm.

However, utilising either of these methods on the full data stream would require a loop that would make our overall computational cost $O(p^2)$, for the same reasons as OCD*: we are updating something about the significances of all other coordinates when we change τ in one. It may still be that this is a useful idea, for instance if we are working in a distributed setting where the limiting cost is communication between coordinates and the central processor (see Yang, Eckley, and Fearnhead (2024) for an example).

6.6 Summary

There are a variety of ways to search for collective anomalies in a multivariate data stream. These methods can have upsides and downsides, both statistically and computationally. If we knew a priori where our anomalies were, then the significance test for them is fairly simple: we sum the univariate significances, and compare to the quantile of a χ^2 statistic with the degrees of freedom equal to the number of coordinates being tested. However, when moving from the univariate to the multivariate setting, two main problems arise. These are coordinate selection, and selection of startpoints across coordinates. If these problems are not handled well, they introduce more noise into our

significance statistics and reduce the power of our detection method.

Both coordinate selection and selection of startpoints are made easier to handle by the introduction of a lower bound $\mu_{\min}$ of the absolute mean of the anomaly, over all coordinates it affects, even if $\mu_{\min}$ is small. For a given significance threshold, this is equivalent to introducing a maximum anomaly length $h_{\max}$ which is allowed to be quite large, and is therefore a feasible constraint for many real-world problems. However, there may be problems where we do not wish to have this constraint.

Coordinate selection is best handled by a choice of local significance threshold a, which can also be combined with the lower bound $\mu_{\min}$ on anomaly length in either a sum or a max way. In order to bound our false positive rate under the null as our number of coordinates $p \to \infty$, we must choose $a \sim \log(p)$. In the absence of such a $\mu_{\min}$, coordinate selection can also be done by considering if estimated startpoints in different data streams are close enough to each other in time to be referring to the same anomaly with high probability. This would only be valid on real data streams where we actually expect anomalies to begin at the same time in every coordinate.

Selection of startpoints across coordinates can be handled by testing all startpoints derived from each univariate data stream, in all other data streams that have passed coordinate selection. This is an $O(p^2)$ method, which may cause problems for large p. We have outlined various methods to instead combine startpoint information across coordinates to give an $O(p)$ method. The choice of methods is informed by statistical bounds on the distance between the true startpoint and its estimation if an anomaly is present, as well as the expected density of startpoints found under the null at different times in the past.

We can also implement distance discounting: a mathematical penalty on the significance of a coordinate if we assume we have estimated the startpoint incorrectly. This can cause computational problems when working with large numbers of startpoints so may be best combined with coordinate selection methods to first reduce the number of startpoints under consideration. This would also only be valid on real data streams where we actually expect anomalies to begin at roughly the same time in every

coordinate.

Our methods provide ways forward for a person wanting to apply multivariate anomaly detection methods on signals with both a long length and with a large number of coordinates p, where an assumed minimum anomaly size $\mu_{\min}$ may not exist, and where a more naive anomaly detection method is liable to being overwhelmed by the large amount of noise present in the signal.

Chapter 7

Conclusion

Schmidl, Wenig, and Papenbrock (2022)'s recent review of 158 methods for time series anomaly detection came to the conclusion that no method seems uniformly better than others, the performance of most methods is extremely sensitive to many of their parameters, and issues of robustness and scalability continue to dominate the concerns of the methods' users. This is in line with what has previously been said on the topic of classification more generally (Hand 2006), where neglecting important aspects of individual real-world problems in pursuit of an overall best method creates an illusion of progress in the field.

Finding the right anomaly detection method to solve your problem starts with picking the right definition of an anomaly for your problem. Choosing a method designed to find anomalies that are different from the anomalies you are looking for will always result in bad performance, even if the method is good. If you have a clear idea of what an anomaly is supposed to look like, translating this into a precise mathematical definition can often be fairly simple and also interpretable.

The methods developed in this thesis have been implemented in Python 3 and are intended to integrate with the rest of the scientific Python ecosystem. They form the basis of the open-source package changepoint_online which can be found on PyPI. They are scaleable, work well with many different types of pre-processing that can address robustness, and are good at detecting the types of anomalies they have been developed

to detect.

7.1 Summary of novel theoretical work

Here, we briefly recap and highlight three novel contributions to the academic literature provided by the work done within this thesis. We also describe the stages of research that led up to these novel theoretical developments, and the implications they may have elsewhere.

7.1.1 Extension to Poisson data form

The main challenge in developing a Poisson form of the FOCuS algorithm came in the definitions of the pre-change and post-change parameters, which were initially unclear. In the Gaussian form, if the pre-change mean and variance are known, then without loss of generality the signal can be rescaled to a standard Gaussian pre-change.

This is impossible with a Poisson pre-change parameter λ which influences the shape of the distribution and cannot be rescaled - there is no standard Poisson distribution. Adding on an additional μ for the anomaly creating a post-change parameter $\lambda+\mu$, as in a direct mimic of the Gaussian case, created challenges in the fact that the log-likelihood for different μ depended on λ in quite a complicated way:

$$\ell(x_T|\lambda+\mu) - \ell(x_T|\lambda) = x_T \log\left(\frac{\lambda+\mu}{\lambda}\right) + \mu$$

If you specify your expected $\sum_{t=\tau}^{T} \lambda_t$ and actual $\sum_{t=\tau}^{T} x_t$ counts as your fundamental data of interest, and attempt to construct your anomaly out of them, an additive definition for estimated anomaly size (actual minus expected count) has a fluctuating range $[-\sum_{t=\tau}^{T} \lambda_t, \infty)$ whereas the multiplicative definition (actual divided by expected count) has a constant range $[0, \infty)$. We also have that the log-likelihood when summed over all points in an interval divides neatly into the sum of an actual count x_T and an expected count λ times their appropriate transformations of μ, if and only if μ is a multiplicative rather than an additive parameter. These factors motivated the switch

to a multiplicative log-likelihood form for this problem:

$$\ell(x_T|\lambda \cdot \mu) - \ell(x_T|\lambda) = x_T \log \mu + \lambda(\mu - 1)$$

It was also very difficult to plot a significance graph of the anomaly height $\mu > 0$ versus the interval size h, because while in the Gaussian case we have the significance $\mu^2 h$, in the Poisson case the significance additionally depends on the background rate λ, which may even change in the interval under consideration, and so there is no such standard two-dimensional graph. Anomalies of the same amount stand out more when the background rate is lower.

When using a algorithm such as Page's method for Poisson data (Lucas 1985) to search for fixed sizes of anomaly, a multiplicative μ parameter is a problem. This is because if your background rate λ is changing, then the size of anomaly you are optimised to detect changes as well. However, we noted that when generalising Page's method to the functional space, this problem disappears. Searching for all $\lambda\mu$ where $\mu > 1$ is the same domain as searching for all $\lambda + \mu$ where $\mu > 0$, that is, all up-changes to something $> \lambda$. Even if we are implementing a minimum parameter $\mu_{\min}$, we simply choose this low enough that any fluctuations in background would not take its additive interpretation above the smallest anomaly size we wish to consider.

7.1.2 Algorithm improvements giving a constant cost per iteration

The FOCuS method covered in this thesis was originally proposed and developed by Romano, Eckley, Fearnhead, and Rigaill (2023) as an extension of the FPOP change-point method (Maidstone et al. 2017) to the anomaly detection setting. When I began work on this PhD, the computational state of FOCuS could be summed up as:

"FOCuS is not an online algorithm, as the number of quadratics to maximise per iteration can fluctuate and is unbounded." (Romano, Eckley, Fearnhead, and Rigaill 2023)

Here, the maximisation step is being referred to as the bottleneck, which is certainly

the case in non-Gaussian data forms where it requires computing logarithms. Additionally to this, the pruning step of FOCuS was based on computing the intervals in μ over which each quadratic contributed to the significance statistic, a computation which was also dependent on the number of quadratics. Changing FOCuS into a truly online algorithm required making both the pruning step and the maximisation step into constant computational costs that were not dependent on the number of quadratics.

This was because, in the version of FOCuS with unknown pre-change mean, the quadratics had three varying coefficients rather than two so it was unclear if they could be sorted.

Both of these required considering the time-series nature of the problem, by looking at the smaller intervals the startpoints associated with each quadratic subdivide the signal into, only one of which alters as the signal advances. Because only the final subinterval $[\tau_n, T]$ is modified as T advances, by rephrasing the algorithm as iterating over sums of subintervals we could give it a constant, low cost.

The trick of fitting an expanded model with more parameters that's easier to compute and comparing this inequality to a threshold is somewhat conceptually similar to the inequality-based pruning used for changepoint methods, for example the PELT changepoint algorithm (Killick, Fearnhead, and Eckley 2012), which also relies on the assumption that placing more changepoints always increases the overall model fit. However, there is a clear difference: while PELT's inequalities prune the set of changepoints under consideration, FOCuS does not remove startpoints by fitting the expanded model, using it instead to avoid iterating over them. It is possible that the trick presented here could be of use more widely to reduce the computational cost (but not the memory cost) of other changepoint methods.

7.1.3 Bound on expected number of startpoints present in FOCuS with $\mu_{\min}$

The bound on the expected number of startpoints printed at the beginning of Chapter 6 was derived quite late in the PhD. The appendix for Chapter 3 contains a proof that

there is a finite bound, based on the idea that resetting FOCuS's memory entirely due to $\mu_{\min}$ takes a finite amount of time, but this is a very loose bound.

This bound also serves as an upper bound on the number of distinct startpoints kept when implementing a Page-on-a-grid scheme. As all such schemes have a finite number of grid values, they have a smallest positive grid value, and setting this value as $\mu_{\min}$ ensures that all grid values are therefore in the range $[\mu_{\min}, \infty)$. All startpoints from this grid would be kept by FOCuS (although FOCuS may keep additional startpoints not on this grid). Assuming the grid is sufficiently fine that only a negligible amount of startpoints are lost between the grid values, the bound will be fairly tight.

One example of such an algorithm is Online Changepoint Detection (OCD) (Chen, Wang, and Samworth 2022), whose author states about the difference between the number of points on its multivariate grid $\mathcal{B}$ and the number of distinct interval sizes giving rise to its set of startpoints T:

"In fact, the computational complexity of ocd can often be reduced, because typically $T := \{t_b^j : j \in [p], b \in \mathcal{B}\}$ *has cardinality much less than* $p\mathcal{B}$*. ... It appears to be difficult to provide theoretical guarantees on* $|T|$*. Nevertheless, we have implemented the algorithm in this form in the R package ocd, and have found it to provide substantial computational savings in practice."* Chen, Wang, and Samworth (2022)

This theoretical guarantee will also apply to all other algorithms that use Page-on-a-grid schemes.

7.2 Summary of applications and collaborations

Here, I highlight three collaborations that developed over the course of this PhD and contributed to this thesis in the application areas of telecommunications, astrophysics, and nuclear radiation monitoring.

7.2.1 British Telecom (BT)'s collaboration with STOR-i

This PhD was jointly funded by BT and EPSRC and was at the project outset provisionally titled "Novel Anomaly Detection Methods for Telecommunication Data Streams".

Previous collaborations between BT and STOR-i had led to the development of other collective anomaly detection methods, notably Collective And Point Anomalies (CAPA) developed by Fisch, Eckley, and Fearnhead (2022) and its sequential form, SCAPA (Fisch, Bardwell, and Eckley 2022) and multivariate extensions (Fisch, Eckley, and Fearnhead 2021; Tveten, Eckley, and Fearnhead 2022). CAPA was determined to be a useful tool that flexibly met BT's anomaly detection requirements. At the outset of this project, the main issue with CAPA was that:

"CAPA infers collective and point anomalies by solving a set of dynamic programme recursions. However both the computational cost of each recursion, and the storage cost, increase linearly in the total number of observations. This is unsuitable for the online setting in which both storage and computational resources are finite. In practice, this problem can be surmounted by imposing a maximum length m for collective anomalies. ... As one might anticipate, within this setting long anomalous segments with low signal strength would not be detectable any more as a result of the approximation." (Fisch, Bardwell, and Eckley 2022)

As stated above, CAPA suffers from what this thesis has referred to as the interval search problem. That is, it takes an $O(m)$ computational and memory cost per iteration when looking for anomalies up to size m. The main aim of this PhD was working with BT to develop a method able to bypass this limitation and therefore handle processing larger volumes of data using fewer resources in situations where this is necessary.

The main weakness FOCuS has when compared to CAPA is that CAPA's dynamic programming recursions can fit multiple anomalous intervals to the past signal. This means that it can accurately detect when anomalies have ended, and start searching for new anomalies immediately afterwards. In contrast, FOCuS in its basic form takes a long time to reset when passing over a large anomaly, and loses detection power

during that shadow period. It was the addressing of this weakness that provided the main motivation for the clearing parameter h_{clear} which is described in Chapter 4. However, it should be noted that CAPA's ability to fit multiple anomalous segments simultaneously does make it more effective when we are interested in the exact nature of what anomalies do and how they end, rather than only wanting to know if they have ended so we can reset our algorithm.

BT's data is often either well-approximated by a Gaussian distribution (after appropriate pre-processing has been done), or it is sufficiently non-parametric as to be not well-approximated by any distribution. The generalisation of FOCuS to the Exponential family in Chapter 5 was used in particular to allow use of FOCuS applied to data following a Binomial distribution. This allowed it to be combined with the use of the quantile-based binomial likelihood Non-Parametric UNbounded Changepoint (NUNC) method developed by Austin et al. (2023). NUNC operates on a sliding window but is able to detect changes in the empirical distribution function of the data. The combination of these two approaches produced NP-FOCuS (Romano, Eckley, and Fearnhead 2024), which is able to find changes in the empirical distribution function of the data on all window sizes.

7.2.2 Gamma-ray bursts and the HERMES group

Approximately two months into my PhD, I was contacted via my supervisors by Giuseppe Dilillo, then an astrophysics PhD student with the HERMES (High Energy Rapid Modular Ensemble of Satellites) group (Fiore et al. 2020). The group were looking for a way to more efficiently allocate their resources on-board the satellite to test incoming photon signals for the presence of gamma-ray bursts. They were already aware that what they referred to as 'diagonal methods' (based on Page's method) could be a better use of limited computational resources than a sliding window grid, and wanted help with the mathematics of this. The collaboration involved the use of FOCuS in a Poisson setting, which could be thought of as an optimal diagonal method, and is applied to this context in Chapter 3.

Much of the work done by the HERMES team involved finding an appropriate way to estimate the background rate. We realised that, as FOCuS is essentially a changepoint-derived method that is explicitly designed to pick up significance into the infinite horizon, it is particularly non-robust to small biases in background rate estimation that occur over long periods of time. The choice of a biased moving window in the background estimation part of the paper forming Chapter 3 is a choice specifically made to exacerbate this bias problem, so we can explore how it manifests and justify the ways in which use of a $\mu_{\min}$ parameter within FOCuS can render the algorithm more robust to it.

In reality, the HERMES team ended up searching around a large number of possible background estimation methods to attempt to eliminate this bias to the greatest degree possible. The least complex method that worked well turned out to be using double exponential smoothing (Dilillo, Ward, et al. 2024), which is a method of online trend estimation designed to track linear trends by using a bias correction parameter. Double exponential smoothing has a problem where it does not track curvature in the signal very well, although this can be mitigated somewhat by updating the estimated bias correction parameter more intensely than you would if you were expecting to only estimate linear trends. Therefore in addition to this, a more complex method based on training a neural network on various satellite and signal data in order to estimate the current background rate was developed to support the HERMES mission (Crupi, Ward, et al. 2023).

7.2.3 The NuSec Sigma Data Challenge

In the middle of the PhD, someone in the audience of one of my presentations pointed out that work on a signal of gamma-ray photons was directly translatable from the space setting to the ground radiation setting, and that the computational constraints of a satellite were similar to the constraints on battery life of a small, handheld radiation detector. This led to an application to the Nuclear Security Science Network (NuSec) for a three-month project under the NuSec Sigma Data Challenge (NuSec 2022). I

was particularly interested in how to ensure that large anomalies in this dataset, which represented events known to not be of interest, did not overshadow the ability to detect smaller ones. This led to the technical report that forms Chapter 4, which has cleared NuSec's permission to publish process for a research audience.

7.3 Future Work

To conclude, I highlight three possible areas of future work that draw on the research presented in this thesis.

First, the application of FOCuS to a general multivariate setting very much lies unfinished. In some ways, there is no 'general' multivariate setting: how the variates interact with each other varies widely by specific application. Cross-correlations between different data streams have yet to be addressed, especially in the setting where we wish to minimise the communication cost of sending data between streams. A more detailed study of the best computational outcomes when working on a real dataset can inform the best choice of what methods to use to combine data across different streams.

Second, the count data work may have wider applications in other subfields beyond the ones I have directly worked on. The work on handheld ground detectors may have use in the field of gamma-ray borehole monitoring, and the work on satellites may be useful for the advance detection of space weather. Some of these generalisations may require an adapting of the algorithm to handle temporal dependence in count data, using models such as a Hawkes process (Hawkes 1971).

Finally, I believe that methods such as FOCuS may have their part to play within the wider streaming data processing setting in applications such as computer vision. FOCuS works well as a component of a wider anomaly detection system. It essentially turns intervals into points. Methods such as neural networks operate on (multivariate) points, and do not process intervals in a time-intelligent way. Adapting FOCuS to fit within a more complex computing architecture such as a neural network may allow the generalisation of computer vision algorithms developed for still images to be able to

scalably handle video.

On a wider note, the rise of large language models and their subsequent use in many different settings has given rise to a new category of streaming data: AI-generated text. Previously, the amount of text in use for any application was limited to what could be written by a human, but this is no longer the case, creating large AI-generated text repositories and new possible anomaly categories based on natural language.

Large language models currently require far larger amounts of processing power than methods such as FOCuS, as they need a number of floating-point operations proportional to the total number of parameter weights in the model to generate one output token, where a query response can be hundreds of tokens. The number of parameter weights in most large language models currently runs into the hundreds of billions, and the consequent amount of processing power needed requires specialist computational architecture setups with computations performed on central servers (Vries 2023). This makes large language models not a viable competitor to FOCuS and other similar anomaly detection methods in any application where computational resources are constrained, such as those discussed in this thesis. Recent advances in large language model architecture lean increasingly towards chain of thought reasoning and other forms of increased per-query computation, compounding this problem.

In parallel to the rise of large language models, the amount of computational resources devoted to streaming data processing has now become very large, and this is opening another active area of research in computational cost reduction. Anomaly detection methods that can handle streamed natural language may have a part to play in this new area, particularly methods that are able to flexibly integrate with large language model processing to reduce its computational cost.

Bibliography

Abbott, Benjamin P, Richard Abbott, et al. (2017). "Multi-messenger observations of a binary neutron star merger". In: *The Astrophysical Journal* 848.2, p. L12.

Abramson, Josh et al. (Jan. 2011). "Convex minorants of random walks and Lévy processes". en. In: *Eff. Clin. Pract.* 16.none, pp. 423–434.

Andersen, Erik Sparre (Dec. 1954). "On the fluctuations of sums of random variables II". In: *Mathematica Scandinavica* 2.2, p. 194.

Andreou, Elena and Eric Ghysels (2002). "Detecting multiple breaks in financial market volatility dynamics". In: *Journal of Applied Econometrics* 17.5, pp. 579–600.

Anscombe, F J (Dec. 1948). "The transformation of Poisson, Binomial and Negative-binomial data". In: *Biometrika* 35.3/4, p. 246.

Areepong, Yupaporn and Wilasinee Peerajit (Feb. 2022). "Integral equation solutions for the average run length for monitoring shifts in the mean of a generalized seasonal ARFIMAX(P, D, Q, r)s process running on a CUSUM control chart". In: *PLoS One* 17.2, e0264283.

Atkinson, A C and D M Hawkins (Dec. 1981). "Identification of Outliers". In: *Biometrics* 37.4, p. 860.

Audibert, Julien et al. (Aug. 2020). "USAD: UnSupervised Anomaly Detection on Multivariate Time Series". In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '20. Virtual Event, CA, USA: Association for Computing Machinery, pp. 3395–3404.

Aue, Alexander and Claudia Kirch (2024). "The state of cumulative sum sequential change point testing seventy years after Page". In: *Biometrika*, To appear.

Austin, Edward et al. (Jan. 2023). "Online non-parametric changepoint detection with application to monitoring operational performance of network devices". en. In: *Comput. Stat. Data Anal.* 177.107551, p. 107551.

Axelsson, M, E Bissaldi, N Omodei, et al. (June 2019). "A decade of gamma-ray Bursts observed by Fermi-LAT: the second GRB catalog". en. In: *The Astrophysical Journal* 878.1, p. 52.

Bailes, M andothers (Apr. 2021). "Gravitational-wave physics and astronomy in the 2020s and 2030s". en. In: *Nature Reviews Physics* 3.5, pp. 344–366.

Bandaragoda, Tharindu R et al. (Dec. 2014). "Efficient Anomaly Detection by Isolation Using Nearest Neighbour Ensemble". In: *2014 IEEE International Conference on Data Mining Workshop*. IEEE, pp. 698–705.

Barrett, Edd et al. (2017). "Virtual machine warmup blows hot and cold". In: *Proceedings of the ACM on Programming Languages* 1.OOPSLA, pp. 1–27.

Basseville, Michele and Igor V Nikiforov (1993). *Detection of abrupt changes: theory and application*. Prentice Hall Englewood Cliffs.

Bateman, H (1910). "The solution of a system of differential equations occurring in the theory of radioactive transformations". In: *Proc. Cambridge Philos. Soc.*

Baudat, G and F Anouar (Oct. 2000). "Generalized discriminant analysis using a kernel approach". en. In: *Neural Comput.* 12.10, pp. 2385–2404.

Beaulieu, Claudie and Rebecca Killick (2018). "Distinguishing trends and shifts from memory in climate data". In: *Journal of Climate* 31.23, pp. 9519–9543.

Bentley, Jon Louis (Sept. 1975). "Multidimensional binary search trees used for associative searching". In: *Commun. ACM* 18.9, pp. 509–517.

Bhat, Narayana (2021). *Fermi Gamma-ray Burst Monitor Untriggered GBM Short GRB Candidates*. URL: https://gammaray.nsstc.nasa.gov/gbm/science/sgrb_search.html (visited on 05/18/2022).

Bloser, Peter F et al. (2022). "CubeSats for Gamma-Ray Astronomy". In: *arXiv preprint arXiv:2212.11413*.

Borucki, William J et al. (Feb. 2010). "Kepler planet-detection mission: introduction and first results". en. In: *Science* 327.5968, pp. 977–980.

Breunig, Markus M et al. (May 2000). “LOF: identifying density-based local outliers”. In: *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*. New York, NY, USA: ACM, pp. 93–104.

Brown, Robert G and Richard F Meyer (Oct. 1961). “The fundamental theorem of exponential smoothing”. en. In: *Oper. Res.* 9.5, pp. 673–685.

Cai, Deng, Xiaofei He, and Jiawei Han (Feb. 2011). “Speed up kernel discriminant analysis”. In: *VLDB J.* 20.1, pp. 21–33.

Campana, Riccardo et al. (2020). “The HERMES-TP/SP background and response simulations”. In: *Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray*. Vol. 11444. SPIE, pp. 817–824.

Chakar, Souhil et al. (2017). “A robust approach for estimating change-points in the mean of an AR(1) process”. In: *Bernoulli* 23.2, pp. 1408–1447.

Chandola, Varun, Arindam Banerjee, and Vipin Kumar (July 2009). “Anomaly detection: A survey”. In: *ACM Comput. Surv.* 41.3, pp. 1–58.

Chen, Yudong, Tengyao Wang, and Richard J Samworth (Feb. 2022). “High-dimensional, multiscale online changepoint detection”. en. In: *J. R. Stat. Soc. Series B Stat. Methodol.* 84.1, pp. 234–266.

— (May 2023). “Inference in high-dimensional online changepoint detection”. en. In: *J. Am. Stat. Assoc.*, pp. 1–12.

Chen, Zhanshou and Zheng Tian (2010). “Modified procedures for change point monitoring in linear models”. In: *Mathematics and Computers in Simulation* 81.1, pp. 62–75.

Cho, Haeran and Piotr Fryzlewicz (2015). “Multiple-change-point detection for high dimensional time series via sparsified binary segmentation”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 77.2, pp. 475–507.

— (2020). “Multiple change point detection under serial dependence: Wild energy maximisation and gappy Schwarz criterion”. In: *arXiv preprint arXiv:2011.13884*.

Chu, Chia-Shang J, Kurt Hornik, and Chung-Ming Kaun (1995). “MOSUM tests for parameter constancy”. In: *Biometrika* 82.3, pp. 603–617.

Chumash, Tzvi (2006). “Obtaining five “ nines ” of availability for Internet services”. In.

Connolly, Euan, Dean Connor, and Peter Martin (2023). "Location and Activity Characterization of Gamma-Ray Point Sources Concealed in Shipping Containers Using Iterative Reconstruction and Modeling Cargo-Specific Attenuation". In: *Nuclear Technology* 209.9, pp. 1382–1397. DOI: 10.1080/00295450.2023.2198473.

Crupi, Riccardo, Kes Ward, et al. (Nov. 2023). "Searching for long faint astronomical high energy transients: a data driven approach". In: *Exp. Astron.*

Dey, N et al. (2018). *Internet of Things and Big Data Analytics Toward Next-Generation Intelligence*. Springer.

Dilillo, Giuseppe, Kes Ward, et al. (Feb. 2024). "Gamma-Ray Burst Detection with Poisson-FOCuS and Other Trigger Algorithms". en. In: *ApJ* 962.2, p. 137.

Ding, Zhiguo and Minrui Fei (Jan. 2013). "An Anomaly Detection Approach Based on Isolation Forest Algorithm for Streaming Data using Sliding Window". In: *IFAC Proceedings Volumes* 46.20, pp. 12–17.

Eichinger, Birte and Claudia Kirch (2018). "A MOSUM procedure for the estimation of multiple random change points". In: *Bernoulli* 24.1, pp. 526–564.

Elísio, Soraia C et al. (Nov. 2023). "Point-spread analysis of γ-ray/depth spectra for borehole monitoring applications". In: *IEEE Trans. Nucl. Sci.* 70.11, pp. 2506–2514.

Enikeeva, Farida and Zaid Harchaoui (2019). "High-dimensional change-point detection under sparse alternatives". In: *The Annals of Statistics* 47.4, pp. 2051–2079.

Ester, M et al. (Aug. 1996). "A density-based algorithm for discovering clusters in large spatial databases with noise". In: *KDD*, pp. 226–231.

Fenimore, EE et al. (2003). "The Trigger algorithm for the burst alert telescope on SWIFT". In: *AIP Conference Proceedings*. Vol. 662. 1. American Institute of Physics, pp. 491–493.

Feroci, Marco, Filippo Frontera, et al. (1997). "In-flight performances of the BeppoSAX gamma-ray burst monitor". In: *EUV, X-Ray, and Gamma-Ray Instrumentation for Astronomy VIII*. Vol. 3114. SPIE, pp. 186–197.

Fiore, Fabrizio et al. (2020). *High Energy Rapid Modular Ensemble of Satellites scientific pathfinder*. URL: https://www.hermes-sp.eu (visited on 08/03/2021).

Fiore, Fabrizio, Luciano Burderi, et al. (2020). "The HERMES-technologic and scientific pathfinder". In: *Space Telescopes and Instrumentation 2020: Ultraviolet to Gamma Ray*. Vol. 11444. SPIE, pp. 214–228.

Fisch, Alex, Lawrence Bardwell, and Idris A Eckley (Aug. 2022). "Real time anomaly detection and categorisation". en. In: *Stat. Comput.* 32.4.

Fisch, Alex, Idris A Eckley, and Paul Fearnhead (2021). "Subset multivariate collective and point anomaly detection". In: *Journal of Computational and Graphical Statistics*, pp. 1–12.

— (Aug. 2022). "A linear time method for the detection of collective and point anomalies". en. In: *Stat. Anal. Data Min.* 15.4, pp. 494–508.

Fisher, R A (Sept. 1936). "The use of multiple measurements in taxonomic problems". en. In: *Ann. Eugen.* 7.2, pp. 179–188.

Friedman, Jerome H (Mar. 1989). "Regularized Discriminant Analysis". In: *J. Am. Stat. Assoc.* 84.405, pp. 165–175.

Gan, Junhao and Yufei Tao (May 2015). "DBSCAN Revisited: Mis-Claim, Un-Fixability, and Approximation". In: *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. SIGMOD '15. New York, NY, USA: Association for Computing Machinery, pp. 519–530.

Garg, Astha et al. (June 2022). "An Evaluation of Anomaly Detection and Diagnosis in Multivariate Time Series". en. In: *IEEE Trans Neural Netw Learn Syst* 33.6, pp. 2508–2517.

Gehrels, Neil et al. (1993). "The Compton Gamma Ray Observatory". In: *Scientific American* 269.6, pp. 68–77.

Grubbs, Frank E (Mar. 1950). "Sample Criteria for Testing Outlying Observations". en. In: *Ann. Math. Stat.* 21.1, pp. 27–58.

Guépin, Claire, Kumiko Kotera, and Foteini Oikonomou (Sept. 2022). "High-energy neutrino transients and the future of multi-messenger astronomy". en. In: *Nature Reviews Physics* 4.11, pp. 697–712.

Hallgren, Karl L, Nicholas A Heard, and Niall M Adams (2021). "Changepoint detection in non-exchangeable data". In: *arXiv preprint arXiv:2111.05054*.

Hampel, Frank R (Dec. 1971). "A General Qualitative Definition of Robustness". In: *Ann. Math. Stat.* 42.6, pp. 1887–1896.

— (June 1974). "The influence curve and its role in robust estimation". en. In: *J. Am. Stat. Assoc.* 69.346, pp. 383–393.

Han, Songqiao et al. (June 2022). "ADBench: Anomaly Detection Benchmark". In: *Adv. Neural Inf. Process. Syst.* abs/2206.09426.

Hand, David J (June 2006). "Classifier technology and the illusion of progress". In: *Stat. Sci.* 21.1.

Harter, Richard (2009). *The minimum on a sliding window algorithm*. URL: https://richardhartersworld.com/slidingmin (visited on 09/06/2023).

Hawkes, Alan G (Apr. 1971). "Spectra of some self-exciting and mutually exciting point processes". In: *Biometrika* 58.1, p. 83.

He, H and E A Garcia (Sept. 2009). "Learning from imbalanced data". In: *IEEE Trans. Knowl. Data Eng.* 21.9, pp. 1263–1284.

HEASARC (2022a). *Fermi GBM daily data*. URL: https://heasarc.gsfc.nasa.gov/W3Browse/fermi/fermigdays.html (visited on 06/07/2022).

— (2022b). *Fermi GBM trigger catalog*. URL: https://heasarc.gsfc.nasa.gov/W3Browse/fermi/fermigtrig.html (visited on 06/07/2022).

Hochenbaum, Jordan, Owen S Vallis, and Arun Kejariwal (Apr. 2017). "Automatic Anomaly Detection in the Cloud Via Statistical Learning". In: arXiv: 1704.07706 [cs.LG].

Huber, Peter J (Mar. 1964). "Robust estimation of a location parameter". In: *Ann. Math. Stat.* 35.1, pp. 73–101.

— (2004). *Robust Statistics*. en. John Wiley & Sons.

Hundman, Kyle et al. (July 2018). "Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '18. London, United Kingdom: Association for Computing Machinery, pp. 387–395.

International Atomic Energy Agency (Mar. 2024). *Radioactive Waste Management: Solutions for a Sustainable Future*. en. International Atomic Energy Agency.

Iwata, Takuma et al. (2018). "Accelerating online change-point detection algorithm using 10 GbE FPGA NIC". In: *European Conference on Parallel Processing*. Springer, pp. 506–517.

Jackson, B et al. (Feb. 2005). "An algorithm for optimal partitioning of data on an interval". In: *IEEE Signal Process. Lett.* 12.2, pp. 105–108.

Jolliffe, Ian T and Jorge Cadima (Apr. 2016). "Principal component analysis: a review and recent developments". en. In: *Philos. Trans. A Math. Phys. Eng. Sci.* 374.2065, p. 20150202.

Juhola, M, J Katajainen, and T Raita (1991). "Comparison of algorithms for standard median filtering". In: *IEEE Trans. Signal Process.* 39.1, pp. 204–208.

Killick, R, P Fearnhead, and I A Eckley (Dec. 2012). "Optimal Detection of Changepoints With a Linear Computational Cost". In: *J. Am. Stat. Assoc.* 107.500, pp. 1590–1598.

Klebesadel, R W, I B Strong, et al. (1973). "Observations of gamma-ray bursts of cosmic origin". In: *The Astrophysical Journal* 182, pp. L85–L88.

Kouveliotou, Chryssa et al. (Aug. 1993). "Identification of two classes of gamma-ray bursts". In: *Astrophys. J.* 413, p. L101.

Kriegel, Hans-Peter, Matthias Schubert, and Arthur Zimek (Aug. 2008). "Angle-based outlier detection in high-dimensional data". In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '08. Las Vegas, Nevada, USA: Association for Computing Machinery, pp. 444–452.

Kumar, Pawan and Bing Zhang (2015). "The physics of gamma-ray bursts & relativistic jets". In: *Physics Reports* 561, pp. 1–109.

Lai, Kwei-Herng et al. (June 2021). "Revisiting Time Series Outlier Detection: Definitions and Benchmarks".

Lavielle, Marc and Eric Moulines (2000). "Least-squares estimation of an unknown number of shifts in a time series". In: *Journal of Time Series Analysis* 21.1, pp. 33–59.

Lavin, A and S Ahmad (Dec. 2015). "Evaluating Real-Time Anomaly Detection Algorithms – The Numenta Anomaly Benchmark". In: *2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*, pp. 38–44.

Lee, M H and Michael B C Khoo (Sept. 2006). "Optimal Statistical Design of a Multivariate EWMA Chart Based on ARL and MRL". In: *Communications in Statistics - Simulation and Computation* 35.3, pp. 831–847.

Leys, Christophe et al. (July 2013). "Detecting outliers: Do not use standard deviation around the mean, use absolute deviation around the median". en. In: *J. Exp. Soc. Psychol.* 49.4, pp. 764–766.

Li and Ma (Sept. 1983). "Analysis methods for results in gamma-ray astronomy." In: *The Astrophysical Journal* 272, pp. 317–324. DOI: 10.1086/161295.

Li, Zheng, Yue Zhao, et al. (Sept. 2020). "COPOD: Copula-based Outlier Detection". In: *arXiv preprint arXiv:2009.09463*. arXiv: 2009.09463 [stat.ML].

Liu, A Moore, and Alex G Gray (Dec. 2006). "New algorithms for efficient high-dimensional nonparametric classification". In: *J. Mach. Learn. Res.*, pp. 265–272.

Liu, Kai Ming Ting, and Zhi-Hua Zhou (Dec. 2008). "Isolation Forest". In: *2008 Eighth IEEE International Conference on Data Mining*. IEEE, pp. 413–422.

Lorden, G (Dec. 1971). "Procedures for Reacting to a Change in Distribution". en. In: *aoms* 42.6, pp. 1897–1908.

LSST: From Science Drivers to Reference Design and Anticipated Data Products (2014). en. United States. Department of Energy. Office of Science.

Lucas, James M (May 1985). "Counted Data CUSUMs". In: *Technometrics* 27.2, pp. 129–144.

Luongo, Orlando and Marco Muccino (Oct. 2021). "A Roadmap to Gamma-Ray Bursts: new developments and applications to cosmology". en. In: *Galaxies* 9.4, p. 77.

Maidstone, Robert et al. (2017). "On optimal multiple changepoint algorithms for large data". en. In: *Statistics and Computing* 27.2, pp. 519–533.

Mardia, Kanti V (Sept. 2024). "Fisher's pioneering work on discriminant analysis and its impact on Artificial Intelligence". en. In: *J. Multivar. Anal.* 203.105341, p. 105341.

McLean, Kassandra et al. (2004). "Setting the triggering threshold on Swift". In: *AIP Conference Proceedings* 727.1, pp. 667–670. DOI: 10.1063/1.1810931. arXiv: astro-ph/0408512.

Meegan, Charles, Giselher Lichti, et al. (2009). "The Fermi gamma-ray burst monitor". In: *The Astrophysical Journal* 702.1, p. 791.

Mei, Yajun (Nov. 2008). "Is Average Run Length to False Alarm Always an Informative Criterion?" In: *Seq. Anal.* 27.4, pp. 354–376.

Mei, Yajun (2010). "Efficient scalable schemes for monitoring a large number of data streams". In: *Biometrika* 97.2, pp. 419–433.

Merzbacher, M and D Patterson (2003). "Measuring end-user availability on the Web: practical experience". In: *Proceedings International Conference on Dependable Systems and Networks*. IEEE Comput. Soc.

Mika, S et al. (1999). "Fisher discriminant analysis with kernels". In: *Neural Networks for Signal Processing IX: Proceedings of the 1999 IEEE Signal Processing Society Workshop (Cat. No.98TH8468)*. IEEE, pp. 41–48.

Miller, M Coleman (2017). "A golden binary". In: *Nature* 551.7678, pp. 36–37.

Muthukrishna, Daniel et al. (Sept. 2022). "Real-time detection of anomalies in large-scale transient surveys". en. In: *Mon. Not. R. Astron. Soc.* 517.1, pp. 393–419.

al-Nabhani, Khalid, Faisal Khan, and Ming Yang (Jan. 2016). "Technologically Enhanced Naturally Occurring Radioactive Materials in oil and gas production: A silent killer". en. In: *Process Saf. Environ. Prot.* 99, pp. 237–247.

Nizan, Ori and Ayellet Tal (2024). "k-NNN: Nearest Neighbors of Neighbors for Anomaly Detection". In: *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1005–1014.

NuSec (2022). *The SIGMA Data Challenge*. URL: https://www.nusec.uk/data-challenge (visited on 09/24/2024).

Omohundro, Stephen M (1989). *Five balltree construction algorithms*. International Computer Science Institute Berkeley.

Paciesas, William S, Charles A Meegan, et al. (1999). "The fourth BATSE gamma-ray burst catalog (revised)". In: *The Astrophysical Journal Supplement Series* 122.2, p. 465.

Paciesas, William S. et al. (2012). "The Fermi GBM gamma-ray burst catalog: the first two years". In: *The Astrophysical Journal Supplement Series* 199, p. 18. DOI: 10.1088/0067-0049/199/1/18. arXiv: 1201.3099 [astro-ph.HE].

Page, E S (1954). "Continuous Inspection Schemes". In: *Biometrika* 41.1/2, pp. 100–115.

— (1955). "A test for a change in a parameter occurring at an unknown point". In: *Biometrika* 42.3-4, pp. 523–527.

Petcharat, Kanita, Saowanit Sukparungsee, and Yupaporn Areepong (Jan. 2015). "Exact solution of the average run length for the cumulative sum chart for a moving average process of order q". In: *ScienceAsia* 41, pp. 141–147. DOI: 10.2306/scienceasia1513-1874.2015.41.141.

Pishchagina, L, G Romano, P Fearnhead, et al. (2023). "Online Multivariate Changepoint Detection: Leveraging Links With Computational Geometry". In: *arXiv preprint arXiv:2311.01174*.

Romano, Gaetano, Idris A Eckley, Paul Fearnhead, and Guillem Rigaill (2023). "Fast online changepoint detection via functional pruning CUSUM statistics". In: *J. Mach. Learn. Res.* 24.81, pp. 1–36.

Romano, Gaetano, Idris A. Eckley, and Paul Fearnhead (2024). "A Log-Linear Nonparametric Online Changepoint Detection Algorithm Based on Functional Pruning". In: *IEEE Transactions on Signal Processing* 72, pp. 594–606. DOI: 10.1109/TSP.2023.3343550.

Romano, Gaetano, Guillem Rigaill, et al. (2022). "Detecting abrupt changes in the presence of local fluctuations and autocorrelated noise". In: *Journal of the American Statistical Association* 117, pp. 2147–2162.

Rosner, Bernard (May 1983). "Percentage Points for a Generalized ESD Many-Outlier Procedure". In: *Technometrics* 25.2, pp. 165–172.

Ross, Gordon J. and Niall M. Adams (2012). "Two Nonparametric Control Charts for Detecting Arbitrary Distribution Changes". In: *Journal of Quality Technology* 44.2, pp. 102–116. DOI: 10.1080/00224065.2012.11917887.

Ross, Gordon J., Dimitris K. Tasoulis, and Niall M. Adams (2011). "Nonparametric Monitoring of Data Streams for Changes in Location and Scale". In: *Technometrics* 53.4, pp. 379–389. DOI: 10.1198/TECH.2011.10069.

Roth, Volker and V Steinhage (Nov. 1999). "Nonlinear discriminant analysis using kernel functions". In: *Adv. Neural Inf. Process. Syst.*, pp. 568–574.

Rousseeuw, Peter J and Mia Hubert (Mar. 2018). "Anomaly detection by robust statistics". In: *WIREs Data Mining Knowl Discov* 8.2, p. 157.

Russell-Pavier, Frederick S et al. (Jan. 2023). "A highly scalable and autonomous spectroscopic radiation mapping system with resilient IoT detector units for dosimetry, safety and security". en. In: *J. Radiol. Prot.* 43.1, p. 011503.

Ryan, Caitríona M, Andrew Parnell, and Catherine Mahoney (Nov. 2019). "Real-Time Anomaly Detection for Advanced Manufacturing: Improving on Twitter's State of the Art". In: arXiv: 1911.05376 [eess.SP].

Ryan, Thomas A (1959). "Multiple comparison in psychological research". en. In: *Psychol. Bull.* 56.1, pp. 26–47.

Sakamoto, Yusuke 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.

Schmidl, Sebastian, Phillip Wenig, and Thorsten Papenbrock (May 2022). "Anomaly detection in time series". en. In: *Proceedings VLDB Endowment* 15.9, pp. 1779–1797.

Šidák, Zbyněk (June 1967). "Rectangular confidence regions for the means of multivariate normal distributions". en. In: *J. Am. Stat. Assoc.* 62.318, pp. 626–633.

Soenen, Jonas et al. (2021). "The effect of hyperparameter tuning on the comparative evaluation of unsupervised anomaly detection methods". In.

Stehman, Stephen V (Oct. 1997). "Selecting and interpreting measures of thematic classification accuracy". en. In: *Remote Sens. Environ.* 62.1, pp. 77–89.

Tharwat, Alaa (Jan. 2021). "Classification assessment methods". en. In: *Appl. Comput. Inform.* 17.1, pp. 168–192.

Tickle, Sam O, IA Eckley, and Paul Fearnhead (2021). "A computationally efficient, high-dimensional multiple changepoint procedure with application to global terrorism incidence". In: *Journal of the Royal Statistical Society: Series A (Statistics in Society)*.

Tveten, Martin, Idris A Eckley, and Paul Fearnhead (June 2022). "Scalable change-point and anomaly detection in cross-correlated data with an application to condition monitoring". In: *Ann. Appl. Stat.* 16.2.

Van Rijsbergen, C J (1979). *Information Retrieval*. en. Butterworths.

Varghese, Blesson et al. (2016). "Challenges and Opportunities in Edge Computing". In: *2016 IEEE International Conference on Smart Cloud (SmartCloud)*, pp. 20–26. DOI: 10.1109/SmartCloud.2016.18.

Von Kienlin, Andreas, Charles A Meegan, William S Paciesas, PN Bhat, et al. (2020). "The fourth Fermi-GBM gamma-ray burst catalog: a decade of data". In: *The Astrophysical Journal* 893.1, p. 46.

Von Kienlin, Andreas, Charles A Meegan, William S Paciesas, PN Bhat, et al. (2014). "The second Fermi GBM gamma-ray burst catalog: the first four years". In: *The Astrophysical Journal Supplement Series* 211.1, p. 13.

Vries, Alex de (Oct. 2023). "The growing energy footprint of artificial intelligence". en. In: *Joule* 7.10, pp. 2191–2194.

Wang, Chen and Hyong Kim (Feb. 2019). "Touchdown on the Cloud: The impact of the Super Bowl on Cloud". In: *arXiv [cs.NI]*.

Ward, Kes, Giuseppe Dilillo, et al. (2023). "Poisson-FOCuS: An Efficient Online Method for Detecting Count Bursts with Application to Gamma Ray Burst Detection". In: *Journal of the American Statistical Association* 0.0, pp. 1–13. DOI: 10.1080/01621459.2023.2235059.

Ward, Kes, Gaetano Romano, et al. (Mar. 2024). "A constant-per-iteration likelihood ratio test for online changepoint detection for exponential family models". In: *Stat. Comput.* 34.3, p. 99.

Wilks, S S (1938). "The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses". In: *The Annals of Mathematical Statistics* 9.1, pp. 60–62.

Wu, Renjie and Eamonn J Keogh (Mar. 2023). "Current Time Series Anomaly Detection Benchmarks are Flawed and are Creating the Illusion of Progress". In: *IEEE Trans. Knowl. Data Eng.* 35.3, pp. 2421–2429.

Xu, Haowen et al. (Apr. 2018). "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications". In: *Proceedings of the 2018 World Wide Web Conference*. WWW '18. Lyon, France: International World Wide Web Conferences Steering Committee, pp. 187–196.

Yang, Ziyang, Idris A Eckley, and Paul Fearnhead (June 2024). "A communication-efficient, online changepoint detection method for monitoring distributed sensor networks". en. In: *Stat. Comput.* 34.3.

Yu, Yi et al. (Oct. 2023). “A note on online change point detection”. en. In: *Seq. Anal.* 42.4, pp. 438–471.

Zurawski, R (2004a). “The Fundamentals of the Quality of Service”. In: *The Industrial Information Technology Handbook*. CRC Press.

Zurawski, Richard (2004b). “RTP RTCP and RTSP 8212 Internet Protocols for Real Time Multimedia Communication”. In: *The Industrial Information Technology Handbook*. CRC Press.

Appendix A

A.1 Derivations of the LR statistic for Chapter 3

A.1.1 Window method

Let $\ell(x_{t+1:t+h};\mu)$ denote the log-likelihood for the data $x_{t+1:t+h}$ under our Poisson model with rate $\mu\lambda$. Then the standard (log) likelihood ratio statistic is

$$LR = 2\left\{\max_{\mu>1}\ell(x_{t+1:t+h};\mu) - \ell(x_{t+1:t+h};1)\right\}.$$

This is 0 if $\bar{x}_{t+1:t+h} \leq \lambda$, otherwise

$$LR = 2h\lambda\left\{\frac{\bar{x}_{t+1:t+h}}{\lambda}\log\left(\frac{\bar{x}_{t+1:t+h}}{\lambda}\right) - \left(\frac{\bar{x}_{t+1:t+h}}{\lambda} - 1\right)\right\}.$$

Proof. On an interval $x_{t+1:t+h}$, we have expected count $h\lambda$ and actual count $h\bar{x}_{t+1:t+h}$. We utilise the Poisson likelihood

$$L(\lambda; x_{t+1:t+h}) = \frac{e^{-h\lambda}(h\lambda)^{h\bar{x}_{t+1:t+h}}}{(h\bar{x}_{t+1:t+h})!},$$

and log-likelihood

$$\ell(\lambda; x_{t+1:t+h}) = -h\lambda + h\bar{x}_{t+1:t+h}\log(h\lambda) + c.$$

Our likelihood ratio statistic then becomes

$$LR = -2\left\{\ell(\lambda; x_{t+1:t+h}) - \ell(\bar{x}_{t+1:t+h}; x_{t+1:t+h})\right\}$$

$$= -2\left\{-h\lambda + h\bar{x}_{t+1:t+h}\log(h\lambda) - (-h\bar{x}_{t+1:t+h} + h\bar{x}_{t+1:t+h}\log(h\bar{x}_{t+1:t+h}))\right\}$$

$$= 2h\lambda\left\{\frac{\bar{x}_{t+1:t+h}}{\lambda}\log\left(\frac{\bar{x}_{t+1:t+h}}{\lambda}\right) - \left(\frac{\bar{x}_{t+1:t+h}}{\lambda} - 1\right)\right\}.$$

□

A.1.2 Page-CUSUM method

We have our hypotheses for the signal at time T:

- $\mathbf{H}_0$: There have been no anomalies, i.e. $X_1, ..., X_T \sim \text{Poisson}(\lambda)$.
- $\mathbf{H}_1$: There has been one anomaly, beginning at some unknown time τ, with known intensity multiplier $\mu > 1$, i.e. $X_1, ..., X_{\tau-1} \sim \text{Poisson}(\lambda)$ and $X_\tau, ..., X_T \sim \text{Poisson}(\mu\lambda)$.

Our LR statistic for this test is 0 if $\bar{x}_{\tau:T} \leq \lambda\frac{\mu-1}{\log(\mu)}$ for all τ, otherwise

$$LR = \max_{1\leq\tau\leq T}\left[2(T-\tau+1)\lambda\left\{\frac{\bar{x}_{\tau:T}}{\lambda}\log(\mu) - (\mu-1)\right\}\right].$$

Proof. Our Poisson likelihood and log-likelihood is as follows:

$$L(\lambda; x_{1:T}) = \frac{e^{-T\lambda}(T\lambda)^{\sum_{t=1}^T x_t}}{(\sum_{t=1}^T x_t)!},$$

$$l(\lambda; x_{1:T}) = -T\lambda + \sum_{t=1}^T x_t\log(\lambda) + c.$$

Under the null hypothesis of no anomaly, and the alternative of one anomaly at τ, we have as our log-likelihoods the following:

$$l(\mathbf{H}_0; x_{1:T}) = \sum_{t=1}^{T} [x_t \log(\lambda) - \lambda] + c$$

$$l(\mathbf{H}_1; x_{1:T}) = \max_{1 \leq \tau \leq T} \left(\sum_{t=1}^{\tau-1} [x_t \log(\lambda) - \lambda] + \sum_{t=\tau}^{T} [x_t \log(\mu\lambda) - \mu\lambda] \right) + c$$

Here, the maximum is because we have no idea where our start point τ actually is, so we look at them all and pick the one with largest likelihood. This gives our log-likelihood ratio statistic as

$$LR = 2 \max_{1 \leq \tau \leq T} \sum_{t=\tau}^{T} (x_t \log(\mu) - \lambda(\mu - 1))$$

$$= \max_{1 \leq \tau \leq T} \left[2(T - \tau + 1)\lambda \left\{ \frac{\bar{x}_{\tau:T}}{\lambda} \log(\mu) - (\mu - 1) \right\} \right].$$

□

A.1.3 Exponentially distributed data

The Poisson-FOCuS algorithm still works in the Exponential case, with the only difference being how we update the coefficients of the curves.

$$a_\tau^{(T+1)} = a_\tau^{(T)} + 1, \quad b_\tau^{(T+1)} = b_\tau^{(T)} + \lambda_{T+1} U_{T+1},$$

where λ_T is the estimate of the background rate at the time of the Tth photon arrival.

Proof. Making the assumption that we can consider the background rate constant between successive photon arrivals, our hypotheses for an individual logarithmic curve $C_\tau^{(T)}$ are as follows:

- $\mathbf{H}_0$: $U_\tau, \ldots, U_T$ has $U_t \sim \text{Exp}(\lambda_t)$.
- $\mathbf{H}_1$: $U_\tau, \ldots, U_T$ has $U_t \sim \text{Exp}(\mu\lambda_t)$, for some $\mu > 1$.

The exponential likelihood and log-likelihood are as follows:

$$L(\lambda_{\tau:T}; u_{\tau:T}) = \prod_{t=\tau}^{T} (\lambda_t) e^{-\sum_{t=\tau}^{T} \lambda_t u_t},$$

$$l(\lambda_{\tau:T}; u_{\tau:T}) = \sum_{t=\tau}^{T} \log(\lambda_t) - \sum_{t=\tau}^{T} \lambda_t u_t.$$

This gives our log-likelihood ratio for the curve as

$$C_{\tau}^{(T)} := l(\mu\lambda_{\tau:T}; u_{\tau:T}) - l(\lambda_{\tau:T}; u_{\tau:T}) = \sum_{t=\tau}^{T} [\log(\mu) - \lambda_t u_t(\mu - 1)]$$

This gives the update coefficients for curves as stated.

□

A.2 Proofs for Chapter 3

A.2.1 Equivalences between Page-CUSUM and window methods

Proposition A.2.1. *For some choice of μ against a background rate of λ, let S_T be significant at the k-sigma level. Then there exists some interval $[\tau, T]$ with associated likelihood ratio statistic that is significant at the k-sigma level.*

Proof. Consider the last time τ where Page's statistic last became non-zero. On $[\tau, T]$ the likelihood ratio with our choice of $\mu > 1$ exceeds a k-sigma threshold, therefore the maximised likelihood ratio over an unconstrained $\mu > 1$ (which occurs at $\mu = \frac{\bar{x}_{\tau:T}}{\lambda}$) also exceeds a k-sigma threshold. □

Proposition A.2.2. *For any k, λ and h there exists a μ and corresponding test statistic, S_T, that relates directly to a window test of length h, and background rate λ as follows: if, for any t, the data $x_{t+1:t+h}$ is significant at the k-sigma level then S_{t+h} will also be significant at the k-sigma level.*

Proof. We choose the value of μ solving the equation

$$2h\lambda \left[\mu \log(\mu) - (\mu - 1)\right] = k^2,$$

i.e. the ideal intensity choice for the expected count $h\lambda$ used in this likelihood ratio test. Since $x_{t+1:t+h}$ is significant at the k-sigma level, we have that

$$2h\lambda\left[\frac{\bar{x}_{t+1:t+h}}{\lambda}\log\left(\frac{\bar{x}_{t+1:t+h}}{\lambda}\right)-\left(\frac{\bar{x}_{t+1:t+h}}{\lambda}-1\right)\right]\geq k^2.$$

As the function $f(x)=x\log x-(x-1)$ is an increasing function, this shows that $\bar{x}_{t+1:t+h}/\lambda\geq\mu$.

We then have that

$$\begin{aligned}
S_{t+h}(\mu) &= \left[\max_{1\leq\tau\leq t+h}\sum_{s=\tau}^{t+h}(x_s\log(\mu)-\lambda(\mu-1))\right]^+ \\
&\geq \sum_{s=t+1}^{t+h}(x_s\log(\mu)-\lambda(\mu-1)) \\
&= h\lambda\left[\frac{\bar{x}_{t+1:t+h}}{\lambda}\log(\mu)-(\mu-1)\right] \\
&\geq h\lambda\left[\mu\log\left(\mu\right)-(\mu-1)\right] \\
&= \frac{k^2}{2}.
\end{aligned}$$

Therefore $S_{t+h}(\mu)$ is significant at a k-sigma level.

□

A.2.2 Conditions for pruning

Proposition A.2.3. *Let $C_{\tau_i}^{(T)}$ and $C_{\tau_j}^{(T)}$ be curves that are positive somewhere on $\mu\in[1,\infty)$, where $\tau_i<\tau_j$ and $C_{\tau_i}^{(\tau_j-1)}$ is also positive somewhere on $\mu\in[1,\infty)$.*

Then $C_{\tau_i}^{(T)}$ dominates $C_{\tau_j}^{(T)}$ if and only if $a_{\tau_j}^{(T)}/b_{\tau_j}^{(T)}\leq a_{\tau_i}^{(\tau_j-1)}/b_{\tau_i}^{(\tau_j-1)}$ or equivalently $a_{\tau_j}^{(T)}/b_{\tau_j}^{(T)}\leq a_{\tau_i}^{(T)}/b_{\tau_i}^{(T)}$. Additionally, it cannot be the case that $C_{\tau_j}^{(T)}$ dominates $C_{\tau_i}^{(T)}$.

Proof. Let μ_{ij} be the non-unit intersection point of $C_{\tau_i}^{(T)}$ and $C_{\tau_j}^{(T)}$, i.e. the root of $C_{\tau_i}^{(\tau_j-1)}$. Then by rearrangement we have that

$$a_{\tau_i}^{(\tau_j-1)}\log(\mu_{ij})-b_{\tau_i}^{(\tau_j-1)}(\mu_{ij}-1)=0,$$

$$\frac{a_{\tau_i}^{(\tau_j-1)}}{b_{\tau_i}^{(\tau_j-1)}} = \frac{\mu_{ij}-1}{\log(\mu_{ij})}.$$

Because $C_{\tau_i}^{(\tau_j-1)}$ is non-negative on $\mu \in [1, \mu_{ij})$, we cannot have $C_{\tau_j}^{(T)}$ dominating $C_{\tau_i}^{(T)}$. For $C_{\tau_i}^{(T)}$ to dominate $C_{\tau_j}^{(T)}$, we must have that $C_{\tau_j}^{(T)} \leq 0$ on $\mu \in [\mu_{ij}, \infty)$, i.e. $C_{\tau_j}^{(T)}(\mu_{ij}) \leq 0$. Rearranging, we have

$$a_{\tau_j}^{(T)} \log(\mu_{ij}) - b_{\tau_j}^{(T)}(\mu_{ij}-1) \leq 0,$$

$$\frac{a_{\tau_j}^{(T)}}{b_{\tau_j}^{(T)}} \leq \frac{\mu_{ij}-1}{\log(\mu_{ij})}.$$

Putting these together gives us the condition $a_{\tau_j}^{(T)}/b_{\tau_j}^{(T)} \leq a_{\tau_i}^{(\tau_j-1)}/b_{\tau_i}^{(\tau_j-1)}$. For the other form, we can rearrange the inequality:

$$a_{\tau_j}^{(T)} b_{\tau_i}^{(\tau_j-1)} \leq a_{\tau_i}^{(\tau_j-1)} b_{\tau_j}^{(T)},$$

$$a_{\tau_j}^{(T)} b_{\tau_i}^{(\tau_j-1)} + a_{\tau_j}^{(T)} b_{\tau_j}^{(T)} \leq a_{\tau_i}^{(\tau_j-1)} b_{\tau_j}^{(T)} + a_{\tau_j}^{(T)} b_{\tau_j}^{(T)},$$

$$a_{\tau_j}^{(T)} b_{\tau_i}^{(T)} \leq a_{\tau_i}^{(T)} b_{\tau_j}^{(T)}.$$

□

A.2.3 Logarithmic curve bound

Proposition A.2.4. *The expected number of curves kept by Poisson-FOCuS without μ_{min} at each timestep T is $\in \left[\frac{\log(T)}{2}, \frac{\log(T)+1}{2}\right]$.*

Proof. Recalling that a logarithmic curve $C_\tau^{(T)}(\mu)$ is defined as

$$C_\tau^{(T)}(\mu) := \sum_{t=\tau}^{T} [X_t \log(\mu) - \lambda(\mu-1)],$$

we define the set of candidate start points $\mathfrak{I}_T$ at time T to be the set of all τ directly contributing to $S_T(\mu)$, i.e.

$$\mathfrak{I}_T := \{\tau : \exists \mu, \forall \tau^{'} \neq \tau, [C_{\tau}^{(T)}(\mu)]^{+} > [C_{\tau^{'}}^{(T)}(\mu)]^{+}\}.$$

The number of curves kept by Poisson-FOCuS at time T is, barring computational implementations that occasionally keep extra curves to avoid repeated pruning checks, exactly $|\mathfrak{I}_T|$.

Lemma A.2.5. *Suppose $\tau^{'} \in \mathfrak{I}_T$. This is equivalent to the following two conditions:*

- *for any $\tau^{'} < \tau^{''} \leq T$, we have that*

$$\lambda < \bar{X}_{\tau^{'}:\tau^{''}}.$$

- *for any $1 \leq \tau < \tau^{'} < \tau^{''} \leq T$, we have that*

$$\bar{X}_{\tau,\tau^{'}-1} < \bar{X}_{\tau^{'},\tau^{''}}.$$

Proof. Suppose $\exists \tau, \tau^{''}$ such that we have

$$\bar{X}_{\tau:\tau^{'}-1} \geq \bar{X}_{\tau^{'}:\tau^{''}}.$$

Consider the two curves $C_{\tau}^{(\tau^{''})}(\mu)$ and $C_{\tau^{'}}^{(\tau^{''})}(\mu)$:

$$
\begin{aligned}
C_{\tau}^{(\tau^{''})}(\mu) &= \sum_{t=\tau}^{\tau^{''}} [X_t \log(\mu) - \lambda(\mu - 1)] \\
&= [(\tau^{'} - \tau)\bar{X}_{\tau:\tau^{'}-1} + (\tau^{''} - \tau^{'} + 1)\bar{X}_{\tau^{'}:\tau^{''}}] \log(\mu) - [\tau^{''} - \tau + 1]\lambda(\mu - 1) \\
&\geq [(\tau^{''} - \tau + 1)\bar{X}_{\tau:\tau^{''}}] \log(\mu) - [\tau^{''} - \tau + 1]\lambda(\mu - 1) \\
&= \frac{\tau^{''} - \tau + 1}{\tau^{''} - \tau^{'} + 1} C_{\tau^{'}}^{(\tau^{''})}(\mu) \\
&\geq C_{\tau^{'}}^{(\tau^{''})}(\mu).
\end{aligned}
$$

So $\tau^{'} \notin \mathfrak{I}_T$.

What this is saying is that if $\bar{X}_{\tau,\tau^{'}-1} < \bar{X}_{\tau^{'},\tau^{''}}$, then the interval $[\tau, \tau^{''}]$ has both greater intensity and greater duration than the interval $[\tau^{'}, \tau^{''}]$, so $\tau^{'}$ cannot be a candidate start point.

To prove the reverse, we note that the non-unit point of intersection between $C_{\tau^{'}}^{(\tau^{''})}(\mu)$ and 0 is a monotone increasing function of $\bar{X}_{\tau^{'}:\tau^{''}}$. Therefore, if for all $\tau < \tau^{'} < \tau^{''} \leq T$,

$$
\bar{X}_{\tau:\tau^{''}} < \bar{X}_{\tau^{'}:\tau^{''}}, \quad \lambda < \bar{X}_{\tau^{'}:\tau^{''}},
$$

we must have that $\exists \mu > 1$ such that $[C_{\tau^{'}}^{(\tau^{''})}(\mu)]^{+} > [C_{\tau}^{(\tau^{''})}(\mu)]^{+}$. This gives $\tau^{'} \in \mathfrak{I}_T$.

□

Lemma A.2.6. *Define the sequence $Z_0 := 0$, $Z_T := \sum_{t=1}^{T} X_t$.*

If τ in $\mathfrak{I}_T$, then:

- *$\tau - 1$ is an extreme point of the largest convex minorant of the sequence $\{Z_t - t\lambda : t \leq T\}$.*
- *$\forall T \geq t > \tau - 1$, we additionally have that $Z_t - t\lambda > Z_{\tau-1} - (\tau - 1)\lambda$, i.e. $\tau - 1$ is on the "right-hand side" of the convex minorant.*

Proof. Let τ in $\mathfrak{I}_T$.

As above, we have that for any $1 \leq \tau < \tau^{'} < \tau^{''} \leq T$:

$$\bar{X}_{\tau,\tau^{'}-1} < \bar{X}_{\tau^{'},\tau^{''}}.$$

This can be equivalently written as

$$\frac{Z_{\tau^{'}-1} - Z_{\tau}}{(\tau^{'}-1)-\tau} < \frac{Z_{\tau^{''}} - Z_{\tau^{'}-1}}{\tau^{''}-(\tau^{'}-1)},$$

which shows that $\tau - 1$ is in the largest convex minorant of Z_t, and therefore of $Z_t - t\lambda$.

To show we are on the right-hand side of this convex minorant, we assume hoping for a contradiction that $\exists \tau^{'} \geq \tau$ such that $Z_{\tau^{'}} - \tau^{'}\lambda < Z_{\tau-1} - (\tau-1)\lambda$. We then have that

$$C_{\tau}^{(\tau^{'})}(\mu) = \sum_{t=\tau}^{\tau^{'}} [X_t \log(\mu) - \lambda(\mu - 1)]$$

$$= [Z_{\tau^{'}} - Z_{\tau-1}] \log(\mu) - [\tau^{'} - (\tau-1)]\lambda(\mu-1)$$

$$< [\tau^{'} - (\tau-1)]\lambda[\log(\mu) - (\mu-1)]$$

$$< 0.$$

So for $\tau^{'} = \tau$ we have that $X_\tau < \lambda$, and for $\tau^{'} > \tau$ we have that $C_{\tau}^{(T)}(\mu) < C_{\tau^{'}}^{(T)}(\mu)$ pointwise. Either way, $\tau \notin \mathfrak{I}_T$.

To prove the reverse, note that the argument for being on the convex minorant is entirely reversible, and that $Z_t - t\lambda < Z_{\tau-1} - (\tau-1)\lambda$ is equivalent to $\bar{X}_{\tau,t} < \lambda$.

□

Figure A.2.1 shows what this looks like. For a signal X_t that Poisson-FOCuS is run over, the random walk $Z_t - t\lambda$ is plotted. Values of $\tau - 1$ for each candidate anomaly

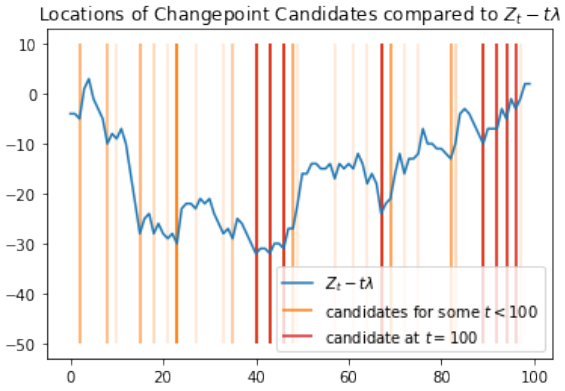


Figure A.2.1: Plots of $\tau - 1$ for each anomaly start point τ compared to the random walk $Z_t - t\lambda$.

start point τ are highlighted in orange, with the intensity of the highlight corresponding to how long they were kept, or highlighted in red if they were still kept by the time $T = 100$.

For each candidate point τ with $\tau - 1$ highlighted in red, we have that:

- Convex minorant: The gradient drawn through $Z_{\tau-1} - (\tau - 1)\lambda$ from any point before $\tau - 1$ must be less than the gradient drawn through $Z_{\tau-1} - (\tau - 1)\lambda$ by any point after $\tau - 1$.

- Right side: It is possible to draw a straight horizontal line from $Z_{\tau-1} - (\tau - 1)\lambda$ to the right side of the graph without crossing any other $Z_t - t\lambda$.

This is approximately half the points in the convex minorant of $Z_t - t\lambda$ (the other half being the left-hand side, with one point - the minimum - being in both).

Theorem A.2.7. *Let $X_t, 1 \leq t \leq T$ be independent identically distributed continuous random variables, and let $S_t := \sum_{s=1}^{t} X_s$ be the corresponding random walk. Then the number of points $H(T)$ on the convex minorant of the sequence $(0, S_1, S_2, ..., S_T)$ (not including endpoints) has the distribution*

$$H(T) \sim \sum_{t=1}^{T-1} Y_t,$$

$$Y_t \sim Bernoulli\left(\frac{1}{t+1}\right),$$

where the Y_t are independent of each other and the distribution of the X_t.

Proof. See Andersen 1954 □

Under the null hypothesis, we have $X_t \sim \text{Poisson}(\lambda)$ independent and identically distributed.

By Lemma A.2.6, the $|\mathfrak{I}_T|$ is the number of points on the right-hand side of the convex minorant of $\{Z_t - t\lambda : t \leq T\}$.

By Therorem A.2.7, the expected number of points on the convex minorant of $\{Z_t - t\lambda : t \leq T\}$ not including endpoints is

$$\mathbb{E}[\text{points on convex minorant}] = \sum_{t=2}^{T} \frac{1}{t}.$$

By symmetry, the expected number of points on the right-hand side of the convex minorant (including the minimum point, which is on both sides), is

$$\mathbb{E}[|\mathfrak{I}_T|] = \frac{1}{2}\sum_{t=1}^{T} \frac{1}{t}.$$

Because $\tau \in \mathfrak{I}_T$ is related to $\tau - 1$ (rather than τ) being on the right-hand side of the convex minorant of $\{Z_t - t\lambda : t \leq T\}$, it is impossible to be on the rightmost endpoint. However, it could be that $\tau - 1 = 0$ could be both the leftmost endpoint and on the right-hand side of the convex minorant, which would give an additional curve beyond those given by Theorem A.2.7. This would require $\min\{Z_t - t\lambda : t \leq T\} \geq 0$, which has a probability that $\to 0$ as $T \to \infty$, and can therefore be discounted for large values of T as it falls within the harmonic upper bound given below.

We have by standard results for harmonic sums that

$$\sum_{t=1}^{T} \frac{1}{t} \in [\log(T), \log(T) + 1],$$

Giving us that

$$\mathbb{E}[|\mathfrak{I}_T|] \in \left[\frac{\log(T)}{2}, \frac{\log(T)+1}{2}\right].$$

□

A.2.4 Bounded number of curves in the $\mu_{\min} > 1$ case

Proposition A.2.8. *The expected number of curves kept by Poisson-FOCuS using some $\mu_{min} > 1$ at each timestep is bounded.*

Proof. Let $\lambda > 0$, $\mu_{\min} > 1$ be fixed, and $X_T \sim \text{Poisson}(\lambda)$. Define $S_0 = 0$, and for each $T \in \mathbb{N}$ recursively define

$$S_{T+1} = S_T + X_{T+1}\log(\mu_{\min}) - \lambda(\mu_{\min} - 1).$$

This gives essentially Page's statistic without resetting negative values to zero. We further define:

$$H(X) := \inf_{T>1}\{T : S_T \leq 0\}$$

i.e. $H(X)$ is the time elapsed between resets to zero of Page's statistic.

In order to prove positive recurrence of Page's statistic, we now show that $\mathbb{E}[H(X)]$ is finite.

We have that

$$\mathbb{E}[S_T] = \lambda T[\log(\mu_{\min}) - (\mu_{\min} - 1)] < 0,$$

$$\text{Var}[S_T] = \lambda T(\log(\mu_{\min}))^2 < \infty.$$

This gives that $H(X) < \infty$ almost surely.

By the central limit theorem, we have that as $T \to \infty$,

$$\frac{S_T - \lambda T[\log(\mu_{\min}) - (\mu_{\min} - 1)]}{\sqrt{\lambda T}\log(\mu_{\min})} \approx N(0,1).$$

Using this approximation we can then calculate

$$S_T \approx N(0,1)\sqrt{\lambda T}\log(\mu_{\min}) + \lambda T[\log(\mu_{\min}) - (\mu_{\min} - 1)].$$

$$\mathbb{P}(S_T < 0) \approx \Phi\left(\sqrt{\lambda T}\left[1 - \frac{(\mu_{\min} - 1)}{\log(\mu_{\min})}\right]\right)$$

This gives us the following bound:

$$\begin{aligned}
\mathbb{E}[H(X)] &= \sum_{T=1}^{\infty} \mathbb{P}[H(X) \leq T] \\
&\leq \sum_{T=1}^{\infty} \mathbb{P}(S_T < 0) \\
&\approx \sum_{T=1}^{\infty} \Phi\left(\sqrt{\lambda T}\left[1 - \frac{(\mu_{\min} - 1)}{\log(\mu_{\min})}\right]\right) \\
&< \infty.
\end{aligned}$$

The last step is because the Gaussian distribution has tails that drop as the square of an exponential, and geometric series have finite sum.

Therefore, the expected number of curves in the FOCuS algorithm running using a $\mu_{\min}$ is bounded, because all curves in the algorithm are removed each time Page's statistic using $\mu_{\min}$ resets to 0, and the expected time between resets is finite.

□

A.3 Plots for Chapter 3

A.3.1 Detectability regions

Here we provide the derivations of the detectability regions. For ease of reference, we reproduce these regions in Figure A.3.2.

Assume we are running a window of length h over a signal containing a burst $x_{t+1:t+h^*}$ of length h^*. Our background rate λ is assumed fixed. We want to figure out what is the smallest intensity μ^* we are able to detect, assuming that μ is the faintest intensity at which a burst of length h is detectable.

Bursts of duration $h^* > h$ will only be detected at the k-sigma level if some subinterval of size h is detected at the k-sigma level. No additional benefit can be provided by the presence of the part of the burst currently outside the window, so $\mu^* = \mu$. Therefore the green line on Figure A.3.2 has been drawn as a straight vertical.

Bursts of a duration $h^* < h$ can be found if they have a higher μ^*. Splitting the window h into anomalous and non-anomalous parts, we have that

$$\mu h\lambda = \mu^* h\lambda^* + \lambda(h - h^*).$$

This rearranges to

$$(\mu - 1)h = (\mu^* - 1)h^*,$$

which gives the other green line shown in Figure A.3.2.

Assume we are using Page's method with parameter μ over a signal containing a burst of intensity μ^*. Our background rate λ is assumed fixed. We want to figure out what is the shortest duration h^* required to detect the burst at a k-sigma threshold. Using our likelihood ratio, we have that:

$$h^*\lambda\left[\mu^* \log(\mu) - (\mu - 1)\right] = \frac{k^2}{2},$$

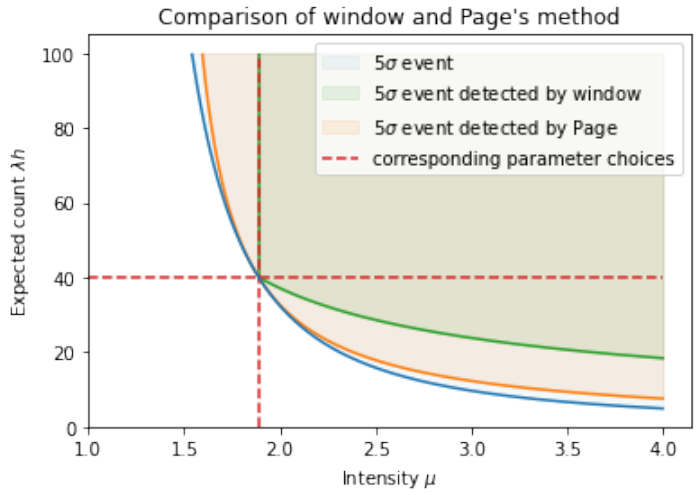


Figure A.3.2: Detectability of Page-CUSUM and Window methods.

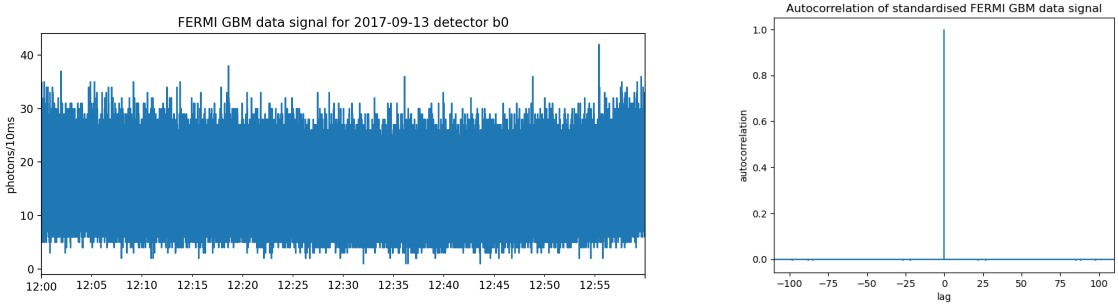


Figure A.3.3: Left: one hour's worth of FERMI data binned into 10ms intervals. Right: autocorrelation plot from the data put through a variance-stabilising transformation (square root) and then rolling mean of window size 500 subtracted off to account for changes in background rate. Negligible autocorrelation is present.

$$\mu^* = \frac{1}{\log(\mu)} \left[\frac{k^2}{2h^*\lambda} + (\mu - 1) \right].$$

This gives the orange line in Figure A.3.2.

A.3.2 Autocorrelation

Figure A.3.3 shows one hour's worth of FERMI data. To detect if any autocorrelation is present after accounting for the varying background rate, we first use a square root transform to stabilise the variance of the Poisson distribution, and then subtract off an estimate of the background rate calculated as the rolling mean using a window of size

500 observations. The empirical autocorrelation of this transformed data is shown in the right-hand plot of figure A.3.3, and shows negligible autocorrelation at all (non-zero) lags.

A.4 Proofs for Chapter 5

A.4.1 Deriving the Exponential family likelihood ratio

For an exponential family model of the form

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

Our differences in likelihood are of the form

$$\ell(x_1{:}x_T|\theta_0, \theta_1, \tau_i) - \ell(x_1{:}x_T|\theta_0, \theta_1, \tau_j) = \\ [\alpha(\theta_1) - \alpha(\theta_0)] \sum_{t=\tau_i+1}^{\tau_j} \gamma(x_t) - [\beta(\theta_1) - \beta(\theta_0)](\tau_j - \tau_i).$$

Proof. We have that

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

Therefore, we have

$$\ell(x_{1:T}|\theta_0, \theta_1, \tau_i) - \ell(x_1{:}x_T|\theta_0, \theta_1, \tau_j) = \sum_{t=\tau_i+1}^{\tau_j} \{\log f(x_t|\theta_1) - \log f(x_t|\theta_0)\}.$$

Substituting in

$$\begin{aligned}\log f(x_t|\theta_1) - \log f(x_t|\theta_0) = &\quad [\alpha(\theta_1)\cdot\gamma(x_t) - \beta(\theta_1) + \delta(x_t)] - [\alpha(\theta_0)\cdot\gamma(x_t) - \beta(\theta_0) + \delta(x_t)] \\ = &\quad [\alpha(\theta_1) - \alpha(\theta_0)]\gamma(x_t) - [\beta(\theta_1) - \beta(\theta_0)]\end{aligned}$$

gives the required result. □

A.4.2 Ordering of roots determined by $\bar{\gamma}$ values

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 $t = \tau_i + 1, \ldots, \tau_j$, and define $\theta_1^\tau (\neq \theta_0)$ to be the root of

$$\ell(x_1{:}x_T|\theta_0, \theta_1^\tau, \tau) - \ell(x_1{:}x_T|\theta_0, \cdot, T) = 0.$$

Proposition A.4.1. *Suppose that for our choice of θ_0 the function*

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

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. We have that

$$[\alpha(\theta_1^\tau) - \alpha(\theta_0)]\sum_{t=\tau+1}^{T}\gamma(x_t) - [\beta(\theta_1^\tau) - \beta(\theta_0)](T - \tau) = 0.$$

Rearrange this to form

$$\frac{\beta(\theta_1^\tau) - \beta(\theta_0)}{\alpha(\theta_1^\tau) - \alpha(\theta_0)} = \bar{\gamma}_{\tau:T}.$$

By monotonicity, we have that θ_1^τ is an increasing function of $\bar{\gamma}_{\tau:T}$. For $\tau_i < \tau_j < T$ we also have that

$$\bar{\gamma}_{\tau_i:T} = \frac{T-\tau_j}{T-\tau_i}\bar{\gamma}_{\tau_j:T} + \frac{\tau_j-\tau_i}{T-\tau_i}\bar{\gamma}_{\tau_i:\tau_j},$$

so the sign of $\bar{\gamma}_{\tau_i:\tau_j} - \bar{\gamma}_{\tau_j:T}$ is the same as the sign of $\bar{\gamma}_{\tau_i:T} - \bar{\gamma}_{\tau_j:T}$ because $\bar{\gamma}_{\tau_i:T}$ is a convex combination of $\bar{\gamma}_{\tau_i:\tau_j}$ and $\bar{\gamma}_{\tau_j:T}$. Putting this together gives the result. □

A.4.3 Maxima checking bound

Define

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

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 A.4.2. *For any $\tau_1 < \tau_2 < ... < \tau_n < T$, we have*

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

Proof. Denote by $\hat{\theta}_0^{\tau_i}$ the argmax of $\sum_{t=1}^{\tau_i} \log f(x_t|\theta_0)$ for $\theta \in H_0$. (Note that in the pre-change mean known case, we always have $\hat{\theta}_0^{\tau_i} = \theta_0$.)

Now, consider the form of

$$m_{\tau_i,\tau_j} = \sum_{t=1}^{\tau_i} \log f(x_t|\hat{\theta}_0^{\tau_i}) + \max_{\theta_1} \sum_{t=\tau_i+1}^{\tau_j} \log f(x_t|\theta_1) - \sum_{t=1}^{\tau_j} \log f(x_t|\hat{\theta}_0^{\tau_j}).$$

Note the similarity of the first and third terms that will allow telescopic cancellations when summing the $m_{\tau_i,\tau_{i+1}}$. Setting $\tau_{n+1} := T$ for convenience, we have that for any $1 \leq k \leq n$,

$$\sum_{i=1}^{n-1} m_{\tau_i,\tau_{i+1}} + m_{\tau_n,T} = \left[\sum_{t=1}^{\tau_1} \log f(x_t|\hat{\theta}_0^{\tau_1}) + \sum_{i=1}^{k-1} \max_{\theta_1} \sum_{t=\tau_i+1}^{\tau_{i+1}} \log f(x_t|\theta_1)\right] + \left[\sum_{i=k}^{n} \max_{\theta_1} \sum_{t=\tau_i+1}^{\tau_{i+1}} \log f(x_t|\theta_1)\right] - \sum_{t=1}^{T} \log f(x_t|\hat{\theta}_0^T).$$

We can compare this against

$$m_{\tau_k,T} = \sum_{t=1}^{\tau_k} \log f(x_t|\hat{\theta}_0^{\tau_1}) + \max_{\theta_1} \sum_{t=\tau_k+1}^{T} \log f(x_t|\theta_1) - \sum_{t=1}^{T} \log f(x_t|\hat{\theta}_0^T),$$

noting that we have inequalities on the first two terms due to maximising the same likelihood over an expansion of the hypothesis set, and equality in the final term. This proves the result. □

The construction $\sum_{i=1}^{n-1} m_{\tau_i,\tau_{i+1}} + m_{\tau_n,T}$ is essentially fitting changepoints at every single one of the τ_i. This compares against the construction $\max_{i=1,...,n} m_{\tau_i,T}$, which fits only one changepoint at the most promising τ_i.

Where $\{\tau_1, ..., \tau_n\} \in \mathcal{I}_T$ and are therefore ordered in increasing/decreasing $\bar{\gamma}_{\tau_i:\tau_{i+1}}$ all representing up-changes/down-changes, it is the case that you don't gain much by fitting all of the τ_i as changepoints rather than just the best one. In the underlying data scenario of no change, the earlier $m_{\tau_i,\tau_{i+1}}$ will be very small, and it is $m_{\tau_n,T}$ that will contribute the most as it captures the fluctuations of recent events in the signal.