# Adaptive estimation with change detection for streaming data

A thesis presented for the degree of

Doctor of Philosophy of the University of London

and the

Diploma of Imperial College

by

**Dean Adam Bodenham**

Department of Mathematics

Imperial College

180 Queen’s Gate, London SW7 2AZ

NOVEMBER 12, 2014

# Copyright

To my parents

# Abstract

Data streams have become ubiquitous over the last two decades; potentially unending streams of continuously-arriving data occur in fields as diverse as medicine, finance, astronomy and computer networks. As the world changes, so the behaviour of these streams is expected to change. This thesis describes sequential methods for the timely detection of changes in data streams based on an adaptive forgetting factor framework. These change detection methods are first formulated in terms of detecting a change in the mean of a univariate stream, but this is later extended to the multivariate setting, and to detecting a change in the variance.

The key issues driving the research in this thesis are that streaming data change detectors must operate sequentially, using a fixed amount of memory and, after encountering a change, must continue to monitor for successive changes. We call this challenging scenario *continuous monitoring* to distinguish it from the traditional setting which generally monitors for only a single changepoint. Additionally, continuous monitoring demands that there be limited dependence on the setting of parameters controlling the performance of the algorithms.

One of the main contributions of this thesis is the development of an efficient, fully sequential change detector for the mean of a univariate stream in the continuous monitoring context. It is competitive with algorithms that are the benchmark in the single changepoint setting, yet our change detector only requires a single control parameter, which is easy to set. The multivariate extension provides similarly competitive performance results. These methods are applied to monitoring foreign exchange streams and computer network traffic.

# Acknowledgements

First and foremost, I would like to thank my supervisors Niall Adams and Nicholas Heard for their incredible patience and support over the last four years.

I am very grateful to Imperial College for funding my studies and providing me with a Roth Doctoral Fellowship. Without the financial support, this work would not have been possible.

Thank you to Christoforos Anagnostopoulos for many helpful discussions and patient explanations, and for the chance to participate in some interesting projects.

Thanks to my office mates, past and present, for all the advice, explanations, quick tips, shared lunches, and for keeping me thoroughly entertained. I consider myself fortunate to count many of you as friends.

Finally, thanks to my parents and family, and Qing, for their love, support and patience over these past few years.

This thesis was typeset using LaTeX, and (almost all) tables and plots were produced in R using the xtable and ggplot2 [158] packages, respectively.

Dean Bodenham

# Table of contents

# Glossary

**SPC** Statistical process control. 14

$N$ The number of observations observed so far, or the number of terms in the weighted sum of $Q_N$ (which only occurs in Chapter 7). 19

$\tau$ the time the *true* changepoint occurs. 19

$\widehat{\tau}$ the time the changepoint $\tau$ is detected. 21

**FFF** Fixed forgetting factor. 33

$\lambda$ A fixed forgetting factor valued in $[0, 1]$. 33

**AFF** Adaptive forgetting factor. 41

$\overrightarrow{\lambda}$ An adaptive forgetting factor that takes values in $[0, 1]$. 41

$\bar{x}_N$ The mean of observations $x_1, x_2, \ldots, x_N$. 39

$\bar{x}_{N,\lambda}$ Fixed forgetting mean for observations $x_1, x_2, \ldots, x_N$ with forgetting factor $\lambda$. 33

$\bar{x}_{N,\overrightarrow{\lambda}}$ Adaptive forgetting mean for observations $x_1, x_2, \ldots, x_N$ with adaptive forgetting factor $\overrightarrow{\lambda}$. 42

$\eta$ The step size, or learning rate, used in the gradient descent step to update the adaptive forgetting factor. 43

$d$ The dimension of a multivariate observation. 91

$\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}}$ Multivariate adaptive forgetting factor mean for observations $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N$. 91

**ARL0** The in-control Average Run Length for a change detection algorithm, i.e. the average number of observations before an algorithm falsely signals that a change has occurred. 20

**ARL1** The out-of-control Average Run Length for a change detection algorithm, i.e. the average number of observations between a changepoint occurring and the algorithm detecting that a change has occurred. 20

$CCD$ Proportion of true changepoints that are correctly detected, a performance metric in the continuous monitoring context. 78

$DNF$ Proportion of detected changepoints that are not false detections, a performance metric in the continuous monitoring context. 78

**cdf** Cumulative distribution function. 17

**pdf** Probability distribution function. 28

**EWMA** Exponentially Weighted Moving Average; a control chart originally described in [128]. 22

**CUSUM** Cumulative Sum; a control chart originally described in [116]. 22

# Chapter 1

# Introduction

*Data streams* can be defined as potentially unending sequences of ordered observations, where each observation may be read exactly once [71]. Such a model is appropriate when the amount of continuously arriving data may be too large to practically store in memory and analyse later [115]. For example, Imperial College's computer network generates 19 GB of NetFlow [1] data every hour. Furthermore, streaming data may be arriving at a high frequency relative to the processing resources available, and so streaming data algorithms ("analytics") should be as efficient as possible [8, 107]. Another consideration is that timely decision-making may be necessary, and hence timely-output from an analytic is required. The combination of these three characteristics results in a preference for sequential or *online* statistical techniques. Moreover, it is inherent to many applications that data streams cannot be stopped while adjustments to an algorithm are made — the data will continue to arrive, whatever action is taken. This unique combination of characteristics dictates that streaming data algorithms should be as *autonomous* as possible. Therefore, algorithms deployed on data streams should have as few *control parameters* as possible. Modern data collection technology has made streaming data ubiquitous; examples of data streams abound in fields as diverse as fraud detection [61], astronomy [11], finance [161], computer networks [44] and medicine [25, 131].

Data streams are time-varying in nature [73]; as the world changes, so are its measurements expected to change. However, we may expect there to be periods of stability, where the observations are generated by some underlying stochastic mechanism. A simple formulation would be that the stream is generated by underlying probability distributions, which irregularly change from one regime to the next. This is the context pursued in this thesis. However, the data stream could equally be the residual sequence output from a model.

With this formulation in mind, a problem of particular interest is the detection of *changepoints* in a data stream, the points at which the underlying distribution of the stream changes from one regime to another. Note that all the work in this thesis focuses on discrete-time sequences.

Traditionally, the area of statistics concerned with timely change detection is known as *statistical process control* (SPC) [9]. For the streaming context outlined above SPC has a number of limitations, primarily that sequences are considered to only contain a *single* changepoint. It is assumed that the process can be stopped, once a changepoint is located, while changes are made to the process generating the sequence [66]. In contrast, streaming data will have *multiple* changepoints, and usually cannot be stopped. For example, in the case of foreign exchange streams, detecting a change may trigger a trading action, but the flow of data will not stop. Many SPC methods further assume that the underlying distributions (and parameters) of the sequence are known, and may require computationally-intensive calculations [65, 66]. Although traditional SPC methods may not be well-suited to streaming data analysis, they find wide application in the area of quality control.

In order to make a clear distinction with SPC, we shall refer to the problem of detecting multiple changes in streaming data as *continuous monitoring*. Besides the applications mentioned above [61, 11, 161, 44, 25, 131], continuous monitoring has particular application in computer network monitoring [29, 13, 16], intrusion detection systems (IDS) [148, 149], and detecting denial of service (DoS) attacks [12]. Measuring the performance of a change detector in the continuous monitoring context is subtle, and requires the development of additional performance metrics. Moreover, in this setting there is a specific need for any control parameters to be automatically set [141]. The framework below addresses this need while developing specific estimation and change detection methodology.

The bulk of this thesis is dedicated to the development of a *forgetting factor* framework. This framework is used to sequentially and adaptively estimate the current mean and variance of a stream, by placing greater weight on more recent observations and thereby "forgetting" older observations. This is a very natural formulation which has been studied before [3, 69], but here we undertake a more rigorous approach in defining an *adaptive* forgetting factor scheme and its extension to a change-detection framework. This will allow for an essentially parameter-free method for tracking the mean and variance for any distribution on a data stream. Methods are developed to use these adaptive estimates of the mean and variance to provide a change detector in a streaming data context. There are at least three benefits to this approach. First, the forgetting factor framework provides good estimates of the time-varying mean and variance, no matter how many changes occur in the stream. As remarked above, data streams are expected to have multiple, unexpected changes. Second, when these estimates are used for change detection, only a single control parameter needs to be specified. Most practical change detection schemes require at least two control parameters to detect even a single change, so this is an important reduction of the burden on the analyst. Recalling the continuous monitoring context above, a change detector on a data stream needs to run autonomously, since intervention to reset control parameters is impractical. This is one motivation for using the adaptive forgetting factor framework in this context. Another advantage of the forgetting factor formulation is that it may recover from missed detections better. A further benefit is that these methods are, computationally, very efficient. Finally, this approach is very flexible, and can be applied to detecting a change in the mean or variance of a stream. While there are Bayesian approaches to parameter estimation (e.g. [114]), these are not considered in this thesis due to their computational complexity.

## Description of the thesis

Chapter 2 provides background for change detection that will be required for later chapters. Descriptions of the Shewhart, CUSUM and EWMA charts are given, and performance metrics such as the Average Run Length (ARL) are discussed. In addition, a review of chi-squared variables is given, and a brief discussion of moments and cumulants is included, since these form the foundation of the methods compared in Chapter 7.

Chapter 3 describes the forgetting factor framework for adaptively estimating the mean. The fixed forgetting factor formulation is explored before introducing the adaptive forgetting factor methodology. This adaptive procedure optimises the control parameter by taking a single step in the direction of a suitably defined derivative. A modest contribution here is the formal definition of the derivative with respect to the adaptive forgetting factor, which makes rigorous previously heuristically-defined update equations. The chapter concludes with a discussion on theoretically optimal values for a forgetting factor, which suggests that the adaptive forgetting factor has almost ideal behaviour.

Chapter 4 describes how the forgetting factor framework can be used to construct a change detector. This approach is developed following the approach in the SPC literature, by first using this method to detect a *single* change in the mean of a univariate stream. Several approaches are discussed for the construction of a decision rule for a change detector, from assuming that the underlying distribution is normal (but with unknown parameters), to a distribution-free method, using probabilistic arguments. A simulation study explores the behaviour of the different forgetting factor change detection methods, as well as the standard CUSUM and EWMA schemes. This a precursor to the next chapter, where the schemes will be more carefully compared in the continous monitoring setting.

Chapter 5 extends the methodology of Chapter 4 to the novel and challenging case of detecting multiple changes in a stream, which we refer to as *continous monitoring*. The standard performance metrics are no longer adequate, and so additional metrics are introduced. Again, a simulation study is performed to compare our forgetting factor methodology to the standard algorithms, which shows similar performance among the algorithms. Notably, our method has the added advantage of only requiring a single control parameter. Finally, the adaptive forgetting factor change detector is compared to an optimal *offline* algorithm on real data obtained from a foreign exchange stream, and shows remarkable agreement in the changepoints detected by the offline algorithm. This contribution is summarised in [15]. This work has also been adapted and applied to a computer monitoring application in [16].

Building on the preceding chapters, Chapter 6 provides an extension of continous monitoring to the case of detecting a change in the mean of a *multivariate* stream. This is a significant contribution, since it appears to be the first method dedicated to sequentially detecting multiple changes in a multivariate stream. Since there are no obvious candidates for comparison, a recently-published self-starting multivariate method [65] from the SPC literature is adapted as a benchmark for comparison in a simulation study. Although our forgetting factor method is computationally far more efficient and only requires a single control parameter, it performs at least as well as — if not better — than this state-of-the-art method. This approach has been published with an application to monitoring computer networks [13].

Chapter 7 describes a general framework for analysing approximate methods for computing the cumulative distribution function (cdf) of weighted sums of arbitrary random variables. The work in this chapter is used to enable the construction of a change detector for the variance, as described in Chapter 8. This general framework is applied to analysing four approximate methods for computing the cdf of a weighted sum of *chi-squared* random variables. These methods were chosen because they utilise moment-matching techniques and are suitable for use in a streaming data context. The main contribution here is that this analysis is far more comprehensive than any previous study. An interesting result is that these four moment-matching methods increase in accuracy as the number of terms in the weighted sum increases — this has not been shown before. The result of the analysis is that a three-moment approximation is recommended for use by most practitioners. This contribution is summarised in [14].

Chapter 8 presents first steps toward change detection for the variance of a stream, presenting equations for the forgetting factor variance. Assuming the stream is normally distributed, an extension of Cochran's theorem [18] allows the forgetting factor variance to be written as a weighted sum of chi-squared random variables. Equations for the first three cumulants are derived in order to allow the methods analysed in Chapter 7 to be used to obtain a decision rule for signalling a change. A formulation for the adaptive forgetting factor variance is presented along with a discussion about the choice of cost function. Potential application to a distribution-free change detector for the mean is discussed.

Chapter 9 concludes the thesis with a summary of the main results and a discussion on potential future work.

Derivations for equations used in the text are provided in the appendix.

# Chapter 2

# Background

This chapter reviews standard definitions and methodology used throughout this thesis. Section 2.1 reviews some background and terminology for change detection. Concepts such as *control charts* and *burn-in period* are introduced, standard algorithms and performance measures are described, and challenges of parameter estimation are discussed. This background is required for Chapters 4, 5, 6 and 8.

## 2.1 Change detection and streaming data

The goal of a change detection algorithm is to detect a change in the probability distribution of a sequence of random observations. One example is the following situation: suppose we have observations $x_1, x_2, \ldots, x_N$ generated from the random variables $X_1, X_2, \ldots, X_N$ which are independent and distributed according to distribution $D_0$ or $D_1$, such that

$$X_1, X_2 \ldots, X_\tau \sim D_0, \tag{2.1}$$

$$X_{\tau+1}, \ldots, X_N \sim D_1, \tag{2.2}$$

where the **changepoint** $\tau$ is unknown. In this thesis, since we are dealing with streaming data, $N$ will indicate the number of observations *observed so far*. Two statistics that are commonly monitored for change are the mean and variance. We say $|\mathrm{E}[X_\tau] - \mathrm{E}[X_{\tau+1}]|$ is the **size of the change** in the mean, and $|\mathrm{Var}[X_\tau] - \mathrm{Var}[X_{\tau+1}]|$ is the size of the change in

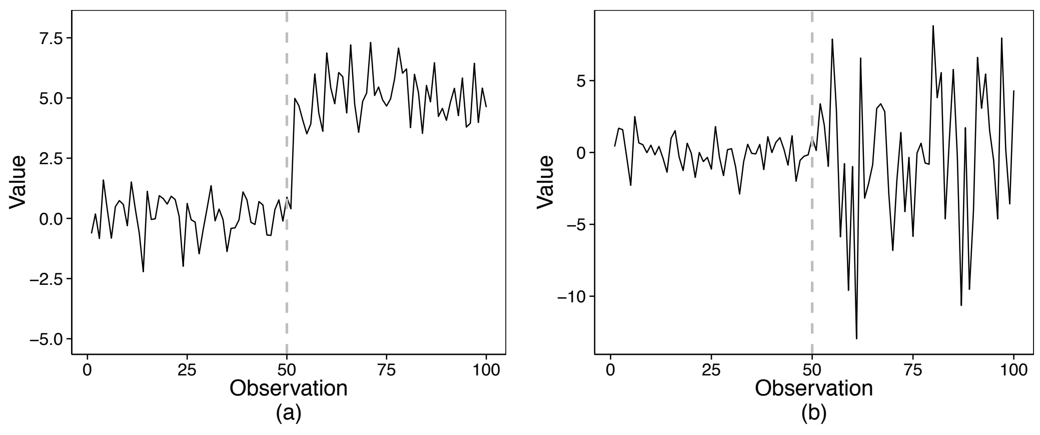

Figure 2.1: (a) a change in the mean, and (b) a change in the variance. The vertical lines indicate the true change points at $\tau = 50$.

the variance.

Figure 2.1 shows two graphs illustrating a change in the mean and a change in the variance of a sequence of observations. This particular example assumes we have a fixed data set of size $N$, with only one changepoint $\tau$. Traditional sequential procedures often require that the pre-change and post-change distributions (and perhaps the parameters of those distributions) are known. Here $N$ is not very large ($N = 100$), and all the observations can be stored and then analysed in order to detect $\tau$ with an offline algorithm. Standard references for online (sequential) change detection and change detection with adaptive filtering are [9] and [58]. Two good reviews can be found in [90] and [145].

In recent years, the problem of detecting changes in data streams has arisen in areas such as astronomy [11], computer network traffic analysis [106] and finance [38]. A **data stream** is a (potentially unending) sequence of ordered data points $x_1, x_2, \ldots$ generated from random variables $X_1, X_2, \ldots$ [71, 115]. Detecting changes in a data stream presents a number of challenges [8]:

1. the number of observations may be very large, making it impractical to store and then analyse the data in an offline manner,

2. the data may be arriving at a very high rate, requiring the change-detection algorithms to be computationally efficient,

3. the pre-change and/or post-change distributions are usually unknown, or at least their parameters may be unknown,

4. there may be multiple change points, requiring us to consider restarting the algorithm once a change has been detected. This is further discussed in Section 2.1.8,

5. timely detection may be critical — possibly even real-time.

This combination of features forces us to abandon attempts at offline analysis, and instead we only consider **online** change detection algorithms. Another consequence of the above points is that the changes in the stream are likely to be of *different sizes*.

The following sections describe issues related to change detection performance, basic change detection and charting procedures. In the streaming context, issues relating to unknown parameters and restarting occur and basic approaches to these issues are discussed.

### 2.1.1 Performance measures: the average run length

The perfect change detection algorithm would not detect a change until one has occurred, and when a change does occur, it would detect that change immediately. However, this ideal will never be achieved due to stochastic variation. In practice there will be times when an algorithm detects a change when none has occurred (a false alarm), and when a change actually does occur, there will always be some delay in that change being detected. This gives rise to two standard performance measures, ARL0 and ARL1, where ARL is an acronym for *average run length*.

We define the ARL0 of an algorithm to be the average time between false alarms raised by that algorithm, while we define the ARL1 to be the average delay between a change occurring and that change being detected. We can define these more precisely as follows: let the data stream be defined as in Equations (2.1) and (2.2), so that the changepoint is at time $\tau$. Let $\widehat{\tau}$ denote the time when the change is detected. Then we can define these measures as

$$\begin{aligned} \text{ARL0} &= \text{E}[\widehat{\tau}|X_1, X_2, \cdots \sim D_0], \\ \text{ARL1} &= \text{E}[\widehat{\tau} - \tau|X_1, \ldots, X_\tau \sim D_0; X_{\tau+1}, \ldots, X_{\widehat{\tau}} \sim D_1]. \end{aligned}$$

Ideally we would like our algorithms to have a high ARL0 and a low ARL1. However, tuning an algorithm's parameters to achieve a desirable value for one of these measures will have a negative effect on the other measure. We shall revisit this topic in Chapter 4 and extend the concept to multiple changes in Chapter 5.

Note that in the SPC literature it is usual to report tables of Monte Carlo simulation results reporting ARL0 and ARL1. However, it is uncommon to report associated estimates of uncertainty (e.g. [66, 100, 26]). In this thesis we prefer to report the estimated standard deviation of the run lengths, denoted SDRL0 and SDRL1. The magnitude of these uncertainty estimates is consistent with that which has been reported in the SPC literature (e.g. [81]).

### 2.1.2 Control charts

The idea of a **control chart** was first described in [138], with the original motivation being the detection of change in manufacturing processes for the purposes of quality control. A control chart consists of points $z_1, z_2, \dots$ representing a **statistic** and control limits $a, b$, with $a < b$. When $z_j \in (a, b)$ we say the process is **in control**, and when $z_j \notin (a, b)$ we say that the process is **out of control**. We call

$$z_j \in (a, b) \Rightarrow \text{in control} \tag{2.3}$$

$$z_j \notin (a, b) \Rightarrow \text{out of control} \tag{2.4}$$

a **decision rule**. Note that here we are using $z_j$ to represent a sequence of statistics in order to distinguish from the observations $x_j$. We call $a$ the lower control limit (LCL) and $b$ the upper control limit (UCL). We call $\widehat{\tau}$ the detected **changepoint** of the data stream if $z_{\widehat{\tau}} \notin (a, b)$, but $z_j \in (a, b)$ for all $j < \widehat{\tau}$, while we reserve the letter $\tau$ for the true changepoint. Figure 2.2 is an example of a control chart, and the data stream has a changepoint at $\widehat{\tau} = 71$. Two well-known control chart schemes are CUSUM and EWMA, first described in [116] and [128], respectively, and discussed below. The genesis of control chart methodology is due to Walter Shewhart [138], and his control chart is described next.

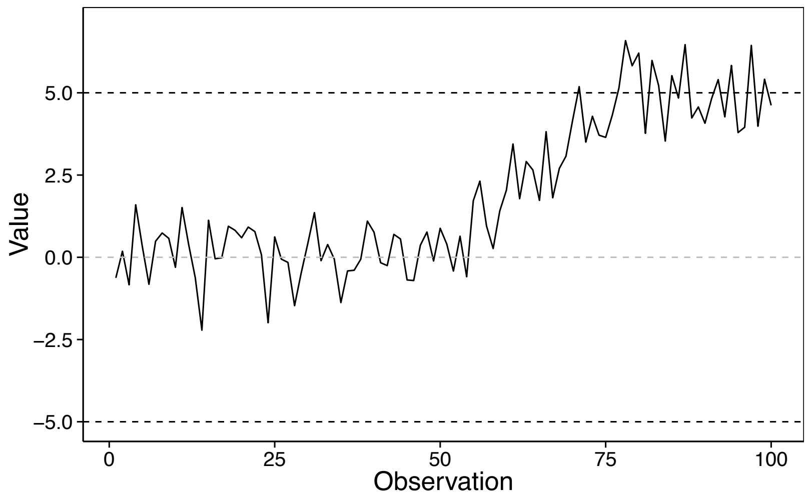

Figure 2.2: A control chart for detecting a change in the mean. The control limits are indicated by the black dashed lines, while the in-control mean is indicated by the grey dashed line.

### 2.1.3 Shewhart charts

Suppose the stream of observations $x_1, x_2, \ldots$ is generated from random variables $X_1, X_2, \ldots$, and it is known that for $k \leq \tau$

$$\begin{aligned} \mathrm{E}[X_k] &= \mu, \\ \mathrm{Var}[X_k] &= \sigma^2. \end{aligned}$$

Then the control limits for a Shewhart chart are defined as

$$\begin{aligned} a &= \mu - \delta\sigma, \\ b &= \mu + \delta\sigma, \end{aligned}$$

where $\delta$ is a parameter controlling the sensitivity of the chart. Therefore, if

$$\begin{aligned} x_1, x_2, \ldots x_{t-1} &\in (a, b), \\ x_t &\notin (a, b), \end{aligned}$$

then the changepoint $\widehat{\tau} = t$. So, for the Shewhart chart, the chart statistics are the observations themselves, i.e. $z_k = x_k$. This will not be the case for later methods.

While $\delta$ is usually set to $\delta = 3$, Figure 2.2 is an example of a Shewhart chart with $\delta = 5$. If $\mu$ and $\sigma^2$ are unknown, then the sample mean $\bar{x}$ and sample variance $s^2$ can provide estimates, and the control limits become

$$a = \bar{x} - \delta s,$$
$$b = \bar{x} + \delta s.$$

The Shewhart chart is known to be effective at detecting large changes, however it is insensitive to small changes in the mean [110].

### 2.1.4 CUSUM

The Cumulative Sum (CUSUM) algorithm was first proposed in [116] and shown to be optimal under certain assumptions [112], in the sense of [97]. However, if the distributions' parameters are unknown, as is often the case, then this optimality is not guaranteed. If the stream is initially N$(\mu, \sigma^2)$-distributed, the CUSUM statistic $S_j$ is defined as: $S_0 = \mu$, and

$$S_j = \max(0, S_{j-1} + x_j - k\mu), \qquad j \in \{1, 2, \dots\}$$

in order to detect an increase in the mean. A change is detected when $S_j > h\sigma$. A statistic to detect a decrease in the mean can be similarly defined.

Here the *control parameters* $k$ and $h$ need to be chosen. These values are often chosen according to the needs of the application, and specifically the magnitude of the changes one is trying to detect. For example, in the context of setting parameters related to $h$ and $k$ in the self-starting CUSUM procedure of [62], it is recommended to used standard tables such as those in [99]. This selection is based on the anticipated change size, $|E[X_\tau] - E[X_{\tau+1}]|$ (see Section 2.1).

Although [112] showed that CUSUM is optimal, this is only the case when both the pre- and post-change distributions (and the parameter values) are known. If this is not the case, we do not have such strong theoretical guarantees. Moreover, the sensitivity of the change detector when deployed in practice will depend on the choice of the parameters $k$ and $h$. According to [134], three common (pairs of) choices for the control parameters are

$$\begin{aligned} k &= 0.25, h = 8.00, \\ k &= 0.50, h = 4.77, \\ k &= 1.00, h = 2.49. \end{aligned}$$

Although these pairs of control parameters may each perform well in a given situation, it is not obvious, given a data stream, which pair should be used. Indeed, as a stream evolves and changes occur, it is unlikely that a fixed pair of parameters will continue to be optimal after each change. This is part of the motivation for introducing our adaptive forgetting factor scheme in Chapters 3 and 4.

### 2.1.5 EWMA

The EWMA (Exponentially Weighted Moving Average) control chart is another online change detection method, first described in [128]. Suppose we have observations $x_1, x_2, \ldots$ sampled from a distribution with known mean $\mu$ and variance $\sigma^2$. We then define the new variables $Z_1, Z_2, \ldots,$ by

$$\begin{aligned} Z_0 &= \mu \\ Z_j &= (1-r)Z_{j-1} + rx_j \end{aligned}$$

where $r \in [0,1]$ acts as an exponential forgetting factor. It can be shown [128] that the standard deviation of $Z_k$ is

$$\sigma_{Z_j} = \left(\sqrt{\frac{r}{2-r}\left[1-(1-r)^{2j}\right]}\right)\sigma.$$

If we wanted to detect an increase in the mean, a change would be detected when

$$Z_j > \mu + L\sigma_{Z_j},$$

where $L$ is a control parameter chosen to give the algorithm a desired performance in terms of ARL0 or ARL1. It is also possible to modify EWMA to perform two-sided detection. According to [100], the parameter $r$ is usually chosen to be $0.05 < r < 0.5$ for detecting small shifts, while $L$ is usually chosen to be close to 3.

The original paper [128] showed that, in practice, EWMA is good at detecting small shifts in the process mean. Interestingly, [100] showed that the properties of EWMA are "similar" to those of CUSUM schemes, a point further discussed in [103]. However, this was the case when an "optimal" choice of parameters were used for EWMA in comparison to a CUSUM scheme using a seemingly arbitrary (not necessarily optimal) choice of parameters. This point was recently revisited in [67], where it was shown that EWMA can outperform CUSUM if the size of the change is smaller than that for which the CUSUM parameters were selected.

For our present purpose, a central difficulty with both CUSUM and EWMA is the selection of control parameters. We shall return to this in Chapter 5 in the context of continuous monitoring.

### 2.1.6 Other sequential change detection methods

Besides CUSUM and EWMA, three other well-known sequential change detection methods, are the *Shiryaev-Roberts* method, the *generalised likelihood ratio (GLR)* method, and the *changepoint model*.

The Shiryaev-Roberts procedure was independently created by A. N. Shiryaev [139] and S. W. Roberts (the creator of EWMA) [129], and is defined as the sum of a sequence of likelihood ratios. It requires knowledge of the pre- and post-change distributions, but has some optimality properties [122].

The GLR method is similar to CUSUM, and is reviewed in [9, 90]. It also requires knowledge of the pre- and post-change distributions and has recently been applied to detecting changes in the mean and variance [126]. Moreover, it can perform well in relation to CUSUM and EWMA [60].

The changepoint model was first proposed in [66] for detecting a change in a univariate mean, and has since been extended to detecting a change in the variance [68], multivariate change detection [164] and the non-parametric setting [133, 132]. However, all implementations of this method require the storage of an ever-growing sequence of statistics, so it can be considered unsuitable for use on streaming data.

All of these methods have been enhanced or extended in some way, for instance to detecting a change in the variance, or to the multivariate setting (e.g. [154, 68]). While each of these methods has its supporters, there is no clear evidence that any of these should be preferred over CUSUM and EWMA.

### 2.1.7 Estimating the parameters of the known distribution

In Section 2.1.4, we mentioned that CUSUM is optimal at detecting a change, but only when the pre- and post-change distributions' parameters are known. With a data stream, however, we may not know what the distributions are, let alone the values of the parameters. However, there are cases where we may confidently model a process by a given family of distributions, but we will not know the values of the parameters. For example, we may know that a given set of observations are sampled from a normal distribution $N(\mu, \sigma^2)$, but we do not know the values of $\mu$ and $\sigma^2$. In these cases, we could try and estimate the values of the parameters during an initial monitoring period, assuming that no change occurs. This is called a **burn-in period**.

However, if the burn-in period is too short, our parameter estimates will be inaccurate, which will then lead to poor performance of the change detection algorithm. An extensive literature review of this approach can be found in [77], where the key issues that are discussed include sample size (for the monitoring period), the impact of parameter estimation on algorithm performance, and other possible approaches. In the continuous monitoring scenario considered in Chapter 5, the burn-in used is relatively short and the changepoints occur after short intervals. This is in contrast to traditional SPC approaches, which usually only consider a single changepoint occurring after a long period of stationarity.

It is notable that much literature involving a burn-in period is concerned with both a long burn-in and detecting a single changepoint. The work developed in Chapter 5 is concerned with much shorter regime shifts which imposes constraints on the length of the burn-in period.

### 2.1.8 Restarting an algorithm after a changepoint

Suppose we are monitoring a data stream $x_1, x_2, \ldots$, which we assume is sampled from a distribution $D$, and detect a change at $\widehat{\tau}$. Once a change has been detected at time $\widehat{\tau}$, this signals that (at least) the parameter values of the distribution have changed. In a streaming data context, it is likely that we would wish to continue monitoring the 'new' data stream $x_{\widehat{\tau}}, x_{\widehat{\tau}+1}, \ldots$ for future changes. This leads to a problem: our change detection algorithm requires the distribution's parameter values, but it is rare that we will know the values of the post-change parameters, and we cannot use the pre-change parameter values (since a change has occurred).

In this situation, one approach would be to estimate the post-change parameters during a new burn-in period, and then use these estimates to detect a change in the new stream $x_{\widehat{\tau}}, x_{\widehat{\tau}+1}, \ldots$. This is the approach adopted later in this thesis.

# Chapter 3

# Adaptive filtering for the mean

The method of change detection in this thesis relies on being able to accurately estimate parameters of a stream such as the mean or variance. Supposing that such stream parameters are well estimated, changepoints are signalled when these statistics deviate from their current estimates beyond a certain threshold, following the conventional approach discussed in Section 2.1.2. Obtaining suitable thresholds for the mean is discussed in Chapter 4, and monitoring the variance is discussed in Chapter 8, but both chapters rely on the methodology developed in this chapter.

In this chapter, a framework utilising adaptive forgetting factors is developed that allows stream parameters to be adaptively estimated and will form the basis of change detection methods discussed in later chapters. Previous work on adaptive forgetting includes [7, 6, 50, 69, 3]. The definition of the derivative with respect to the adaptive forgetting factor $\overrightarrow{\lambda}$, described in Section 3.3.2, is a key contribution of this thesis as it allows for recursive update equations that do not require the underlying distribution of the stream to be known. This method is well-suited to a streaming data context (see Chapter 1), as everything can be computed sequentially.

This chapter proceeds as follows: Section 3.1 introduces the idea of estimating the mean, Section 3.2 describes the fixed forgetting factor framework, Section 3.3 describes the adaptive forgetting factor framework, and Section 3.4 investigates the optimal values of fixed and adaptive forgetting factors when the pre-change and post-change stream parameters are known. In Section 3.3.8 a result from [3] shows that in a special case the optimal linear filter is closely related to this adaptive forgetting factor scheme.

## 3.1 Estimating the mean

Suppose that the stream $x_1, x_2, \dots$ is generated by the random variables $X_1, X_2, \dots$ and that $N$ observations have been observed so far. The goal is to estimate the current mean of the stream, namely $\mathrm{E}\left[X_N\right]$. This estimate will be used for detecting changepoints in the stream, which is the subject of Chapter 4. One way to estimate $\mathrm{E}\left[X_N\right]$ would be to compute the sample mean of the stream,

$$\bar{x}_N = \frac{1}{N}\sum_{i=1}^{N} x_i.$$

If the stream had been stationary[*] up until this point, i.e.

$$\mathrm{E}[X_i] = \mu, \qquad i = 1, 2, \dots, N,$$

it follows that $\widehat{\mu} = \bar{x}_N$ would estimate $\mu$ well, since

$$\mathrm{E}\left[\bar{X}_N\right] = \mathrm{E}\left[\frac{1}{N}\sum_{i=1}^{N} X_i\right] = \frac{1}{N}\sum_{i=1}^{N}\mathrm{E}\left[X_i\right] = \frac{1}{N}\sum_{i=1}^{N}\mu = \mu.$$

If, however, there had been a change in the mean at some point $\tau < N$,

$$\mathrm{E}[X_i] = \left\{ \begin{array}{ll} \mu', & i = 1, 2, \dots, \tau \\ \mu, & i = \tau+1, \tau+2, \dots, N, \dots \end{array} \right. , \qquad \mu' \neq \mu \tag{3.1}$$

then $\widehat{\mu} = \bar{x}_N$ may not estimate $\mu$ very well, if the difference between $\mu$ and $\mu'$ is large. In fact, a better estimate would be obtained by only taking the mean of those observations that occur after the changepoint $\tau$,

$$\widehat{\mu} = \frac{1}{N-\tau}\left[x_{\tau+1} + x_{\tau+2} + \cdots + x_N\right]. \tag{3.2}$$

[*]At present, only the weaker condition of stationarity is needed for tracking the mean.

Of course, since the location of the changepoint $\tau$ is generally unknown, such an approach is infeasible. An approach to approximate Equation (3.2) by computing a weighted sum of the observations $x_1, x_2, \ldots, x_N$ is explored in the next section.

## 3.2 Fixed forgetting factor mean

The method at the core of this thesis attempts to estimate the the current mean $\mu$ of a non-stationary stream as in Equation (3.1) by computing a weighted sum of the observations $x_1, x_2, \ldots, x_N$ using an (exponential) **fixed forgetting factor** $\lambda \in [0, 1]$. The **fixed forgetting factor mean** $\bar{x}_{N,\lambda}$ after $N$ observations is defined as

$$\bar{x}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} x_i, \tag{3.3}$$

where the **effective sample size** $w_{N,\lambda}$ is defined as

$$w_{N,\lambda} = \sum_{i=1}^{N} \lambda^{N-i}. \tag{3.4}$$

Writing these two equations out in full,

$$\begin{aligned} \bar{x}_{N,\lambda} &= \frac{1}{w_{N,\lambda}} \left[ \lambda^{N-1} x_1 + \lambda^{N-2} x_2 + \cdots + \lambda x_{N-1} + x_N \right], \\ w_{N,\lambda} &= \lambda^{N-1} + \lambda^{N-2} + \cdots + \lambda + 1, \end{aligned}$$

it can be immediately seen that

- for $\lambda = 1$, $\bar{x}_{N,\lambda} = \bar{x}_N$, the sample mean,
- for $\lambda = 0$, $\bar{x}_{N,\lambda} = x_N$, the most recent observation.

This is where the forgetting factor gets its name: the closer $\lambda$ is to zero, the more that $\bar{x}_{N,\lambda}$ "forgets" early observations. Excluding these two limit cases, for $\lambda \in (0, 1)$ the fixed forgetting factor (FFF) mean is the weighted sum of all the observations $x_1, x_2, \ldots, x_N$, with more weight placed on recent observations and less weight on older observations.

Section 3.2.3 illustrates how different values of $\lambda$ can affect the estimation of the current mean of the stream. Before this, we first discuss the effective sample size $w_{N,\lambda}$ in Section 3.2.1 and compute the expectation and variance of $\bar{x}_{N,\lambda}$ in Section 3.2.2. These statistics will be used to reason about setting a value for $\lambda$ and will be crucial when constructing change detection rules in Chapter 4.

### 3.2.1 The effective sample size

The quantity $w_{N,\lambda}$ defined in (3.4) is often referred to as the **effective sample size** since it approximates the number of observations over which we are averaging; for the two trivial cases

- if $\lambda = 1$ then $w_{N,\lambda} = N$,
- if $\lambda = 0$ then $w_{N,\lambda} = 1$.

However, if $\lambda \in (0, 1)$ then $w_{N,\lambda} \in (1, N)$. In other words, the sum of the weights measures the effective size of the sample used to compute $\bar{x}_{N,\lambda}$. It is worth investigating the value of $w_{N,\lambda}$ in the limit. If $\lambda \in (0, 1)$ then

$$w_{N,\lambda} = \sum_{i=1}^{N} \lambda^{N-i} = \frac{1-\lambda^N}{1-\lambda}$$

and as $N \to \infty$,

$$w_{\infty,\lambda} \equiv \lim_{N\to\infty} w_{N,\lambda} = \frac{1}{1-\lambda}. \tag{3.5}$$

Equation (3.5) indicates that, for example, if $\lambda = 0.95$ and $N$ is large, then the effective sample size $w_{\infty,0.95} = 20$. Equation (3.5) will be significant in Section 3.2.5, when the FFF and EWMA schemes are compared. Note that the effective sample size is referred to as the *memory* in [69].

### 3.2.2 Expectation and variance of forgetting factor mean

Following the definition of the forgetting factor mean $\bar{x}_{N,\lambda}$ in Equations (3.3) and (3.4), suppose that for $i = 1, 2, \ldots, N$, the random variables $X_i$ generate the observations $x_i$. We can then define the forgetting factor mean of the random variables $X_i$ by

$$\bar{X}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} X_i, \tag{3.6}$$

where $w_{N,\lambda}$ is again defined in Equation (3.4). Suppose further that the random variables $X_i$ are independent and identically-distributed (i.i.d.) with expectation and variance

$$\begin{aligned} \mathrm{E}\left[X_i\right] &= \mu, \\ \mathrm{Var}\left[X_i\right] &= \sigma^2. \end{aligned}$$

Then the expectation of $\bar{X}_{N,\lambda}$ is

$$\begin{aligned} \mathrm{E}\left[\bar{X}_{N,\lambda}\right] &= \mathrm{E}\left[\frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} X_i\right], \\ &= \frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} \mathrm{E}\left[X_i\right], \\ &= \frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} \mu, \\ &= \mu, \end{aligned}$$

and (using the independence assumption) the variance of $\bar{X}_{N,\lambda}$ is

$$\begin{aligned} \mathrm{Var}\left[\bar{X}_{N,\lambda}\right] &= \mathrm{Var}\left[\frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} X_i\right], \\ &= \frac{1}{(w_{N,\lambda})^2} \sum_{i=1}^{N} \left(\lambda^{N-i}\right)^2 \mathrm{Var}\left[X_i\right], \\ &= \frac{1}{(w_{N,\lambda})^2} \sum_{i=1}^{N} \left(\lambda^2\right)^{N-i} \sigma^2, \\ &= \left(u_{N,\lambda}\right) \sigma^2, \end{aligned}$$

where we define

$$u_{N,\lambda} = \frac{w_{N,\lambda^2}}{(w_{N,\lambda})^2} = \frac{1}{(w_{N,\lambda})^2} \sum_{i=1}^{N} \left(\lambda^2\right)^{N-i}. \tag{3.7}$$

In summary, $\bar{X}_{N,\lambda}$ is a random variable with expectation and variance

$$\begin{aligned} \text{E}\left[\bar{X}_{N,\lambda}\right] &= \mu, \\ \text{Var}\left[\bar{X}_{N,\lambda}\right] &= (u_{N,\lambda})\,\sigma^2. \end{aligned}$$

It is interesting to note that

$$\begin{aligned} \lim_{\lambda\to 1} \text{E}\left[\bar{X}_{N,\lambda}\right] &= \mu = \text{E}\left[\bar{X}_N\right], \\ \lim_{\lambda\to 1} \text{Var}\left[\bar{X}_{N,\lambda}\right] &= \frac{1}{N}\sigma^2 = \text{Var}\left[\bar{X}_N\right], \end{aligned}$$

which shows that the FFF mean behaves as the sample mean in the limit, as implied earlier.

### 3.2.3 The relationship between $\bar{x}_{N,\lambda}$ and $\lambda$

The closer $\lambda$ is to zero, the more that $\bar{x}_{N,\lambda}$ forgets earlier data, since greater weight is placed on recent data. This is illustrated in Figure 3.1 where the stream $x_1, x_2, \ldots x_{300}$ has been sampled from

$$X_1, X_2, \ldots, X_{100} \sim \text{N}(0, 1), \tag{3.8}$$

$$X_{101}, \ldots, X_{300} \sim \text{N}(3, 1), \tag{3.9}$$

and the value of the forgetting factor mean $\bar{x}_{1,\lambda}, \bar{x}_{2,\lambda}, \ldots, \bar{x}_{N,\lambda}$ is shown for a selection of values of $\lambda \in [0.9, 1]$.

If $M$ streams are sampled as in Equations (3.8) and (3.9), then $M$ sequences $\bar{x}_{1,\lambda}, \bar{x}_{2,\lambda}, \ldots, \bar{x}_{N,\lambda}$ can be produced, and their average $\bar{x}^{av}_{1,\lambda}, \bar{x}^{av}_{2,\lambda}, \ldots, \bar{x}^{av}_{N,\lambda}$ can be plotted, as in Figure 3.2 (which shows the average value over $M = 100$ streams).

This figure shows that when $\lambda = 0.9$, the forgetting factor mean $\bar{x}_{N,0.9}$ quickly reacts to the change at $t = 100$, and is close to $\mu = 3$ soon after the changepoint. However, if $\lambda = 1$, then $\bar{x}_{N,1}$ does not estimate the mean $\mu = 3$ very well after the changepoint (in fact

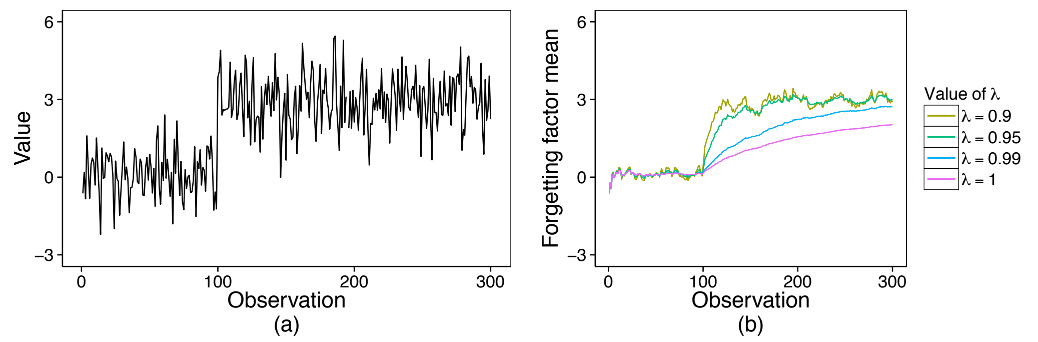

Figure 3.1: (a) A stream $x_1, x_2, \ldots, x_{300}$ sampled from $X_1, \ldots, X_{100} \sim \mathrm{N}(0,1)$, $X_{101}, \ldots, X_{300} \sim \mathrm{N}(3,1)$, and (b) the value of the fixed forgetting factor mean $\bar{x}_{N,\lambda}$ (on this stream) for different values of $\lambda$.

it is still less than 2 at observation 300). Error bars, with a width of one standard deviation, are provided at certain points (including those where the standard deviation is largest).

If, therefore, smaller values of $\lambda$ allow $\bar{x}_{N,\lambda}$ to react to changes faster, it would seem as if a value of $\lambda = 0.1$ would be better than $\lambda = 0.9$. However, recall from Section 3.2.2 that Var $\left[\bar{X}_{N,\lambda}\right] = (u_{N,\lambda})\,\sigma^2$, where $u_{N,\lambda}$ is defined in Equation (3.7). As Figure 3.3(b) shows, lower values of $\lambda$ lead to larger values of $u_{N,\lambda}$. This implies that for lower values of $\lambda$, the variance of $\bar{X}_{N,\lambda}$ is higher. This is because the effective sample size, $w_{N,\lambda}$, shown in Figure 3.3(a), is much smaller for lower values of $\lambda$. In other words, if $\lambda$ is too close to 1, then $\bar{X}_{N,\lambda}$ will be slow to react to changes, but if $\lambda$ is too small, then the behaviour of $\bar{X}_{N,\lambda}$ may be subject to large variations. This is a manifestation of the familiar tradeoff between bias and variance. Figure 3.4 has been included to show the behaviour of $w_{N,\lambda}$ and $u_{N,\lambda}$ for $\lambda \in [0.6, 0.99]$. This will be relevant in Section 3.3.5, where the issue of truncating the adaptive forgetting factor is discussed.

One might prefer to have $\lambda$ closer to 1 when the stream is not experiencing a change (for stability), but then have $\lambda$ closer to 0 after a change occurs, at least temporarily to allow $\bar{X}_{N,\lambda}$ to react to the change quickly. This leads to the idea of an adaptive forgetting factor, which is discussed in Section 3.3: one might hope that a time-varying forgetting factor could allow for both stability and quick reaction to changes. In fact, using an adaptive forgetting factor has the added benefit that the value of a fixed forgetting factor $\lambda$ no longer needs to be specified, because the algorithm developed in Section 3.3 automatically selects

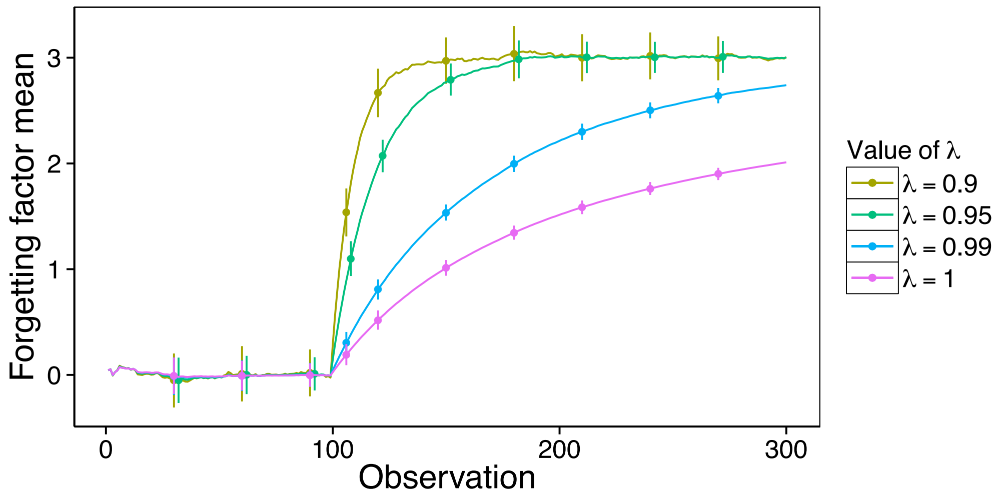

Figure 3.2: The average behaviour of $\bar{x}_{N,\lambda}$ for different values of $\lambda$, for $X_1, \ldots, X_{100} \sim \mathrm{N}(0, 1)$, $X_{101}, \ldots, X_{300} \sim \mathrm{N}(3, 1)$, averaged over 100 simulations. Error bars indicate a width of one standard deviation on either side of $\bar{x}_{N,\lambda}$.

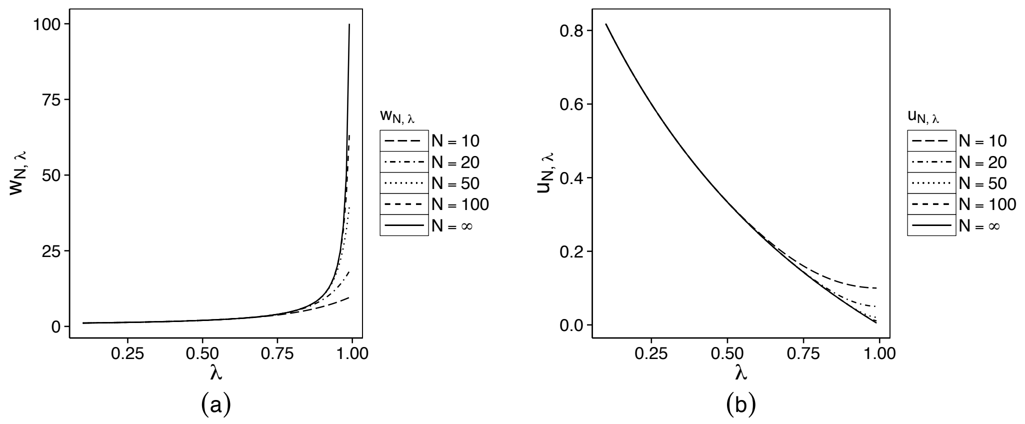

Figure 3.3: Values of (a) $w_{N,\lambda}$ and (b) $u_{N,\lambda}$ for various values of $N$, for $\lambda \in [0.1, 0.99]$.

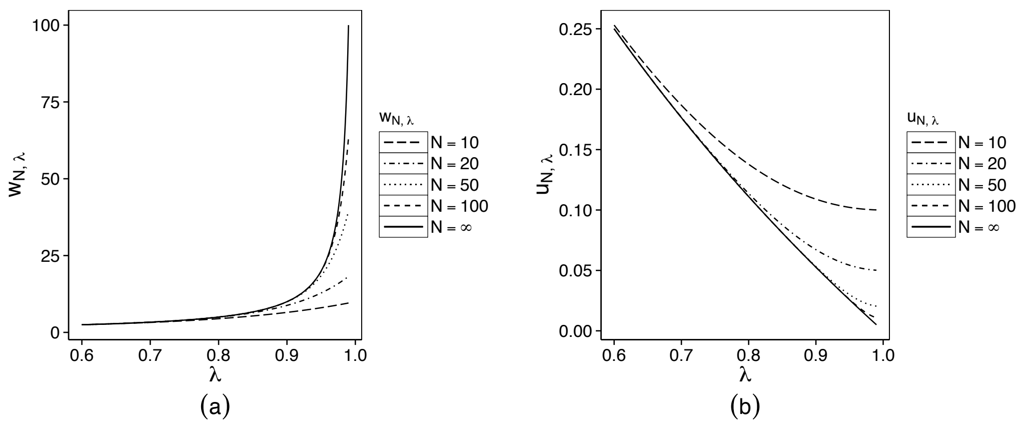

Figure 3.4: Values of (a) $w_{N,\lambda}$ and (b) $u_{N,\lambda}$for various values of $N$, for $\lambda \in [0.6, 0.99]$.

a value based on the observations so far observed. In Chapter 4 it is shown that this leads to a change detection algorithm which has fewer control parameters to be specified, in comparison to other change detection methods. First, however, in Section 3.2.4 it is shown that $\bar{X}_{N,\lambda}$ can be updated sequentially, and these sequential update equations will form the basis of the adaptive forgetting factor framework in Section 3.3.

### 3.2.4 Sequential updating

It is important for streaming data applications to have sequential update equations for $\bar{x}_{N,\lambda}$. Such equations show that the computation of $\bar{x}_{N,\lambda}$ only requires a finite number of statistics to be stored, and show that the computation per datum is of constant complexity. Moreover, the sequential update equations for the $\bar{x}_{N,\lambda}$ will form the basis for defining the adaptive forgetting factor framework.

The sample mean $\bar{x}_N$ could be computed sequentially by

$$\begin{aligned} m_N &= m_{N-1} + x_N, \\ w_N &= w_{N-1} + 1, \\ \bar{x}_N &= m_N / w_N \end{aligned}$$

for $i = 1, 2, \ldots$ and $m_0 = w_0 = 0$. Similarly, the FFF mean $\bar{x}_{N,\lambda}$ can also be defined sequentially by

$$\begin{aligned} m_{N,\lambda} &= \lambda m_{N-1,\lambda} + x_N, \\ w_{N,\lambda} &= \lambda w_{N-1,\lambda} + 1, \\ \bar{x}_{N,\lambda} &= \frac{m_{N,\lambda}}{w_{N,\lambda}}, \end{aligned} \tag{3.10}$$

for $N = 1, 2, \dots$ and $m_{0,\lambda} = w_{0,\lambda} = 0$. Alternatively, we can write

$$\bar{x}_{N,\lambda} = \left(1 - \frac{1}{w_{N,\lambda}}\right)\bar{x}_{N-1,\lambda} + \frac{1}{w_{N,\lambda}} x_N, \tag{3.11}$$

and so Equations (3.10) and (3.11) are all that is needed to recursively update $\bar{x}_{N,\lambda}$. It is shown in Appendix A.3.7 that $u_{N,\lambda}$ can also be updated sequentially by $u_{0,\lambda} = 0$ and

$$u_{N,\lambda} = \left(\frac{w_{N,\lambda} - 1}{w_{N,\lambda}}\right)^2 u_{N,\lambda} + \left(\frac{1}{w_{N,\lambda}}\right)^2. \tag{3.12}$$

The update equations in this section now provide the basis for the adaptive forgetting factor framework, discussed in Section 3.3.

### 3.2.5 Relation to EWMA

One might notice that the FFF scheme resembles the EWMA scheme described in Section 2.1.5. Indeed, the two are closely related; starting with Equation (3.11),

$$\begin{aligned} \bar{x}_{N,\lambda} &= \left(1 - \frac{1}{w_{N,\lambda}}\right)\bar{x}_{N-1,\lambda} + \frac{1}{w_{N,\lambda}} x_N \\ &= \lambda\left(\frac{1-\lambda^N}{1-\lambda^{N+1}}\right)\bar{x}_{N-1,\lambda} + \frac{1-\lambda}{1-\lambda^{N+1}} x_N, \end{aligned}$$

and then if $\lambda \in (0, 1)$, as $N \to \infty$, this becomes

$$\bar{x}_{N,\lambda} = \lambda\bar{x}_{N-1,\lambda} + (1-\lambda)x_N,$$

which is equivalent to the EWMA scheme if we set $r = 1 - \lambda$. Therefore, it would seem as if the EWMA scheme is the limit case of the FFF scheme, since as $N$ gets very large, the update equations of the FFF scheme tend to the EWMA update equation. This derivation was sketched in [36], but not commented on.

It might appear that there is little difference between the two schemes, but in fact the FFF formulation has two key advantages over EWMA. Firstly, since EWMA is essentially the limit case of FFF, the FFF formulation might react quicker to changes in the short-term; this is explored in Section 4.1.2. Secondly, the FFF formulation leads to much simpler batch definitions, which will then be used to define an adaptive forgetting factor scheme in Section 3.3.

## 3.3 Adaptive forgetting factor

The FFF scheme, like other filtering schemes such as EWMA, suffers from one major drawback: it is not clear how to set the value of $\lambda$. In this section the concept of an **adaptive forgetting factor** (AFF) $\overrightarrow{\lambda}$ is introduced, where $\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \dots)$ implies that the value $\lambda_i$ is used to downweight observations up to and including $x_i$. Other adaptive forgetting factor procedures have been discussed in [7, 6, 50, 69, 3]. Note the difference in notation: the AFF is denoted $\overrightarrow{\lambda}$, while the FFF is denoted by $\lambda$. The AFF scheme, explained in detail below, allows the value of the forgetting factor to be set automatically after each observation. Besides alleviating the significant burden of setting a value for the forgetting factor, this scheme has the nice feature that the components of $\overrightarrow{\lambda}$ will be close to 1 while the stream is in-control, but will drop in value after a change occurs in order to forget the past regime and react to the change quickly.

### 3.3.1 Adaptive forgetting factor mean

This AFF scheme closely mirrors the FFF scheme described in Section 3.2 and will lead to a change detection method in Chapter 4. Suppose the stream $x_1, x_2, \dots$ is generated by the random variables $X_1, X_2, \ldots$; the **AFF mean** $\bar{x}_{N,\overrightarrow{\lambda}}$ is defined for $N = 1, 2, \ldots$ by

$$\begin{aligned}
m_{N,\overrightarrow{\lambda}} &= \lambda_{N-1} m_{N-1,\overrightarrow{\lambda}} + x_N, && (3.13)\\
w_{N,\overrightarrow{\lambda}} &= \lambda_{N-1} w_{N-1,\overrightarrow{\lambda}} + 1, && (3.14)\\
\bar{x}_{N,\overrightarrow{\lambda}} &= \frac{m_{N,\overrightarrow{\lambda}}}{w_{N,\overrightarrow{\lambda}}},
\end{aligned}$$

where $\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \ldots)$ and $m_{0,\overrightarrow{\lambda}} = w_{0,\overrightarrow{\lambda}} = 0$ and $\lambda_0 = 1$. Alternatively, $\bar{x}_{N,\overrightarrow{\lambda}}$ can be updated by

$$\bar{x}_{N,\overrightarrow{\lambda}} = \left(1 - \frac{1}{w_{N,\overrightarrow{\lambda}}}\right) \bar{x}_{N-1,\overrightarrow{\lambda}} + \frac{1}{w_{N,\overrightarrow{\lambda}}} x_N. \tag{3.15}$$

Note the similarity between Equations (3.11) and (3.15); the FFF $\lambda$ in Equation (3.11) has simply been replaced by the AFF $\overrightarrow{\lambda}$ in Equation (3.15). Following Section 3.2.2, suppose again that the random variables $X_i$ are i.i.d. with

$$\begin{aligned}
\mathrm{E}\left[X_i\right] &= \mu,\\
\mathrm{Var}\left[X_i\right] &= \sigma^2.
\end{aligned}$$

Defining $\bar{X}_{0,\overrightarrow{\lambda}} = 0$ and $\bar{X}_{N,\overrightarrow{\lambda}}$ for $N = 1, 2, \ldots$ to be

$$\bar{X}_{N,\overrightarrow{\lambda}} = \left(1 - \frac{1}{w_{N,\overrightarrow{\lambda}}}\right) \bar{X}_{N-1,\overrightarrow{\lambda}} + \frac{1}{w_{N,\overrightarrow{\lambda}}} X_N,$$

the expectation and variance of $\bar{X}_{N,\overrightarrow{\lambda}}$ can be computed as in Section 3.2.2 to be

$$\begin{aligned}
\mathrm{E}\left[\bar{X}_{N,\overrightarrow{\lambda}}\right] &= \mu, && (3.16)\\
\mathrm{Var}\left[\bar{X}_{N,\overrightarrow{\lambda}}\right] &= (u_{N,\overrightarrow{\lambda}})\sigma^2, && (3.17)\\
u_{N,\overrightarrow{\lambda}} &= \left(\frac{w_{N,\overrightarrow{\lambda}} - 1}{w_{N,\overrightarrow{\lambda}}}\right)^2 u_{N-1,\overrightarrow{\lambda}} + \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2, \qquad u_{1,\overrightarrow{\lambda}} = 1.
\end{aligned}$$

The method for updating $\lambda_i \to \lambda_{i+1}$ — the key part of the algorithm and crucial for streaming data — is described in the next section.

### 3.3.2 Updating $\lambda_N \to \lambda_{N+1}$

Suppose that $\lambda_1, \lambda_2, \dots, \lambda_N$ have already been defined and that $\lambda_{N+1}$ should be chosen to minimise a particular cost function $L_{N+1,\overrightarrow{\lambda}}$ involving $\bar{x}_{N,\overrightarrow{\lambda}}$. Since this will be applied in a streaming data context, *online* optimisation is required. Supposing further that there is a definition for the derivative $\partial/\partial\overrightarrow{\lambda}$, one-step gradient descent [19, 17, 58] can be used to define

$$\lambda_{N+1} = \lambda_N - \eta \frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}}, \tag{3.18}$$

where $\eta$ is the step size, and $\eta \ll 1$. The remainder of this section discusses a definition for

$$\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda}}, \tag{3.19}$$

since once this derivative is defined, the derivative of any continuous function of $\bar{x}_{N,\overrightarrow{\lambda}}$ can be computed using the chain rule. Recalling that $\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \dots)$, for any $\epsilon \in \mathbb{R}$ define

$$\overrightarrow{\lambda} + \epsilon = (\lambda_1 + \epsilon, \lambda_2 + \epsilon, \dots).$$

Using the definition of $m_{N,\overrightarrow{\lambda}}$ in Equation (3.13), $m_{N,\overrightarrow{\lambda}+\epsilon}$ is defined as

$$m_{N,\overrightarrow{\lambda}+\epsilon} = (\lambda_{N-1} + \epsilon)\, m_{N-1,\overrightarrow{\lambda}+\epsilon} + x_N,$$

and then the derivative of $m_{N,\overrightarrow{\lambda}}$ is defined in a "first principles" manner by

$$\Delta_{N,\overrightarrow{\lambda}} = \frac{\partial}{\partial \overrightarrow{\lambda}} m_{N,\overrightarrow{\lambda}} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left[ m_{N,\overrightarrow{\lambda}+\epsilon} - m_{N,\overrightarrow{\lambda}} \right].$$

In order to proceed further, the non-sequential form of $m_{N,\overrightarrow{\lambda}}$ is required. In Appendix A.3.1 it is shown that using Equation (3.13),

$$m_{N,\overrightarrow{\lambda}} = \sum_{i=1}^{N} \left[ \Big( \prod_{p=i}^{N-1} \lambda_p \Big) x_i \right].$$

The following result (Lemma 1 proved in Appendix A.3.3) is also needed:

$$\prod_{p=i}^{N} (\lambda_p + \epsilon) = \prod_{p=i}^{N} \lambda_p + \epsilon \left( \sum_{j=i}^{N} \Big( \prod_{\substack{p=i \\ p\neq j}}^{N} \lambda_p \Big) \right) + O(\epsilon^2).$$

The non-sequential equation for the derivative $\Delta_{N,\overrightarrow{\lambda}}$ can now be computed (with Lemma 1 used between lines (3.20) and (3.21)) by

$$\begin{aligned}
\Delta_{N,\overrightarrow{\lambda}} &= \lim_{\epsilon\to 0} \frac{1}{\epsilon} \left[ m_{N,\overrightarrow{\lambda}+\epsilon} - m_{N,\overrightarrow{\lambda}} \right] \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon} \left[ \sum_{i=1}^{N} \left[ \left( \prod_{p=i}^{N-1} (\lambda_p + \epsilon) \right) x_i \right] - \sum_{i=1}^{N} \left[ \left( \prod_{p=i}^{N-1} \lambda_p \right) x_i \right] \right] \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon} \sum_{i=1}^{N} \left[ \prod_{p=i}^{N-1} (\lambda_p + \epsilon) - \prod_{p=i}^{N-1} \lambda_p \right] x_i & (3.20) \\
&= \lim_{\epsilon\to 0} \frac{1}{\epsilon} \sum_{i=1}^{N} \left[ \epsilon \left( \sum_{j=i}^{N-1} \Big( \prod_{\substack{p=i \\ p\neq j}}^{N-1} \lambda_p \Big) \right) + O(\epsilon^2) \right] x_i & (3.21) \\
&= \lim_{\epsilon\to 0} \sum_{i=1}^{N} \left[ \frac{1}{\epsilon} \cdot \epsilon \left( \sum_{j=i}^{N-1} \Big( \prod_{\substack{p=i \\ p\neq j}}^{N-1} \lambda_p \Big) \right) + \frac{1}{\epsilon} \cdot O(\epsilon^2) \right] x_i \\
&= \lim_{\epsilon\to 0} \sum_{i=1}^{N} \left[ \sum_{j=i}^{N-1} \Big( \prod_{\substack{p=i \\ p\neq j}}^{N-1} \lambda_p \Big) + O(\epsilon) \right] x_i \\
\Rightarrow \Delta_{N,\overrightarrow{\lambda}} &= \sum_{i=1}^{N} \left[ \sum_{j=i}^{N-1} \Big( \prod_{\substack{p=i \\ p\neq j}}^{N-1} \lambda_p \Big) x_i \right]. & (3.22)
\end{aligned}$$

Next, a sequential definition of $\Delta_{N,\overrightarrow{\lambda}}$ can be computed (see Appendix A.3.4) from Equation (3.22) as,

$$\Delta_{N,\overrightarrow{\lambda}} = \lambda_{N-1} \Delta_{N-1,\overrightarrow{\lambda}} + m_{N-1,\overrightarrow{\lambda}}. \qquad (3.23)$$

Similarly, by defining

$$\Omega_{N,\overrightarrow{\lambda}} = \frac{\partial}{\partial \overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}} = \lim_{\epsilon \to 0} \frac{1}{\epsilon} \left[ w_{N,\overrightarrow{\lambda}+\epsilon} - w_{N,\overrightarrow{\lambda}} \right],$$

it can be shown (see Appendix A.3.5) that

$$\Omega_{N,\overrightarrow{\lambda}} = \lambda_{N-1} \Omega_{N-1,\overrightarrow{\lambda}} + w_{N-1,\overrightarrow{\lambda}}. \tag{3.24}$$

The derivative of $\bar{x}_{N,\overrightarrow{\lambda}}$ can now be **defined** by

$$\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda}} = \frac{\partial}{\partial \overrightarrow{\lambda}} \left( \frac{m_{N,\overrightarrow{\lambda}}}{w_{N,\overrightarrow{\lambda}}} \right) = \frac{\Delta_{N,\overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}} - m_{N,\overrightarrow{\lambda}} \Omega_{N,\overrightarrow{\lambda}}}{\left( w_{N,\overrightarrow{\lambda}} \right)^2}, \tag{3.25}$$

and can be sequentially updated using the definitions of $m_{N,\overrightarrow{\lambda}}$ and $w_{N,\overrightarrow{\lambda}}$ in Equations (3.13)-(3.14), the update equations for $\Delta_{N,\overrightarrow{\lambda}}$ and $\Omega_{N,\overrightarrow{\lambda}}$ in Equations (3.23) and (3.24), and the definition in Equation (3.25). It only remains to choose a cost function $L_{N+1,\overrightarrow{\lambda}}$ that would be desirable to minimise. Since the mean is being estimated, one natural choice is

$$L_{N+1,\overrightarrow{\lambda}} = \left[ \bar{x}_{N,\overrightarrow{\lambda}} - x_{N+1} \right]^2, \tag{3.26}$$

which has derivative

$$\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}} = 2 \left[ \bar{x}_{N,\overrightarrow{\lambda}} - x_{N+1} \right] \frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda}}. \tag{3.27}$$

Minimising this cost function attempts to ensure that the current AFF mean is as close as possible to the next observation. Of course, other cost functions could be used for the AFF mean. In Section 8.1, cost functions for the AFF variance are discussed. Indeed, the choice of the cost function will depend on the statistic that is being monitored (e.g. the mean or the variance).

It should be noted that a similar procedure was used to update an adaptive forgetting factor $\overrightarrow{\lambda}$ in [4], where the cost function was taken to be the negative log-likelihood. Interestingly, the procedure used there results in the same update equations for $\Delta_{N,\overrightarrow{\lambda}}$ and $\Omega_{N,\overrightarrow{\lambda}}$,

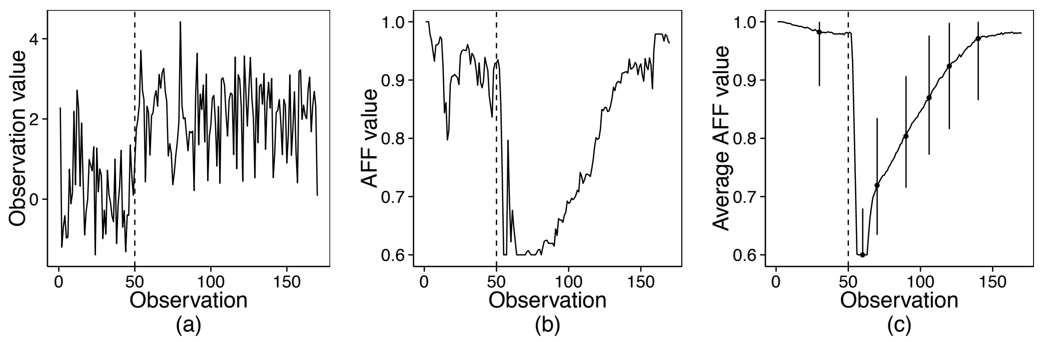

Figure 3.5: (a) A single stream generated by $X_1, \dots, X_{50} \sim \mathrm{N}(0,1)$ and $X_{51}, X_{52}, \dots \sim \mathrm{N}(2,1)$. (b) The behaviour of the AFF for the stream in (a). (c) Median value of $\overrightarrow{\lambda}$ (over 1000 streams) generated as in (a). The step-size used is $\eta = 0.01$. Error bars indicating the empirical 70% confidence interval are provided.

and in the case where it can be assumed that the stream is generated from normal random variables, it results in the same cost function as in Equation (3.26) being used. Note that the formulation here does not require any assumptions about the distribution of the stream, which is one of the key contributions of this thesis.

### 3.3.3 Behaviour of $\overrightarrow{\lambda}$ when encountering a change

With the derivations now completed, the behaviour of the AFF $\overrightarrow{\lambda}$ is explored. Figure 3.5(b) shows the behaviour of the AFF for a stream sampled from

$$\begin{aligned} X_1, X_2, \dots, X_{50} &\sim \mathrm{N}(0,1), \\ X_{51}, X_{52}, \cdots &\sim \mathrm{N}(2,1), \end{aligned}$$

where one realisation of stream is shown in Figure 3.5(a). Figure 3.5(b) shows the value of $\overrightarrow{\lambda}$ for this stream. Figure 3.5(c) shows the average behaviour of the AFF for such a stream, where the average is taken over 1000 such streams. It can be observed from these figures that soon after the change occurs, there is a large drop in $\overrightarrow{\lambda}$. This is desirable, as it allows the previous regime to be more quickly "forgotten". The optimal behaviour of a forgetting factor will be explored in Section 3.4. Note that the step-size used in Figures

3.5(b) and 3.5(c) is $\eta = 0.01$. The behaviour of $\overrightarrow{\lambda}$ for different values of $\eta$ will be explored in Section 3.3.6.

### 3.3.4 Plotting the average behaviour of $\overrightarrow{\lambda}$

Note that while Figure 3.2 shows the average (mean) value of $\bar{x}_{N,\lambda}$ with error bars that are one standard deviation of $\bar{x}_{N,\lambda}$, the average behaviour of $\overrightarrow{\lambda}$ is plotted differently. The quantity $\overrightarrow{\lambda}$ is likely to be asymmetric, and (as is described in Section 3.3.5) it is truncated to be in the interval $[0.6, 1]$. Therefore, to illustrate the average behaviour, Figure 3.5(c) shows the median value of $\overrightarrow{\lambda}$ with an empirical $70\%$ confidence interval. All subsequent plots displaying the average behaviour of $\overrightarrow{\lambda}$ will follow this procedure. A width of $70\%$ was chosen in order to try and correspond to the case of $\bar{x} \pm s$ for a normal random variable, since approximately $68\%$ of the values of a normal random variable are within one standard deviation of its mean.

### 3.3.5 Truncating the range of $\overrightarrow{\lambda}$

The value of the derivative in Equation (3.18) is not clearly bounded, and so updating $\overrightarrow{\lambda}$ with Equation (3.18) could allow $\overrightarrow{\lambda}$ to fall outside the interval $[0.1]$. However, Figures 3.5(b) and 3.5(c) show that the value of $\overrightarrow{\lambda}$ does not appear to drop below $0.6$. This is by design; after updating $\lambda_N \to \lambda_{N+1}$, then the algorithm implements the following rule

$$\lambda_{N+1} = \min(\lambda_{N+1}, \lambda_{\text{max}}), \tag{3.28}$$

$$\lambda_{N+1} = \max(\lambda_{N+1}, \lambda_{\text{min}}), \tag{3.29}$$

where $\lambda_{\text{min}}$ and $\lambda_{\text{max}}$ are the minimum- and maximum-allowed values for $\overrightarrow{\lambda}$.

Equations (3.28) and (3.29) ensure that $\lambda_N \in [\lambda_{\text{min}}, \lambda_{\text{max}}]$. It is clear that $\lambda_{\text{max}}$ should be set to $\lambda_{\text{max}} = 1$. One might think that $\lambda_{\text{min}} = 0.01$ is a good choice ("forget" almost everything). However, this leads to two problems. First, as Figure 3.6 shows, allowing $\overrightarrow{\lambda}$ to decrease so that it is close to 0 can have the effect of making recovery to pre-change values very slow. Figure 3.6(b) shows that with truncation at $0.01$, after 250 observations $\overrightarrow{\lambda}$ has still not recovered to pre-change levels of around 0.95. Second — and this is a less obvious

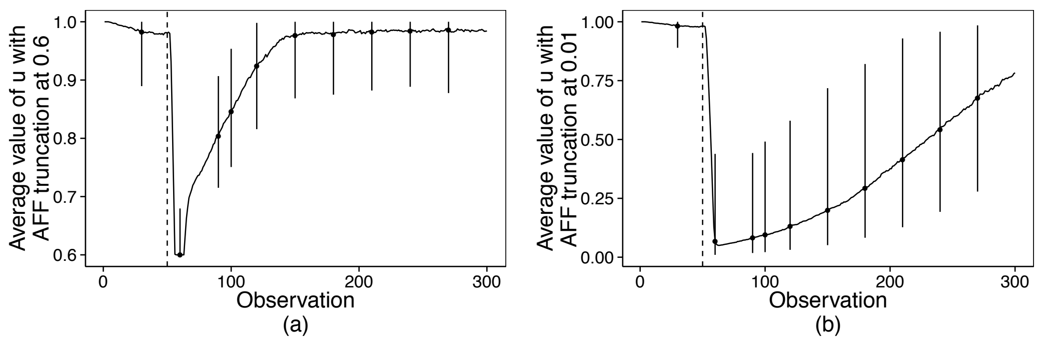

Figure 3.6: Median value of $\overrightarrow{\lambda}$ (over 1000 simulations) with (a) truncation at $0.6$ (b) truncation at $0.01$. The step-size is $\eta = 0.01$. Error bars indicating the empirical 70% confidence interval are provided.

point — if $\overrightarrow{\lambda}$ is too close to 0, the estimation of the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ is subject to a greater amount of variance. Recall from Equation (3.17) that the variance of $\bar{x}_{N,\overrightarrow{\lambda}}$ is controlled by the quantity $u_{N,\overrightarrow{\lambda}}$. Figure 3.7 shows the median values of $u_{N,\overrightarrow{\lambda}}$, corresponding to the median values of $\overrightarrow{\lambda}$ in Figure 3.6. It can be seen that when truncation at $\lambda_{\text{min}} = 0.01$ is used the value of $u_{N,\overrightarrow{\lambda}}$ increases dramatically after the changepoint. On the other hand, when truncation at $\lambda_{\text{min}} = 0.6$ is used, the increase in $u_{N,\overrightarrow{\lambda}}$ minor and short-lived. Furthermore, the error bars are much smaller for truncation at 0.6 than at 0.1. Other values besides $\lambda_{\text{min}} = 0.6$ could be used, but this value seems to offer a good balance between allowing as much of a drop in $\overrightarrow{\lambda}$ as possible, while avoiding the problems of recovery and increasing $u_{N,\overrightarrow{\lambda}}$. Figures 3.3 and 3.4 offer some justification for the choice of $0.6$: for $\lambda < 0.6$, values of $w_{N,\lambda}$ and $u_{N,\lambda}$ (for different values of $N$) appear to be indistinguishable. One might hope that increasing the step-size from $\eta = 0.01$ to $\eta = 0.1$ may may fix these problems for $\lambda_{\text{min}} = 0$. While it is true that recovery is improved for $\eta = 0.1$, there is still the problem of an increase in $u_{N,\overrightarrow{\lambda}}$. The next section discusses the choice of $\eta$ further.

### 3.3.6 Choice of step size $\eta$

It may appear that the choice of forgetting factor $\lambda$ has been replaced with a choice of step size $\eta$, where any value in the range $[0.001, 0.1]$ would seem reasonable. It would be expected that the value of $\eta$ would affect the behaviour of $\overrightarrow{\lambda}$, as shown in Figure 3.8 which,

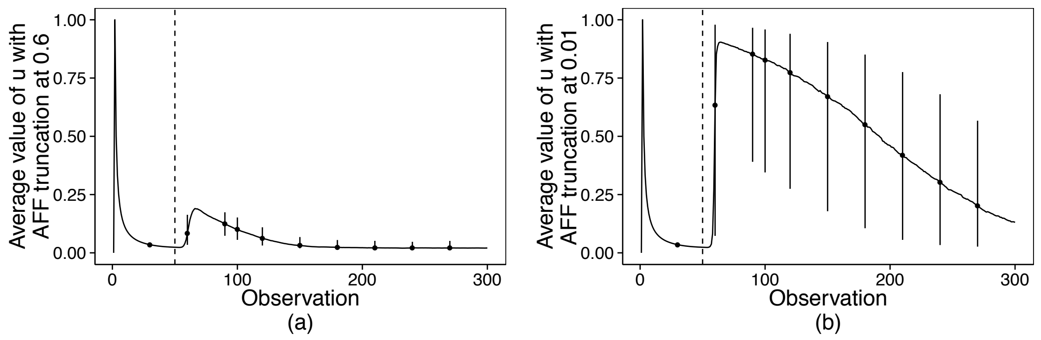

Figure 3.7: Median value of $u_{N,\overrightarrow{\lambda}}$ (over 1000 simulations), with (a) truncation of $\overrightarrow{\lambda}$ at 0.6 (b) truncation of $\overrightarrow{\lambda}$ at 0.01 The step-size is $\eta = 0.01$. Error bars indicating the empirical 70% confidence interval are provided.

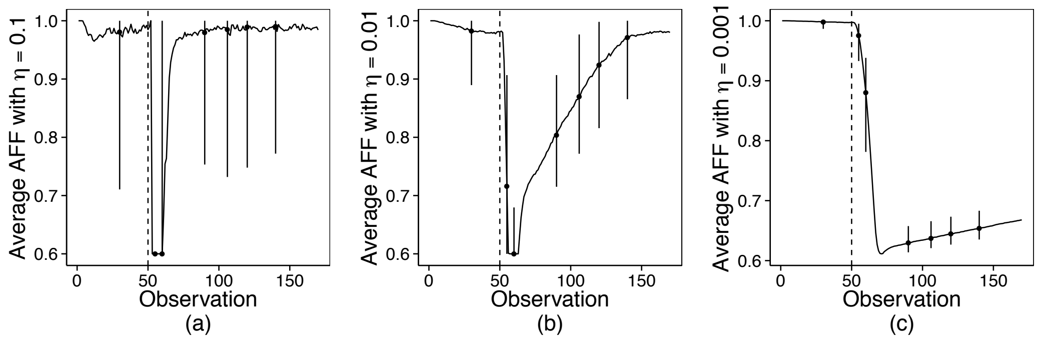

Figure 3.8: Median value of AFF $\overrightarrow{\lambda}$ (over 1000 simulations) for $\eta = 0.1$ (a), $\eta = 0.01$ (b) and $\eta = 0.001$ (c), for stream generated by $X_1, \ldots, X_{50} \sim \mathrm{N}(0, 1)$ and $X_{51}, X_{52}, \cdots \sim \mathrm{N}(2, 1)$. Error bars indicating the empirical 70% confidence interval are provided.

as in Figure 3.2 shows the average behaviour. However, Figure 3.9 appears to suggest that the choice of $\eta$ does not seem to affect the behaviour of the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ as much as the choice of $\lambda$. Furthermore, it is shown in Appendix A.3.9 that for $L_{N,\overrightarrow{\lambda}}$ defined in Equation (3.26),

$$\mathrm{E}\left[\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N,\overrightarrow{\lambda}}\right] \sim O(\sigma^2), \tag{3.30}$$

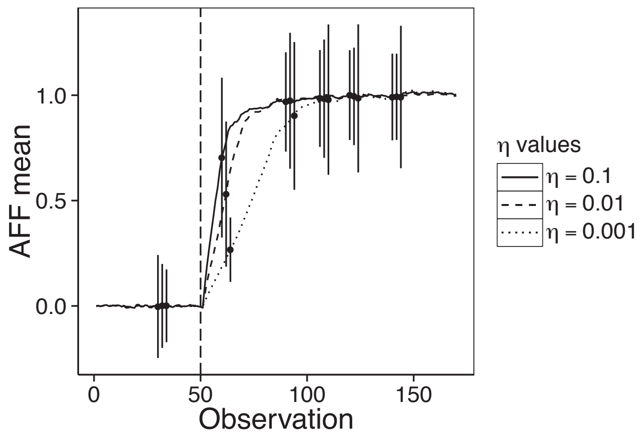

Figure 3.9: Average behaviour of AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ (over 1000 simulations) for $\eta = 0.1, 0.01, 0.001$ for stream generated by $X_1, \ldots, X_{50} \sim \mathrm{N}(0,1)$ and $X_{51}, X_{52}, \cdots \sim \mathrm{N}(1,1)$. Error bars with a width of one standard deviation are provided.

and if $\sigma^2$ is known or an estimate is obtained, then the derivative should be scaled by $\sigma^2$, i.e. the update equation, Equation (3.18), should be

$$\lambda_{N+1} = \lambda_N - \frac{\eta}{\sigma^2}\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}}. \tag{3.31}$$

From now on, we shall assume that the derivative is always scaled by a (possibly estimated) value of $\sigma^2$, or our update equation is Equation (3.31).

With this scaling, there will be little dependence on the choice of $\eta$ (at least for the range $[0.001, 0.1]$); in Section 5.4 it is shown that the AFF mean change detection scheme performs very similarly for different choices of $\eta$, which provides some evidence that the choice of $\eta$ is not crucial in continuous monitoring.

## 3.3.7 Comparison of the AFF mean and the FFF mean

Figure 3.10 compares the average behaviours of the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ and the FFF mean $\bar{x}_{N,\lambda}$ as in Figure 3.2. Figure 3.10 shows that, at least for this example where there is a change from $\mu_0 = 0$ to $\mu_1 = 3$, the average $\bar{x}_{N,\overrightarrow{\lambda}}$ reacts to the change faster than any of the fixed FFF schemes, yet also exhibits stability when the stream is in control. This

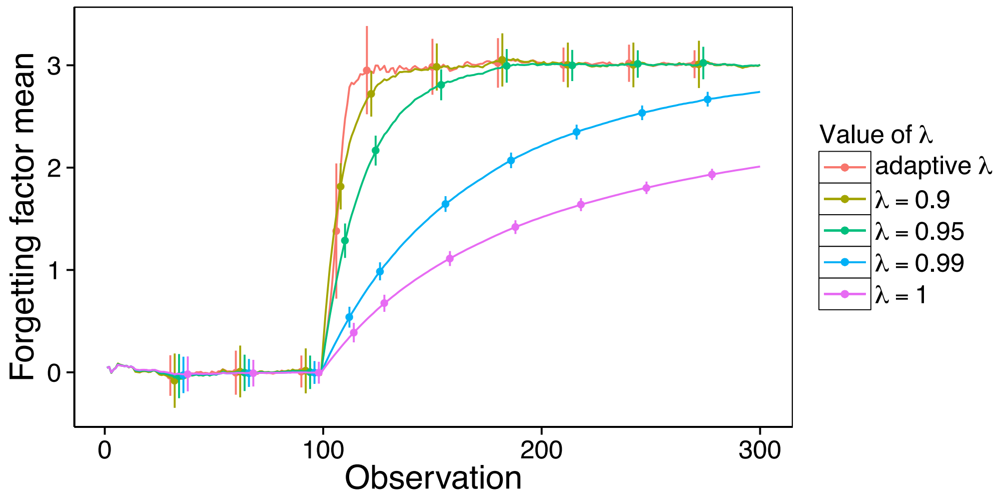

Figure 3.10: A comparison between $\bar{x}_{N,\lambda}$ for different values of $\lambda$ and $\bar{x}_{N,\overrightarrow{\lambda}}$, for $X_1, \dots, X_{100} \sim \mathrm{N}(0,1)$, $X_{101}, \dots, X_{300} \sim \mathrm{N}(3,1)$, averaged over 100 simulations. The step size used in this figure is $\eta = 0.1$, but other values yield very similar results. Error bars with a width of one standard deviation are provided.

combination of the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ being very responsive to a change, yet also being stable during periods of stationarity (Figure 3.8 shows $\overrightarrow{\lambda}$ increasing back to pre-change levels after a change), are ideal characteristics for the estimator of a time-varying quantity.

### 3.3.8 Relation to Kalman Filter

Suppose a random walk is characterised for $i = 1, 2, \dots$ by

$$X_i \sim \mathrm{N}(\mu_i, \sigma_X^2),$$
$$\mu_i = \mu_{i-1} + \xi_i,$$
$$\xi_i \sim \mathrm{N}(0, \sigma_\xi^2).$$

for some parameters $\sigma_X$ and $\sigma_\xi$. In [3, Sec. 3.1.6] it is remarked that for such a random walk, if the parameters $\sigma_X$ and $\sigma_\xi$ are known and the optimal filter estimate after observation $N$, given all observations $x_1, x_2, \ldots, x_N$, is denoted by

$$\widehat{\mu}_N^{KF} = \mathrm{E}\left[X_N | x_1, \ldots, x_N; \sigma_X, \sigma_\xi\right],$$

then $\widehat{\mu}_N^{KF}$ is recursively computable by the Kalman Filter equations [83]. Furthermore, it is shown in [3, Sec. 3.1.6] that (for this special case)

$$\widehat{\mu}_N^{KF} = \left(1 - \frac{1}{w_N^{KF}}\right)\widehat{\mu}_{N-1}^{KF} + \left(\frac{1}{w_N^{KF}}\right), \qquad \widehat{\mu}_0^{KF} = 0 \tag{3.32}$$

$$w_{N+1}^{KF} = \lambda_N w_N^{KF} + 1, \qquad w_0^{KF} = 0 \tag{3.33}$$

$$\lambda_N = \left[\frac{\sigma_\xi^2}{\sigma_X^2} w_N^{KF} + 1\right]^{-1} \tag{3.34}$$

It is interesting to compare Equations (3.32) and (3.32) with Equations (3.14) and (3.15) for the AFF mean. It shows that the optimal filter equations are of the same form as the AFF mean equations, although the method for setting the forgetting factor $\lambda_N$ in Equation (3.34) is different. In this simple context, this argument provides, to some extent, a theoretical interpretation of the forgetting factor.

## 3.4 Optimal forgetting factors $\lambda$ and $\overrightarrow{\lambda}$

It is natural to wonder if there is an optimal value for a fixed forgetting factor $\lambda$ or for an adaptive forgetting factor $\overrightarrow{\lambda}$. In order to proceed with this question, we first need to define what we might mean by "optimal", and since we are ultimately concerned with change detection, it is natural to consider the problem of finding an optimal $\lambda$ with respect to a single change-point in the data. Assume, as before, that we have a sequence of at least $N$ observations $x_1, x_2, \ldots$ generated from i. i. d. random variables $X_1, X_2, \ldots$, and suppose that

$$X_1, X_2, \ldots, X_\tau \sim \mathrm{N}(\mu_0, \sigma_0^2), \tag{3.35}$$

$$X_{\tau+1}, X_{\tau+2}, \ldots \sim \mathrm{N}(\mu_1, \sigma_1^2), \tag{3.36}$$

where $\tau$ is the changepoint and $\mu_0 \neq \mu_1$. For $N > \tau$, define $D = N - \tau$.

### 3.4.1 Definition of optimal $\lambda$

Given the data stream in Equations (3.35) and (3.36), the optimal fixed $\lambda$ is defined to be

$$\widehat{\lambda}_{\tau,N} = \begin{cases} \arg\big\{\inf_{\lambda\in[0,1]} \mathrm{E}\big[(\bar{X}_{N,\lambda}-\mu_0)^2\big]|A_0\big\} & \text{for } N \le \tau, \\ \arg\big\{\inf_{\lambda\in[0,1]} \mathrm{E}\big[(\bar{X}_{N,\lambda}-\mu_1)^2\big]|A_1\big\} & \text{for } N > \tau, \end{cases} \tag{3.37}$$

$$A_0 = X_1,\dots,X_N \sim \mathrm{N}(\mu_0,\sigma_0^2),$$

$$A_1 = X_1,\dots,X_\tau \sim \mathrm{N}(\mu_0,\sigma_0^2), X_{\tau+1},\dots,X_N \sim \mathrm{N}(\mu_1,\sigma_1^2).$$

If more than one $\lambda$ gives the same infimum, the larger $\lambda$ value is used. For $N = 1, 2, \dots$ the optimal fixed forgetting factor vector

$$\widehat{\lambda}_\tau = \big(\widehat{\lambda}_{\tau,1}, \widehat{\lambda}_{\tau,2}, \dots, \widehat{\lambda}_{\tau,N}, \dots\big) \tag{3.38}$$

is obtained. It will be convenient to use the shorthand

$$\mathrm{E}\big[(\bar{X}_{N,\lambda}-\mu_{0,1,N})^2|A_{0,1}\big],$$

where we define $\mu_{0,1,N}$ as

$$\mu_{0,1,N} = \begin{cases} \mu_0 & \text{for } N \le \tau, \\ \mu_1 & \text{for } N > \tau, \end{cases} \tag{3.39}$$

and $A_{0,1}$ as

$$A_{0,1} = \begin{cases} A_0 & \text{for } N \le \tau, \\ A_1 & \text{for } N > \tau, \end{cases} \tag{3.40}$$

where $A_0$ and $A_1$ are defined in Equation (3.37). It is shown in Appendix A.4 that for $N > \tau$,

$$\begin{aligned} \mathrm{E}\big[(\bar{X}_{N,\lambda}-\mu_1)^2|A_{0,1}\big] = {} & \left[\left(\frac{1}{w_{N,\lambda}}\right)\left(\lambda^D w_{\tau,\lambda}\mu_0 + w_{D,\lambda}\mu_1\right) - \mu_1\right]^2 + \\ & + \left(\frac{1}{w_{N,\lambda}}\right)^2\left[\lambda^{2D} w_{\tau,\lambda^2}\sigma_0^2 + w_{D,\lambda^2}\sigma_1^2\right], \end{aligned} \tag{3.41}$$

while for $N \leq \tau$,

$$\mathrm{E}\big[(\bar{X}_{N,\lambda} - \mu_1)^2 | A_{0,1}\big] = (u_{N,\lambda})\sigma_0^2.$$

This provides us with sufficient information in order to solve for each $\widehat{\lambda}_{\tau,N}$.

### 3.4.2 Solving for optimal fixed $\lambda$

In order to find

$$\widehat{\lambda}_{\tau,N} = \arg\big\{ \inf_{\lambda\in[0,1]} \mathrm{E}\big[(\bar{X}_{N,\lambda} - \mu_{0,1,N})^2 | A_{0,1}\big]\big\}, \tag{3.42}$$

Equation (3.41) can be numerically evaluated for $\lambda \in \{0, \delta, 2\delta, \ldots, (L-1)\delta, L\delta = 1\}$, where $\delta = 1/L$ and $L$ is a large number (e.g. $L = 1000$). Note that while one could try to find an analytical solution for Equation (3.42) by trying to solve

$$\frac{\partial}{\partial\lambda}\mathrm{E}\big[(\bar{X}_{N,\lambda} - \mu_{0,1,N})^2 | A_{0,1}\big] = 0, \tag{3.43}$$

this approach seems unlikely to succeed, since the derivative of the expression in Equation (3.41) is unlikely to have an analytical solution in terms of $\lambda$. However, we are justified in using an iterative (numerical) method to solve Equation (3.41) here, since we are exploring theoretical optimal values. In practice, an iterative method would be unsuitable in a streaming data context (see Chapter 1).

### 3.4.3 Results for optimal fixed $\lambda$: Example

The development above provides the general framework for reasoning about optimal fixed $\lambda$. To illustrate this using a specific example, suppose consider the stream generated by

$$X_1, X_2, \ldots, X_{50} \sim \mathrm{N}(0,1), \tag{3.44}$$

$$X_{51}, \ldots, X_{100}, \cdots \sim \mathrm{N}(2,1). \tag{3.45}$$

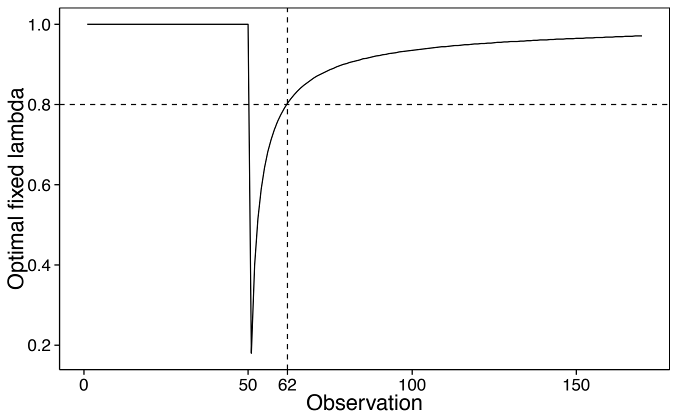

Figure 3.11: A plot of the optimal forgetting factor $\widehat{\lambda}_{50}$ for the random variables given in Equations (3.44) and (3.45). A vertical line at observation $62$ intersects with a horizontal line at optimal fixed $\lambda$ value $0.8$. This means that, for a changepoint at $\tau = 50$ from $\mathrm{N}(0,1)$ to $\mathrm{N}(2,1)$, the FFF mean that minimises Equation (3.37) after 62 observations is $\bar{x}_{N,0.8}$.

If $\widehat{\lambda}_{50,N}$ is computed for each $N = 1, 2, \ldots, 100, \ldots$, the vector

$$\widehat{\lambda}_{50} = \left(\widehat{\lambda}_{50,1}, \widehat{\lambda}_{50,2}, \ldots, \widehat{\lambda}_{50,100}, \ldots\right),$$

is obtained. Figure 3.11 is a plot of $\widehat{\lambda}_{50}$. Notice that $\widehat{\lambda}_{50,1}, \ldots, \widehat{\lambda}_{50,50} = 1$, indicating that the optimal forgetting factor to use in this range is 1, which will give equal weight to all the random variables up until the changepoint at 50. Figure 3.11 shows that after the changepoint at 50, the optimal forgetting factor drops almost to 0, before increasing back towards 0.99. So, for example, in this simulation the optimal forgetting factor that will give the smallest residual for the first 62 observations, $x_1, \ldots, x_{62}$, is approximately 0.8.

### 3.4.4 Optimal adaptive $\overrightarrow{\lambda}$

Analogously to Section 3.4.1, define the optimal adaptive forgetting factor vector $\widehat{\overrightarrow{\lambda}}_{\tau,N}$, where

$$\widehat{\overrightarrow{\lambda}}_{\tau,N} = \left(\lambda_{\tau,1}, \lambda_{\tau,2}, \ldots, \lambda_{\tau,N}\right)$$

consists of $\lambda_{\tau,i}$ unrelated to the optimal fixed $\lambda$'s of Section 3.4.1, and

$$\widehat{\overrightarrow{\lambda}}_{\tau,N} = \begin{cases} \arg\left\{ \inf_{\lambda_{\tau,N}\in[0,1]} \mathrm{E}\big[(\bar{X}_{N,\widehat{\overrightarrow{\lambda}}_{\tau,N}} - \mu_0)^2\big]\right\} & \text{for } N \leq \tau, \\ \arg\left\{ \inf_{\lambda_{\tau,N}\in[0,1]} \mathrm{E}\big[(\bar{X}_{N,\widehat{\overrightarrow{\lambda}}_{\tau,N}} - \mu_1)^2\big]\right\} & \text{for } N > \tau, \end{cases} \tag{3.46}$$

$$A_0 = x_1, \ldots, x_N \sim \mathrm{N}(\mu_0, \sigma_0^2),$$
$$A_1 = x_1, \ldots, x_\tau \sim \mathrm{N}(\mu_0, \sigma_0^2), x_{\tau+1}, \ldots, x_N \sim \mathrm{N}(\mu_1, \sigma_1^2).$$

Note that in order to define $\widehat{\overrightarrow{\lambda}}_{\tau,N}$, the information $\widehat{\overrightarrow{\lambda}}_{\tau,N-1} = \big(\lambda_{\tau,1}, \lambda_{\tau,2}, \ldots, \lambda_{\tau,N-1}\big)$ is needed. It will be convenient to use shorthand

$$\mathrm{E}\big[(\bar{X}_{N,\overrightarrow{\lambda}} - \mu_{0,1,N})^2 | A_{0,1}\big]$$

to refer to the expectation in Equation (3.46). For the rest of this section, in order to increase readability, define

$$\overrightarrow{\lambda} = \widehat{\overrightarrow{\lambda}}_{\tau,N}.$$

In other words, for the rest of this section $\overrightarrow{\lambda}$ denotes the *optimal adaptive forgetting factor*. In order to calculate $\overrightarrow{\lambda}$, recall that $D = N - \tau$ and define

$$P^{\tau+D-1}_{\tau,\overrightarrow{\lambda}} = \prod_{p=\tau}^{\tau+D-1} \lambda_{\tau,p}. \tag{3.47}$$

Following the same procedure shown in Appendix A.4 (the calculation of optimal fixed $\lambda$), it can then be shown that

$$\begin{aligned}\mathrm{E}\big[(\bar{X}_{N,\overrightarrow{\lambda}} - \mu_{0,1,N})^2 | A_{0,1}\big] = &\left[\left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)\left(P^{\tau+D-1}_{\tau,\overrightarrow{\lambda}} w_{\tau,\overrightarrow{\lambda}}\mu_0 + w_{D,\overrightarrow{\nu}}\mu_1\right) - \mu_1\right]^2 + \\ &+ \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \left[P^{\tau+D-1}_{\tau,\overrightarrow{\lambda}^2} w_{\tau,\overrightarrow{\lambda}^2}\sigma_0^2 + w_{D,\overrightarrow{\nu}^2}\sigma_1^2\right].\end{aligned} \tag{3.48}$$

Now that an expression for the cost function used in Equation (3.46) has been found (note the similarity to the solution for fixed case given in Equation 3.41), the optimal $\widehat{\overrightarrow{\lambda}}_{N,\tau}$ can be solved numerically, as in Section 3.4.3. Furthermore, it can also be shown that

$$\mathrm{E}\big[\bar{X}_{N,\overrightarrow{\lambda}}\big] = \frac{1}{w_{N,\overrightarrow{\lambda}}}\Big[P^{\tau+D-1}_{\tau,\overrightarrow{\lambda}}(w_{\tau,\overrightarrow{\lambda}})\mu_0 + (w_{D,\overrightarrow{\nu}})\mu_1\Big] \tag{3.49}$$

$$\mathrm{Var}\big[\bar{X}_{N,\overrightarrow{\lambda}}\big] = \frac{1}{(w_{N,\overrightarrow{\lambda}})^2}\Big[\big(P^{\tau+D-1}_{\tau,\overrightarrow{\lambda}}\big)^2(w_{\tau,\overrightarrow{\lambda}^2})\sigma_0^2 + (w_{D,\overrightarrow{\nu}^2})\sigma_1^2\Big]. \tag{3.50}$$

Note the similarity to the fixed optimal $\lambda$ equations. However, these two sets of equations lead to very different results. For the fixed case, it is shown in the Appendix A.4 (Equations (A.46) and (A.48)) that

$$\mathrm{E}\big[\bar{X}_{N,\lambda}\big] = \frac{1}{w_{N,\lambda}}\Big(\lambda^D(w_{\tau,\lambda})\mu_0 + (w_{D,\lambda})\mu_1\Big),$$
$$\mathrm{Var}\big[\bar{X}_{N,\lambda}\big] = \frac{1}{(w_{N,\lambda})^2}\Big(\lambda^{2D}(w_{\tau,\lambda^2})\sigma_0^2 + (w_{D,\lambda^2})\sigma_1^2\Big).$$

### 3.4.5 Results for optimal $\overrightarrow{\lambda}$: Example

As in Section 3.4.3, suppose the stream $x_1, x_2, \dots$ is generated by the random variables

$$X_1, X_2, \dots, X_{50} \sim \mathrm{N}(0,1) \tag{3.51}$$

$$X_{51}, \dots, X_{100}, \dots \sim \mathrm{N}(5,1) \tag{3.52}$$

Again, the vector $\widehat{\overrightarrow{\lambda}}_{\tau,N}$

$$\widehat{\overrightarrow{\lambda}}_{50} = \big(\widehat{\overrightarrow{\lambda}}_{50,1}, \widehat{\overrightarrow{\lambda}}_{50,2}, \dots, \widehat{\overrightarrow{\lambda}}_{50,100}\big)$$

can be solved numerically and the optimal vector is shown in Figure 3.12. When compared to the fixed case in Figure 3.11, Figure 3.12 suggests that adaptive forgetting can respond to changes faster and recover quicker than fixed forgetting, in principle. The two schemes are compared more closely in the next section.

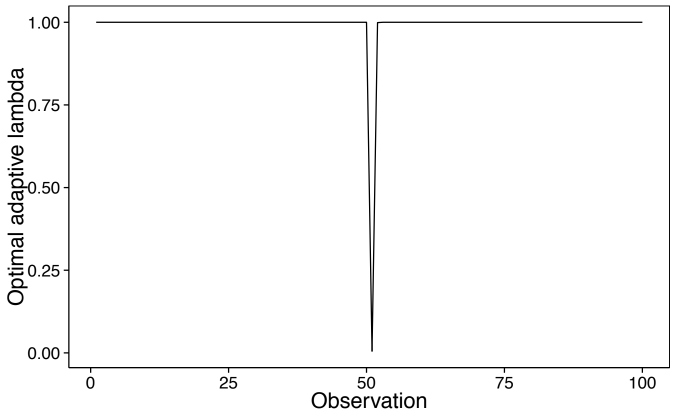

Figure 3.12: A plot showing the behaviour of the optimal adaptive forgetting factor $\widehat{\overrightarrow{\lambda}}$ for the random variables given in Equations (3.51) and (3.52).

### 3.4.6 Comparison of $\widehat{\lambda}_{51,N}$ and $\widehat{\overrightarrow{\lambda}}_{51,N}$

We compare the values of $\widehat{\Lambda}_{\tau,N}$, where $\Lambda = \lambda$ is either the optimal fixed forgetting factor from Sections 3.4.1-3.4.3 or the optimal adaptive forgetting factor $\overrightarrow{\lambda}$ from Section 3.4.4. We do this by looking at two particular examples, where in both cases the observations are

$$\begin{aligned} X_1, X_2, \ldots, X_{50} &\sim \mathrm{N}(0,1) \\ X_{51}, \ldots &\sim \mathrm{N}(\mu_1, 1) \end{aligned}$$

where $\mu_1 = 2$ or $\mu_1 = 5$ and the changepoint is $\tau = 50$. The tables compare (where $\Lambda$ is either $\lambda$ or $\overrightarrow{\lambda}$):

1. $\widehat{\Lambda}_{50,1}, \ldots, \widehat{\Lambda}_{50,50}$, the optimal values of $\Lambda$ before the change-point,
2. $\widehat{\Lambda}_{51,1}$, the optimal value of $\Lambda$ immediately after the change-point,
3. the effective sample size immediately after the change-point $w_{51,\Lambda}$,
4. the expectation of $\bar{X}_{51,\Lambda}$, and
5. the variance of $\bar{X}_{51,\Lambda}$, where the optimal value of the forgetting factor has been used for each observation.

By numerically solving Equations (3.41) and (3.48), we obtain

| | $\mu_2$ | $\widehat{\Lambda}_{50,1}, \ldots, \widehat{\Lambda}_{50,50}$ | $\widehat{\Lambda}_{50,51}$ | $w_{51,\Lambda}$ | $\text{E}[\bar{X}_{51,\Lambda}]$ | $\text{Var}[\bar{X}_{51,\Lambda}]$ |
|---|---|---|---|---|---|---|
| $\Lambda = \lambda$ | 2 | 0.180 | 0.180 | 1.220 | 1.640 | 0.825 |
| $\Lambda = \overrightarrow{\lambda}$ | 2 | 1.000 | 0.005 | 1.250 | 1.600 | 0.801 |
| $\Lambda = \lambda$ | 5 | 0.037 | 0.037 | 1.038 | 4.815 | 0.963 |
| $\Lambda = \overrightarrow{\lambda}$ | 5 | 1.000 | 0.0008 | 1.040 | 4.808 | 0.962 |

Table 3.1: Comparison of optimal fixed $\lambda$ and adaptive $\overrightarrow{\lambda}$ immediately after a change at $\tau = 50$.

First, we notice that $\widehat{\Lambda}_{50,1}, \ldots, \widehat{\Lambda}_{50,50} = 1$ for both $\Lambda = \lambda$ and $\Lambda = \overrightarrow{\lambda}$, as one might expect, since there is no change-point. We then see that $\widehat{\Lambda}_{50,51}$ is close to zero in both experiments, for both pairs, with $\widehat{\overrightarrow{\lambda}}_{50,51}$ being closer to zero than $\widehat{\lambda}_{50,51}$.

However, it is interesting that when we compare the pairs of $w_{51,\Lambda}$, $\text{E}[\bar{X}_{51,\Lambda}]$ and $\text{Var}[\bar{X}_{51,\Lambda}]$, we see all the pairs have similar values. It seems as if there are optimal values for these quantities, regardless of whether the forgetting factor is fixed or adaptive.

It is worth highlighting, again, the difference between the two approaches: when $\mu_2 = 5$, $\widehat{\lambda}_{50,51} = 0.037$ means that $\text{E}\big[(\bar{X}_{51,\lambda} - \mu_{0,1,N})^2\big]$ will be minimised if the fixed forgetting factor of $\lambda = 0.037$ is used for observations $x_1, \ldots x_{50}, x_{51}$. In the adaptive case, though, $\big[(\bar{X}_{51,\lambda} - \mu_{0,1,N})^2\big]$ will be minimised when the forgetting factors are

$$(\lambda_1, \ldots, \lambda_{50}, \lambda_{51}) = (1, \ldots, 1, 0.005)$$

for observations $x_1, \ldots x_{50}, x_{51}$, when $\mu_1 = 2$. Finally, as mentioned in the last section, comparing Figures 3.11 and 3.12 suggests that adaptive forgetting can respond to changes faster and recover quicker than fixed forgetting.

## 3.5 Discussion

In this chapter we introduced the forgetting factor framework, starting with the *fixed* forgetting factor scheme. Next we introduced the idea of an adaptive forgetting factor $\overrightarrow{\lambda}$, and defined the *adaptive* forgetting factor mean $\bar{x}_{N,\overrightarrow{\lambda}}$. A method for updating $\overrightarrow{\lambda}$ was proposed, which needed the notion of a derivative with respect to $\overrightarrow{\lambda}$. All quantities of interest were shown to have recursive update equations, making this framework suitable for sequential change detection.

The idea of an optimal adaptive forgetting factor vector was also proposed, following on from the fixed case described in Sections 3.4.1-3.4.3. Finally, a comparison between the optimal fixed and optimal adaptive forgetting factors was then provided in Section 3.4.6, which explored the relationship between the two schemes. The performance of $\overrightarrow{\lambda}$ seems be behave similarly to the optimal $\lambda$.

The next chapter embeds this chapter's adaptive estimation methodology into a change detection framework.

# Chapter 4

# Change detection using adaptive estimation

The previous chapter developed various adaptive estimation procedures for the mean of a data stream. This chapter extends that development by embedding the estimation schemes in a change detector. This is achieved by the imposition of a decision rule (see Section 2.1.2). Several novel schemes are considered, including: assuming the stream is normally-distributed, distribution-free schemes based on probabilistic bounds, a scheme combining adaptive and fixed forgetting and a change detector based on monitoring the value of the adaptive forgetting factor $\overrightarrow{\lambda}$ itself. Various issues arriving with these new detectors are discussed, and crude experimental comparisons are conducted. Recalling that the final objective of this work is continuous monitoring, these comparisons are intended to fix ideas rather than seek optimal solutions. A concern throughout these comparisons is the role of control parameters and their impact on change detection performance measures.

Section 4.1 develops the embedding of adaptive estimation into a change detection framework. Additionally, the relationship between FFF and EWMA is illuminated, and the role of the step-size $\eta$ is explored. Section 4.2 introduces a variety of distribution-free change detection approaches with forgetting factors. Finally, Section 4.3 provides experimental results comparing these new algorithms with one another, and additionally, with standard approaches.

| Label | Description |
|---|---|
| EWMA | Exponentially weighted moving average scheme (see Section 2.1.5) |
| CUSUM | Cumulative sum scheme (see Section 2.1.4) |
| FFF | Fixed forgetting factor scheme |
| AFF | Adaptive forgetting factor scheme |
| AFFcheby | modification of AFF using Chebyshev's method |
| AFFdrop | modification of AFF; significant decrease in $\overrightarrow{\lambda}$ signals a change |
| F-AFF | modification of AFF; uses FFF mean to create control limits |
| ARL0 | Average number of observations between false alarms |
| SDRL0 | Standard deviation of ARL0 |
| ARL1 | Average delay in detecting a true changepoint |
| SDRL1 | Standard deviation of ARL1 |

Table 4.1: Explanation of labels used for different change detection schemes

## 4.1 Change detection using forgetting factors

Suppose the univariate stream $x_1, x_2, \dots$ is generated by i.i.d. random variables $X_1, X_2, \dots$ with

$$\mathrm{E}\left[X_i\right] = \mu, \qquad \mathrm{Var}\left[X_i\right] = \sigma^2, \qquad i = 1, 2, \dots. \tag{4.1}$$

Recall the definition of the AFF mean $\bar{X}_{N,\overrightarrow{\lambda}}$ from Section 3.3.1 where it is shown that for such a stream

$$\mathrm{E}\left[\bar{X}_{N,\overrightarrow{\lambda}}\right] = \mu, \qquad \mathrm{Var}\left[\bar{X}_{N,\overrightarrow{\lambda}}\right] = (u_{N,\overrightarrow{\lambda}})\sigma^2, \tag{4.2}$$

where the definition of $u_{N,\overrightarrow{\lambda}}$ is given in Section 3.3.1 and its derivation is given in Appendix A.3.7. From this starting point, several change detection schemes can be defined. Table 4.1 provides a reference for the abbreviations of the methods used in the subsequent tables.

### 4.1.1 Normal streams

Following the tradition of most statistical process control literature, suppose that the random variables $X_1, X_2, \dots$ are i.i.d. normal,

$$X_1, X_2, \dots \sim \mathrm{N}(\mu, \sigma^2). \tag{4.3}$$

It then immediately follows from Equation (4.2) that $\bar{X}_{N,\overrightarrow{\lambda}}$ is also normally distributed,

$$\bar{X}_{N,\overrightarrow{\lambda}} \sim \mathrm{N}\left(\mu, (u_{N,\overrightarrow{\lambda}})\sigma^2\right).$$

Denoting the cumulative distribution function (cdf) of a $\mathrm{N}(\mu, \sigma^2)$ distribution by $F_{\mathrm{N}(\mu,\sigma^2)}$, the quantity

$$p = F_{\mathrm{N}\left(\mu,(u_{N,\overrightarrow{\lambda}})\sigma^2\right)}\left(\bar{x}_{N,\overrightarrow{\lambda}}\right) \in [0,1], \tag{4.4}$$

provides a measure for how well a particular value $\bar{x}_{N,\overrightarrow{\lambda}}$ follows a $\mathrm{N}\left(\mu, (u_{N,\overrightarrow{\lambda}})\sigma^2\right)$ distribution, assuming that the stream is in-control and that all the observations are generated according to Equation (4.3). For a significance level $\alpha$, a $100(1-\alpha)\%$-**prediction interval** could be given by

$$p \in \left(\frac{\alpha}{2}, 1 - \frac{\alpha}{2}\right).$$

Alternatively, $p$ could be rescaled to be one-sided via

$$p' = 1 - |1 - 2p|,$$

and a decision rule for signalling a change is then given by

$$p' < \alpha. \tag{4.5}$$

We call this the AFF change detection scheme. Note that we could consider $p'$ to be a $p$-value, since it is a measure of how well $\bar{x}_{N,\overrightarrow{\lambda}}$ follows a $\mathrm{N}(\mu, (u_{N,\overrightarrow{\lambda}})\sigma^2)$ distribution. This terminology will be revisited in the multivariate setting in Chapter 6.

Although only the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ has been discussed in this section, the same decision rule is derived for the FFF mean $\bar{x}_{N,\lambda}$, by simply considering the case when the AFF $\overrightarrow{\lambda} = (\lambda, \lambda, \dots)$. This is called the FFF change detection scheme. In Section 4.1.2 the FFF and EWMA schemes are shown to be closely-related, and their change detection performance is compared.

### 4.1.2 Comparison between EWMA and FFF

In Section 3.2.5 the relationship between the FFF and EWMA schemes was discussed. In short, the FFF scheme tends to the EWMA scheme as $N$, the number of observations, tends to infinity. It therefore seems as if the FFF change detector should perform similarly to the EWMA change detector, although one might expect the FFF scheme to be better at detecting changepoints that occur near the start of the monitoring process. This section provides a brief experimental analysis comparing the FFF and EWMA schemes.

The experiments will involve repeated Monte Carlo trials for each method attempting to detect a single changepoint. Several locations of $\tau$ will be considered,

$$\tau \in \{30, 50, 100, 200, 500\},$$

to test performance for short-, medium- and long-term changes. In this section, and throughout this chapter, the analysis of the change detection methods will follow the standard procedure of assuming that the pre-change mean and variance of the stream are *known*. However, in later chapters these values will be estimated during a burn-in period; this approach will be crucial when considering multiple changepoints in a stream (Chapter 5).

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| EWMA | $(r, L)$ | (0.25, 3.00) | 442.86 | (461.07) | 11.09 | (7.31) |
| EWMA | $(r, L)$ | (0.01, 3.00) | 1210.90 | (883.05) | 20.02 | (7.84) |
| EWMA | $(r, L)$ | (0.25, 2.00) | 28.45 | (33.51) | 5.44 | (3.77) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.01) | 405.43 | (403.31) | 10.89 | (4.93) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.05) | 92.29 | (108.01) | 7.92 | (3.88) |
| FFF | $(\lambda, \alpha)$ | (0.99, 0.01) | 1067.49 | (761.43) | 19.65 | (7.70) |

Table 4.2: An ARL table for EWMA and FFF over 1000 trials for $\tau = 100$. For ARL1, $X_1, \ldots X_{100} \sim N(0, 1)$ and $X_{101}, \ldots X_{200} \sim N(1, 1)$.

A table summarising the results for $\tau = 100$ is given in Table 4.2, for a selection of control parameters. This table provides our first example showing how ARL0 and ARL1 are coupled; tuning parameters to improve ARL0 has a negative effect on ARL1 — if ARL0 increases, ARL1 also increases, and vice versa. This is further demonstrated in Figure 4.4.

If the same experiment is repeated for different values of $\tau$, Figure 4.1 is obtained. Noticing the scale on the $y$-axis, this figure shows that the ARL1 does not vary for dif-

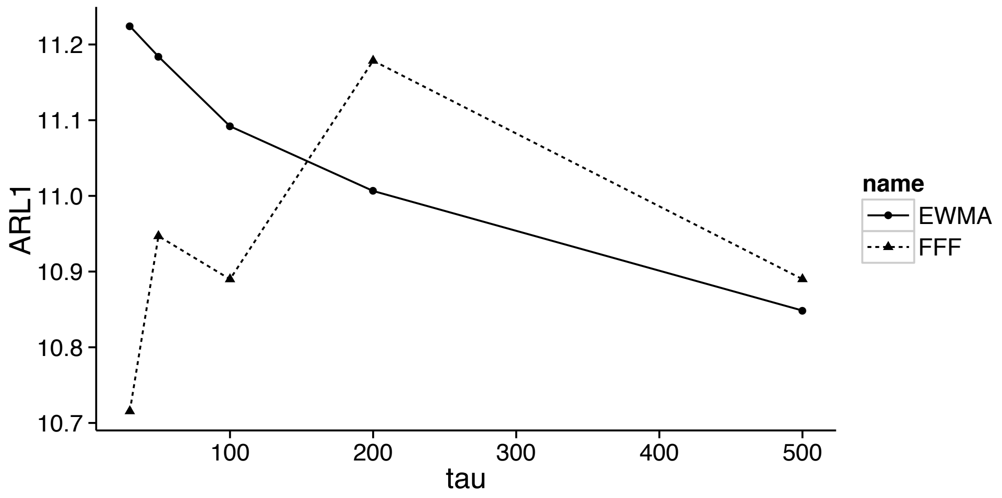

Figure 4.1: EWMA $(0.25, 3.00)$ and FFF $(0.95, 0.01)$ showing the values of ARL1 for different locations of $\tau$.

ferent values of $\tau$, if the pre-change parameters are assumed to be known. The ARL0 of an algorithm does not depend on $\tau$, and so is constant for different choices of $\tau$. Of interest here are the parameter choices where ARL0 approximately matches; in Table 4.2 EWMA $(0.25, 3.00)$ and FFF $(0.95, 0.01)$ both have ARL0 $\sim 400$. Figure 4.1 provides some evidence in support of the claim in Section 3.2.5 that FFF has better ARL1 for short-term changes (smaller values of $\tau$) than EWMA. Admittedly, the variation in ARL1 in Figure 4.1 is small (all values in $[10.7, 11.2]$).

### 4.1.3 Performance of the AFF scheme for different choices of step size

The AFF scheme described in Section 3.3 relies on a gradient descent method to update $\lambda_N \to \lambda_{N+1}$. In Section 3.3.6 the value of $\eta$, the step size in the update equation given in Equation (3.18), is discussed.

Table 4.3 shows that the performance of the AFF algorithm depends on the value of step size $\eta$ to some extent. In particular, the false positive rate as represented by ARL0 is rather variable. While there is a difference in the case of a single changepoint, when there are multiple changepoints, as in Chapter 5, this dependence on $\eta$ is not as apparent, as Table 5.2 shows.

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| AFF | $(\eta, \alpha)$ | $(0.1, 0.01)$ | 390.97 | (399.94) | 10.41 | (5.74) |
| AFF | $(\eta, \alpha)$ | $(0.01, 0.01)$ | 610.71 | (577.21) | 12.58 | (6.43) |
| AFF | $(\eta, \alpha)$ | $(0.001, 0.01)$ | 977.10 | (731.61) | 18.09 | (6.82) |
| AFF | $(\eta, \alpha)$ | $(0.1, 0.05)$ | 76.39 | (87.44) | 7.21 | (4.04) |
| AFF | $(\eta, \alpha)$ | $(0.01, 0.05)$ | 142.46 | (168.92) | 9.35 | (4.39) |
| AFF | $(\eta, \alpha)$ | $(0.001, 0.05)$ | 326.19 | (410.66) | 14.95 | (6.24) |

Table 4.3: An ARL table for AFF for different step size $\eta$ over 1000 trials. For ARL1, $X_1, \dots X_{100} \sim N(0,1)$ and $X_{101}, \dots X_{200} \sim N(1,1)$.

## 4.2 Distribution-free forgetting factor methods

Suppose that $\mu$ and $\sigma^2$ in Equation (4.1) are known, but the underlying distribution of the stream cannot be assumed to be normal. In this case, it would be preferable to use a decision rule that does not have any distributional assumptions in order to signal a change in the AFF mean. In this section, three methods are introduced that have the potential to provide such a decision rule.

### 4.2.1 AFF-Chebyshev method

Under reasonable conditions, Chebyshev's inequality [48] states that for any random variable $X$ with E$[X] = \mu$ and Var$[X] = \sigma^2$, for any $\delta > 0$,

$$\Pr\left(|X - \mu| \geq \delta\sigma\right) \leq \frac{1}{\delta^2}, \tag{4.6}$$

which is equivalent to

$$\Pr\left(|X - \mu| < \delta\sigma\right) > 1 - \frac{1}{\delta^2}.$$

Using the results in Equation (4.2), Chebyshev's inequality can be applied to the AFF equations to yield

$$\Pr\left(|\bar{X}_{N,\overrightarrow{\lambda}} - \mu| < \delta(\sqrt{u_{N,\overrightarrow{\lambda}}})\sigma\right) > 1 - \frac{1}{\delta^2}, \tag{4.7}$$

which for a choice of $\delta$ provides a decision rule

- $|\bar{x}_{N,\lambda} - \mu| < \delta(\sqrt{u_{N,\overrightarrow{\lambda}}})\sigma \Rightarrow$ stream is in-control,
- $|\bar{x}_{N,\lambda} - \mu| \geq \delta(\sqrt{u_{N,\overrightarrow{\lambda}}})\sigma \Rightarrow$ a changepoint has occurred.

A choice of $\delta = 5$ will give $1 - 1/\delta^2 = 0.96$, essentially providing a $96\%$ prediction interval. However, Chebyshev's inequality is conservative, and as Table 4.4 shows, choosing $\delta = 5$ will result in a good ARL0, but a very poor ARL1. Other values of $\delta$ give ARL pairs in a more desirable range, but then relating these $\delta$ values back to the $100(1 - 1/\delta^2)\%$ prediction interval does not provide a good interpretation. So, while this method is appealing, it is not clear how to make a theory-based choice of a value for $\delta$. Table 4.4 shows the behaviour of this scheme for some choices of $\delta$. This method will not be investigated further in this thesis.

As a final point, if $\mu$ and $\sigma^2$ are unknown and estimated during a burn-in period (as in Chapter 5), then a minor modification to the above argument should be made. Rather than using Equation (4.6), a version of Chebyshev's inequality based on estimated parameters [136] should be employed. This version of the inequality is used in Section 7.5.2.

### 4.2.2 Fixed-adaptive forgetting factor method

It was shown in Figure 3.10 that the AFF mean reacts much faster to changes in the mean than the FFF mean when a high fixed $\lambda$ value is used (e.g. when $\lambda = 0.99$). This suggests that if the AFF mean is changing faster than the FFF mean, a changepoint may have occurred. This observation leads to the following decision rule for the AFF mean: for a parameter $\beta > 0$, and a choice of $\lambda$, and supposing the variance in the stream is $\sigma^2$, define

$$a_N = \bar{x}_{N,\lambda} - \beta\sigma,$$
$$b_N = \bar{x}_{N,\lambda} + \beta\sigma.$$

The, a decision rule for signalling a changepoint using the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ is

- $\bar{x}_{N,\overrightarrow{\lambda}} \in (a_N, b_N) \Rightarrow$ stream is in-control,
- $\bar{x}_{N,\overrightarrow{\lambda}} \notin (a_N, b_N) \Rightarrow$ changepoint has occurred.

Of course, this method requires a choice of $\lambda$ and $\beta$. While setting $\lambda = 0.99$ may be straightforward enough, it is not clear what range of values could be used for $\beta$. This method, which we abbreviate to F-AFF, bears some resemblance to the Shewhart chart (see Section 2.1.3), and could be considered to be an "adaptive Shewhart chart". Table 4.4 shows the performance of this method for some values of $\beta$.

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.65) | 230.86 | (172.76) | 9.87 | (5.90) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.70) | 165.44 | (125.18) | 9.24 | (5.35) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 2.50) | 514.04 | (522.70) | 11.92 | (5.92) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 1.00) | 12.02 | (24.02) | 5.64 | (3.46) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 1.00, 0.99) | 1789.26 | (499.28) | 22.22 | (12.40) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 0.50, 0.99) | 463.60 | (420.03) | 15.22 | (7.88) |

Table 4.4: An ARL table for AFF for different learning rates $\eta$ over 1000 trials. For ARL1, $X_1, \dots X_{100} \sim N(0,1)$ and $X_{101}, \dots X_{200} \sim N(1,1)$. A description of the abbreviations used can be found in Table 4.1.

### 4.2.3 Monitoring the AFF

As Figure 3.8 shows, the AFF $\overrightarrow{\lambda}$ reacts to a changepoint by dropping in value. At the time, this was seen as a useful feature that allows the AFF mean to rapidly adjust to the value of the new mean of the stream. However, a natural idea is to use this feature to signal that a changepoint has occurred at time $N$ when $\lambda_N < \gamma$ for some threshold $\gamma$.

The two key parts of this method are the choice of the cost function used for updating $\overrightarrow{\lambda}$, and the threshold $\gamma$ at which a change is signalled. Section 3.3.2 described how the choice of cost function is directly related to the statistic (e.g. mean or variance) that is being monitored for a change. While there are methods that are designed for monitoring the mean and variance simultaneously [54, 62], most methods are designed for monitoring a single statistic. Indeed, there are different versions of CUSUM (and EWMA) for monitoring the mean and for monitoring the variance [104, 30]. Therefore, the choice of cost function is straightforward, since it is based on the statistic being monitored.

On the other hand, it is not immediately clear how to set the value of the threshold $\gamma$. Clearly, $\gamma \in [0, 1)$, since $\overrightarrow{\lambda}$ takes values in $[0, 1]$. It also appears from Figure 3.8

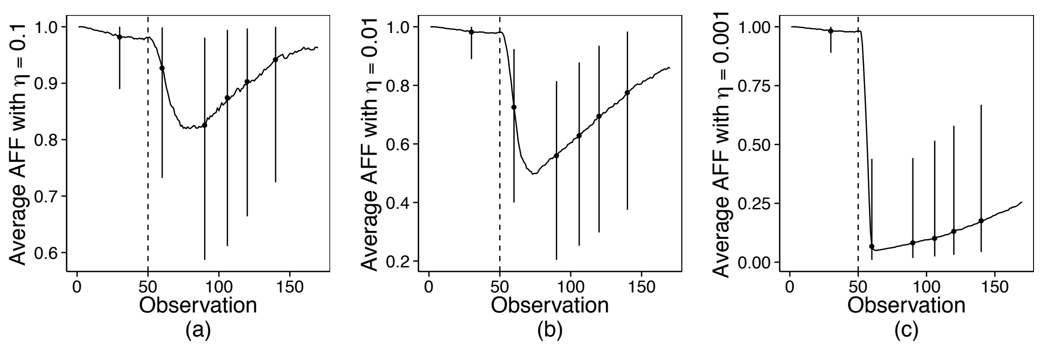

Figure 4.2: Median value of AFF $\overrightarrow{\lambda}$ (over 1000 simulations) for stream generated by $X_1, \dots, X_{50} \sim \mathrm{N}(0,1)$ and $X_{51}, X_{52}, \cdots \sim \mathrm{N}(\mu_2, 1)$, where (a) $\mu_2 = 0.5$, (b) $\mu_2 = 1$ and (c) $\mu_2 = 2$. In all cases $\eta = 0.01$. Error bars indicating the empirical 70% confidence interval are provided.

and other experiments that when the stream is in control $\overrightarrow{\lambda}$ settles down to a value in the range $(0.9, 1)$. While values of $0.8$ or $0.7$ might seem to be appropriate for $\gamma$, the amount that $\overrightarrow{\lambda}$ drops after a change in fact depends on the size of the changepoint, as Figure 4.2 shows. Consequently, it is not clear how to set $\gamma$ when the size of the anticipated change is unknown. Table 4.4 shows this method for a few choices of threshold $\gamma$.

One solution to this reliance on choosing $\gamma$ would be to employ a self-starting method — or perhaps use the AFF-Chebyshev method — to monitor the stream $\lambda_1, \lambda_2, \dots$. While this suggestion may have merit, we do not pursue it further here and leave it for future work.

### 4.2.4 Summary of distribution-free AFF methods

The methods proposed in this section have the benefit of being free of distributional assumptions. However, they all require control parameters to be selected that are not easy to set. This places these methods in a similar position to CUSUM and EWMA, which also rely on the specification of control parameters values where there is no particular theoretical insight into what the values should be.

For this reason, these methods will not be pursued further here and will rather be explored in future work. As a final point, though, it is worth noting that all of the above

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | (1.00, 4.00) | 1784.49 | (489.31) | 25.68 | (19.36) |
| CUSUM | $(k, h)$ | (0.25, 8.00) | 343.37 | (316.99) | 9.87 | (4.41) |
| EWMA | $(r, L)$ | (0.25, 3.00) | 442.86 | (461.07) | 11.09 | (7.31) |
| EWMA | $(r, L)$ | (0.01, 3.00) | 1210.90 | (883.05) | 20.02 | (7.84) |
| EWMA | $(r, L)$ | (0.25, 2.00) | 28.45 | (33.51) | 5.44 | (3.77) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.01) | 405.43 | (403.31) | 10.89 | (4.93) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.05) | 92.29 | (108.01) | 7.92 | (3.88) |
| FFF | $(\lambda, \alpha)$ | (0.99, 0.01) | 1067.49 | (761.43) | 19.65 | (7.70) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.01) | 610.71 | (577.21) | 12.58 | (6.43) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.05) | 142.46 | (168.92) | 9.35 | (4.39) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.65) | 230.86 | (172.76) | 9.87 | (5.90) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.70) | 165.44 | (125.18) | 9.24 | (5.35) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 2.50) | 514.04 | (522.70) | 11.92 | (5.92) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 1.00) | 12.02 | (24.02) | 5.64 | (3.46) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 1.00, 0.99) | 1789.26 | (499.28) | 22.22 | (12.40) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 0.50, 0.99) | 463.60 | (420.03) | 15.22 | (7.88) |

Table 4.5: An ARL table for all the algorithms over 1000 trials. For ARL1, $X_1, \ldots X_{100} \sim N(0,1)$ and $X_{101}, \ldots X_{200} \sim N(1,1)$. Normal streams, parameters are known. A description of the abbreviations used can be found in Table 4.1.

methods can be deployed using estimated values of the stream's mean and variance, and so they also need not rely on assuming the stream parameters are known.

## 4.3 Experiments and results

Having embedded forgetting factor estimation methodology in various change detectors, we turn to consider their performance. In this comparison we consider a single changepoint, following the style of work in statistical process control. The purpose of this exercise is to gain a feel for how change detection algorithms behave against a single change when deployed without a view of the post-change distribution. Such a comparison is a precursor to the multiple changepoint context, which we call *continuous monitoring*, considered in the next chapter.

The experiments examine three situations:

- The streams are normally-distributed, and the pre-change mean and variance are *known*, shown in Table 4.5,

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | (1.00, 4.00) | 1407.18 | (717.11) | 25.13 | (21.06) |
| CUSUM | $(k, h)$ | (0.25, 8.00) | 249.21 | (321.54) | 10.09 | (5.77) |
| EWMA | $(r, L)$ | (0.25, 3.00) | 465.51 | (556.98) | 12.11 | (10.87) |
| EWMA | $(r, L)$ | (0.01, 3.00) | 699.55 | (782.76) | 20.96 | (11.18) |
| EWMA | $(r, L)$ | (0.25, 2.00) | 34.05 | (42.22) | 5.19 | (3.60) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.01) | 285.88 | (385.00) | 11.42 | (6.79) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.05) | 87.18 | (100.16) | 8.38 | (4.72) |
| FFF | $(\lambda, \alpha)$ | (0.99, 0.01) | 562.00 | (614.84) | 20.57 | (11.34) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.01) | 418.50 | (500.93) | 13.30 | (8.84) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.05) | 143.75 | (151.96) | 9.80 | (5.17) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.65) | 188.18 | (216.23) | 9.72 | (5.97) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.70) | 125.73 | (135.44) | 8.97 | (4.74) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 2.50) | 355.64 | (439.24) | 12.60 | (8.02) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 1.00) | 39.64 | (38.86) | 7.00 | (4.09) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 1.00, 0.99) | 1697.72 | (531.34) | 21.95 | (12.12) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 0.50, 0.99) | 463.04 | (453.25) | 14.92 | (7.82) |

Table 4.6: An ARL table for all the algorithms over 1000 trials. For ARL1, $X_1, \ldots X_{100} \sim N(0, 1)$ and $X_{101}, \ldots X_{200} \sim N(1, 1)$. Normal streams, but parameters are estimated during initial an burn-in period of length $B = 50$. A description of the abbreviations used can be found in Table 4.1.

- The streams are normally-distributed, and the pre-change mean and variance are *unknown* and estimated during a burn-in period, shown in Table 4.6,
- The streams are *not* normally-distributed, but the pre-change mean and variance are known, shown in Table 4.7.

In all cases, the mean increases by one multiple of the standard deviation. To reiterate, the purpose of the experiments is not to seek an optimal change detector, but rather to examine performance under different choices of control parameters. As such, no attempt has been made to match algorithms on ARL0.

The notable features in Table 4.5 are that all algorithms performance depend critically on control parameters and, as has been discussed earlier, an increase in ARL0 leads to an increase in ARL1. Indeed, this is thematic in all the tables arising from this experiment.

Table 4.6 shows results for unknown pre-change parameters, which are estimated during a burn-in period. The parameter settings are the same as in Table 4.5. Notably, there is a decrease in ARL0, while ARL1 remains stable. An obvious explanation for this phenomenon is that the false positive rate suffers as a consequence of estimation in the burn-in phase. An exception to this observation is the F-AFF scheme, which has similar performance in both tables.

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | (1.00, 4.00) | 252.68 | (241.87) | 25.77 | (20.80) |
| CUSUM | $(k, h)$ | (0.25, 8.00) | 320.57 | (302.28) | 10.13 | (4.62) |
| EWMA | $(r, L)$ | (0.25, 3.00) | 28.16 | (82.42) | 10.49 | (6.82) |
| EWMA | $(r, L)$ | (0.25, 2.00) | 5.64 | (17.86) | 5.71 | (1.70) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.01) | 412.43 | (419.86) | 11.52 | (5.06) |
| FFF | $(\lambda, \alpha)$ | (0.95, 0.05) | 116.58 | (120.69) | 8.73 | (3.94) |
| FFF | $(\lambda, \alpha)$ | (0.99, 0.01) | 1092.83 | (771.24) | 20.20 | (7.62) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.01) | 453.53 | (448.78) | 12.97 | (6.52) |
| AFF | $(\eta, \alpha)$ | (0.01, 0.05) | 174.65 | (197.45) | 9.97 | (4.46) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.65) | 259.07 | (214.59) | 10.77 | (8.23) |
| AFFdrop | $(\eta, \theta)$ | (0.01, 0.70) | 189.53 | (145.32) | 9.42 | (4.91) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 2.50) | 413.58 | (420.26) | 12.52 | (6.22) |
| AFFcheby | $(\eta, \delta)$ | (0.01, 1.00) | 15.10 | (26.64) | 8.14 | (2.77) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 1.00, 0.99) | 1311.80 | (707.52) | 22.15 | (12.02) |
| F-AFF | $(\eta, \beta, \lambda)$ | (0.01, 0.50, 0.99) | 373.17 | (404.87) | 14.77 | (7.86) |

Table 4.7: An ARL table for all the algorithms over 1000 trials. For ARL1, $X_1, \dots X_{100} \sim \Gamma(1, 1)$ and $X_{101}, \dots X_{200} \sim \Gamma(4, 0.5)$. Non-normal (Gamma) streams, and pre-change mean and variance assumed known. A description of the abbreviations used can be found in Table 4.1.

Table 4.7 considers Gamma-distributed streams. This choice is made to assess detection performance when assumptions are violated. The natural comparison is with Table 4.5 since pre-change mean and variance are treated as known in both cases. On one hand, the ARL0 of CUSUM and EWMA appears to dramatically suffer for certain parameter pairs. On the other hand, the forgetting factor methods all degrade more gently (some not at all).

Two other interesting features arise from this experiment. First, Figure 4.3 shows how different algorithms' detection delay (ARL1) varies with the size of the change. The algorithms exhibit very similar performance. Second, Figure 4.4 shows how ARL0 and ARL1 are coupled for CUSUM, EWMA, FFF and AFF, for different control parameter settings. An ideal algorithm would manifest in the bottom-right corner of each frame of the figure, having high ARL0 (infrequent false positives) and low ARL1 (fast detections). However,

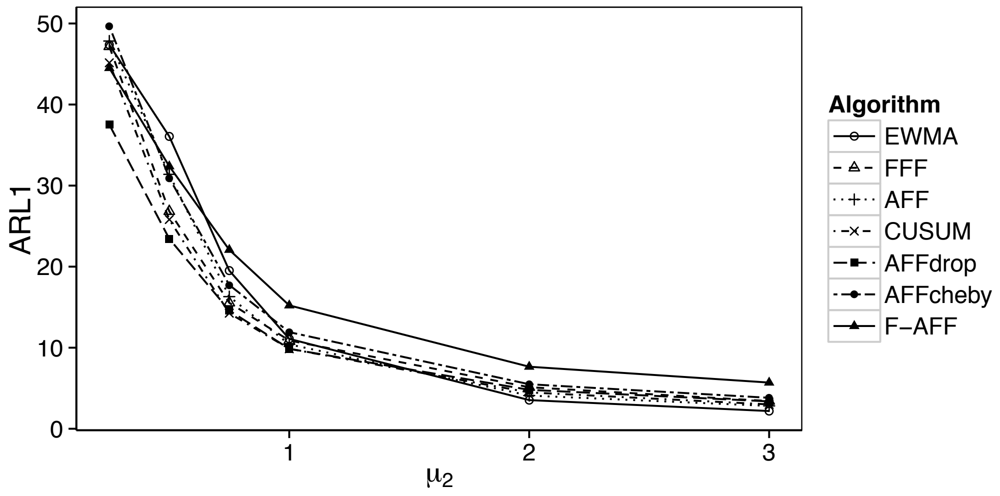

Figure 4.3: EWMA $(0.25, 3.00)$, FFF $(0.95, 0.01)$, AFF $(0.1, 0.01)$, CUSUM $(0.25, 8.00)$, AFFdrop $(0.01, 0.65)$, AFFcheby $(0.01, 2.50)$ and F-AFF $(0.01, 0.50, 0.99)$ showing ARL1 for increasing values of $\mu_2$. The parameters of the normally-distributed streams are assumed known.

no choice of parameter pair yields this behaviour; if ARL0 increases, so does ARL1.

## 4.4 Discussion

A collection of change detection schemes utilising the forgetting factor estimation framework have been proposed and explored. The distribution-free methods appear to have some promise, but are compromised by the difficulty of selecting control parameter values. The main conclusions from this exploration are that FFF appears to detect sudden changes more effectively than EWMA and forgetting factor methods appear more robust to model misspecification than traditional approaches. Note however that these conclusions are based on detecting a single change. In the next chapter certain of the forgetting factor methods are carried forward to the problem of continuous monitoring.

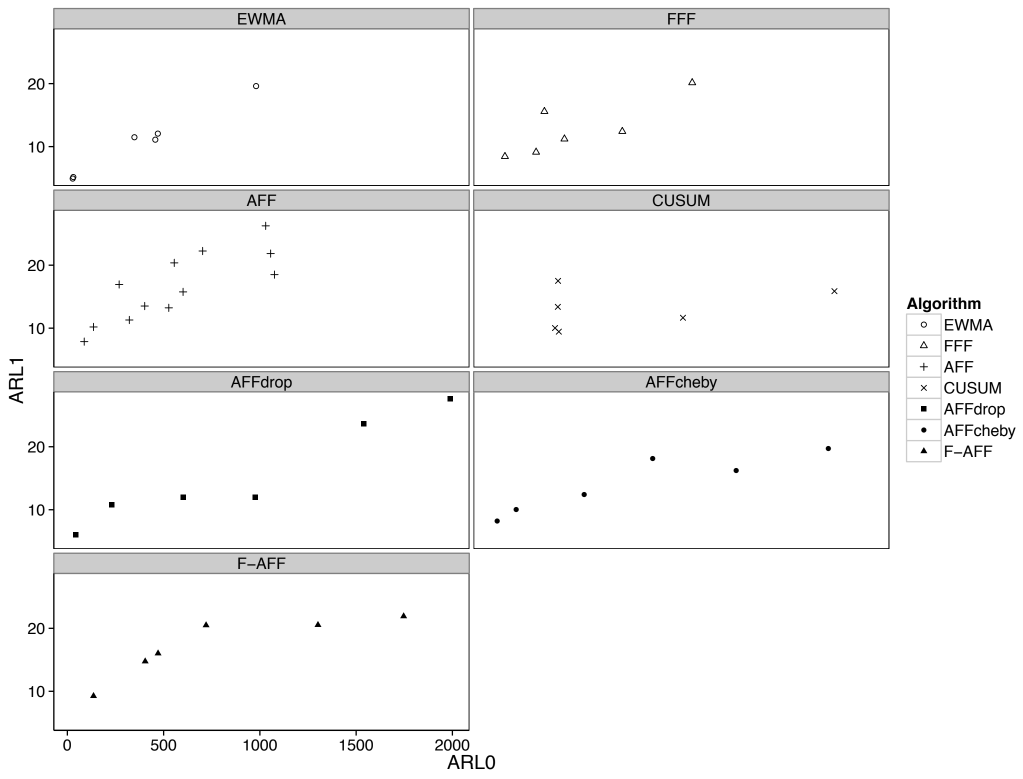

Figure 4.4: ARL0 vs ARL1 for EWMA, FFF, AFF, CUSUM, AFFdrop, AFFcheby and F-AFF for different choices of control parameters. The parameters of the normally-distributed streams are assumed known.

# Chapter 5

# Continuous Monitoring

The previous chapter explored the utility of the forgetting factor methods for detecting a *single* changepoint in streaming data. This included consideration of the effects of burn-in for parameter estimation. However, a data stream, as described in Chapter 1, is potentially unending and is expected to contain multiple changepoints. In this chapter forgetting factor methods are applied to this more difficult problem, referred to as *continuous monitoring*, of detecting *multiple* changepoints in a data stream.

This is a subtle and unexamined problem, and it is not clear whether there is any extant method that is well-matched to meet its challenges. While some methods have characteristics that satisfy certain aspects of the problem, there does not seem to be a single method which satisfies all the requirements. The subtleties in continuous monitoring, in relation to existing literature, are discussed in Section 5.1.1. A key finding in this chapter is that the AFF scheme is well-suited to continuous monitoring; it performs comparably to CUSUM and EWMA, yet only requires a single control parameter, which can be easily set. This reduces the burden on the analyst to set control parameters in a streaming data context.

Section 5.1 formulates the continuous monitoring framework, reviews some literature, and discusses performance metrics in the continuous monitoring context. Section 5.2 describes how we construct streams for a simulation study, and how the performance metrics are computed. Section 5.3 presents results suggesting that, in the continuous monitoring context, the performance of the AFF scheme does not depend on the choice of step size $\eta$. Section 5.4 presents results comparing the AFF scheme and restarting CUSUM and

EWMA. Finally, Section 5.5 demonstrates our change detection methodology in an application related to financial data.

## 5.1 Detecting multiple changepoints in streaming data

Change detection algorithms are usually compared by their ability to find a single changepoint; this was explored in Chapter 4. However, in many real-world situations such as financial monitoring (exemplified in Section 5.5), multiple changepoints are expected and an algorithm must continue to monitor the process for successive changes. Similar problems occur in certain types of security and surveillance applications [52]. In this section we discuss the multiple changepoint scenario and relevant performance metrics.

Denote a stream of observations as $x_1, x_2, \ldots$, sampled from i.i.d. random variables $X_1, X_2, \ldots$, with changepoints $\tau_1, \tau_2, \ldots$, such that

$$\begin{aligned} X_1, X_2, \ldots, X_{\tau_1} &\sim F_1, \\ X_{\tau_1+1}, X_{\tau_1+2}, \ldots, X_{\tau_2} &\sim F_2, \\ X_{\tau_2+1}, X_{\tau_2+2}, \ldots, X_{\tau_3} &\sim F_3, \quad \text{etc,} \end{aligned} \tag{5.1}$$

where $F_1, F_2, \ldots$ represent distributions such that $F_k \neq F_{k+1}$ for all $k$. Recall from Section 2.1 that the size of the $i$th change is defined to be $|E[X_{\tau_i}] - E[X_{\tau_{i-1}}]|$ for a change in the mean. As described in Sections 2.1.7 and 2.1.8, it will be necessary to estimate the stream parameters (mean and variance) for each new regime, i.e. when monitoring starts, and after each detected changepoint. We expect multiple changepoints to occur and each regime could have a different underlying distribution.

### 5.1.1 Recent approaches and continuous monitoring

In Chapter 4 the forgetting factor methods were compared to CUSUM and EWMA because they are two of the most basic and well-studied approaches for sequential change detection. Of course, many sophisticated variations have been proposed, each of which typically handle only one of the challenges in continuous monitoring. Considering the requirements of continous monitoring provides a convenient way to partition the relevant literature:

**(A)** Sequential and efficient computation

**(B)** Handling changes of unknown size

**(C)** Few control parameters

**(D)** Self-starting, or detects multiple changes

Requirement **(B)** has been studied extensively in the context of a *single* change-point. Much of this work is related to so-called adaptive-CUSUM and adaptive-EWMA, see [151] for a review. Note that we are not aware of any literature where both **(B)** and re-starting are addressed together.

An optimal filtering mechanism is provided in [5] which reduces to standard EWMA in special cases. The approach is shown to be effective for both large and small changes, and so satisfies **(B)**. However, this approach is inadequate for continuous monitoring due to **(A)** and **(C)**, specifically the large number of coefficients that need to be estimated in the filter.

In addition to addressing **(B)**, [28] proposes a method that is suitable for different size shifts in the presence of post-change dependence, in the context of a single change. This sophistication comes at some computational cost, which make this approach unsuitable for continuous monitoring in relation to **(A)** and **(C)**.

Again, with respect to **(B)**, [78] propose a hybrid EWMA/CUSUM procedure in the context of a single change. While this approach looks effective in experiments, there are four control parameters to be determined, which violates requirement **(C)**.

The issue of self-starting has been addressed in both univariate and multivariate contexts. For example, [62] is an early approach on multivariate self-starting. This method has two parameters, the setting of which is suggested by reference to standard tables, such as those in [99]. Other approaches to self-starting include [146] and [163]. In all these examples, one way or another, there are control parameter settings that are challenging in the context of continuous monitoring, which violates requirement **(C)**.

There are self-starting methods [93, 150] that appear to be promising for a sequential analysis context, but require the storage of an increasing window of statistics, and so are not suitable for a streaming data context. Moreover, it is not clear how these methods could be modified to detect multiple changepoints.

Finally, while there are approaches for detecting multiple changepoints in a stream (e.g. [163, 102]), in general these are either non-sequential [102] (violating **(A)**) or require several control parameters [163] (violating **(C)**).

The methods discussed in this section are all good approaches when considered in the context for which they were designed, however none of them seem to satisfy all the requirements for detecting changes in streaming data. Furthermore, as mentioned in Section 2.1.7, while most of the traditional SPC literature focuses on detecting a single change after a long period of stationarity, in this chapter we shall consider the scenario where changepoints occur frequently, and so will be using shorter burn-in periods.

### 5.1.2 Performance measures

Assessment of performance becomes complicated once we depart from the most basic sequential change detection setting. For example, in the context of a multivariate change detection problem, [146] is forced to develop an extra performance measure.

Performance assessment is complicated in the continuous monitoring problem, and extends beyond the standard approaches used in the literature. We consider conventional metrics, then performance metrics relevant to the continuous monitoring scenario.

#### Average Run Length

As described in Section 2.1.1, two standard performance measures are the Average Run Lengths, ARL0 and ARL1 [116]. ARL0 is computed as the average number of observations until a changepoint is detected, when the algorithm is run over a sequence of observations with no changepoints, while ARL1 is the average number of observations between a changepoint occurring and the change being detected. Note that ARL1 typical refers to a single change of a given magnitude.

As noted in Chapter 1, the challenge of continuous monitoring involves a sequence of changes of unknown and varying magnitude. These measures alone are insufficient to characterise detection performance in a continuous monitoring framework. Issues related to calculating the ARLs in a continuous monitoring setting are discussed in Section 5.2.2.

### Detection rates

The ARL1 value neither reflects how many changepoints are detected nor how many are missed. Moreover, ARL1 and ARL0 together do not reflect the ratio of true detections to false positives. In a single-change context, these might be difficult to measure, since any reasonable algorithm will detect a change given enough time. However, in a data stream, there is a finite amount of time between changepoints, and some changes might not be detected before another changepoint occurs, and we then classify these as *missed changes*.

Now, suppose that we have a data stream with $C$ changepoints, and our algorithm makes a total of $D$ detections, $T$ of which are true (correct) detections, while $D - T$ are false detections. We then define:

- $CCD = T/C$, the proportion of *changepoints* correctly detected
- $DNF = T/D$, the proportion of *detections* that are not false detections

These intuitive definitions are the same as *sensitivity* and *predicted value positive* (*PVP*) in the surveillance literature [55, 51]. Similar metrics are discussed in [85].

Although the "complements" of $CCD$ and $DNF$ are more intuitively defined (proportions of missed changepoints and false detections, respectively), these definitions are preferred since the closer $CCD$ and $DNF$ are to 1, the better the performance of the algorithm.

## 5.2 A simulation study

In developing new change detection methodology, it is customary to consider the case of normally-distributed data (e.g. [27, 62]). This simulation study follows this custom, letting $F_i \sim N(\mu_i, \sigma_i)$ for all $i$. For this simulation, however, $\sigma_i = 1$ for all $i$.

In order to obtain randomly spaced changepoints, first sample $\xi_i, \xi_2, \ldots \sim$ Pois$(\nu)$, for some value $\nu$, to obtain random interval widths, and then pad these value with $G$ and $D$.

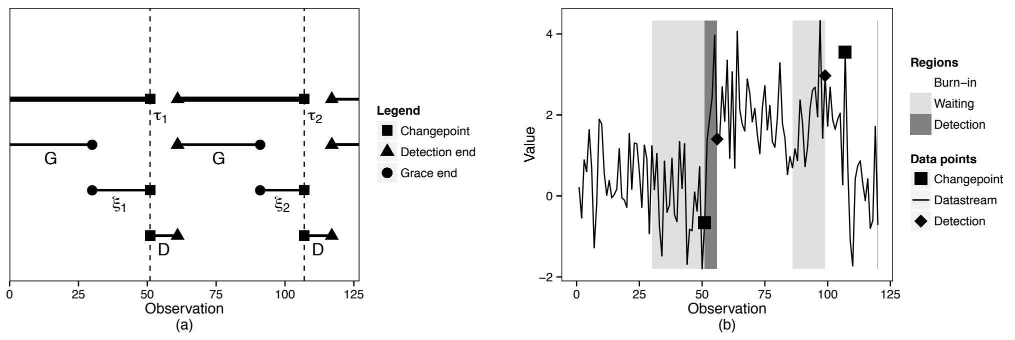

Figure 5.1: (a) Generating the stream. (b) Schematic representation of detection regions.

$G$ is a *grace* period to give the algorithm time to estimate the streams parameters, and $D$ is a period that allows the algorithm to detect a change. The changepoints are then specified by:

$$\begin{aligned} \tau_1 &= G + \xi_1 \\ \tau_k &= \tau_{k-1} + D + G + \xi_k, \qquad k \in \{2, 3, \ldots M\}. \end{aligned}$$

This is schematically represented in Figure 5.1(a). The stream is then generated in blocks $[\tau_k + 1, \tau_{k+1}]$. The first block is sampled from a normal distribution with mean $\mu_1 = 0$, and then block $k$ is sampled with mean $\mu_k = \mu_{k-1} + \delta_k$, where $\delta_k$ is a random jump size in some set $S$.

For the simulations below, the stream is generated with parameters

$$\nu = 30, \qquad G = 30, \qquad D = 30, \qquad M = 50000,$$

and the set of jump sizes $\delta_k$ is uniformly sampled from the set

$$S = \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}.$$

### 5.2.1 Classifying the detected changes

After running over the stream, an algorithm will return a sequence of detected changepoints $\{\widehat{\tau}_1, \widehat{\tau}_2, \dots\}$, and we must then classify these as correct, missed or false detections. In a simulation setting, we will use the sequence of true changepoints $\{\tau_1, \tau_2, \dots\}$ to do this.

Recall that after detecting a changepoint $\widehat{\tau}_n$, our algorithm uses the next $B$ observations in the interval $[\widehat{\tau}_n + 1, \widehat{\tau}_n + B]$ as a *burn-in region* to estimate the parameters of the post-change distribution. The algorithm then monitors the stream until the next true changepoint of the stream occurs at $\tau_m$. Now, if the next detected changepoint is $\widehat{\tau}_{n+1}$, then

- if $\widehat{\tau}_{n+1} \in [\widehat{\tau}_n + B, \tau_m]$, then $\widehat{\tau}_{n+1}$ is a *false detection*,
- if $\widehat{\tau}_{n+1} \in [\tau_m + 1, \tau_{m+1}]$, then $\widehat{\tau}_{n+1}$ is a *correct detection*,
- if $\tau_m$ and $\tau_{m+1}$ occur without a detected changepoint in the interval $[\tau_m, \tau_{m+1}]$, then $\tau_m$ is a *missed detection*.

In order to visualise the situation better, one can imagine that our stream is divided into three regions of different coloured backgrounds, *burn-in*, *waiting*, and *detection* regions. Then, a detected changepoint $\widehat{\tau}_n$ is classified according to the region in which it lies. For example, in Figure 5.1(b) the first detected changepoint is a correct detection, while the second detected changepoint is a false detection. To clarify the definitions, the waiting region is the interval $[\widehat{\tau}_n + B + 1, \tau_{n+1}]$, i.e. the region between the end of the burn-in period and the next true changepoint. The detection region is the interval $[\tau_{n+1} + 1, \widehat{\tau}_{n+1}]$ or $[\tau_{n+1} + 1, \tau_{n+2}]$, depending on whether a changepoint is detected or not.

### 5.2.2 Average run length for a data stream

The calculations of ARL0 and ARL1 are simple using this framework. The ARL0 is the sum of the lengths of the waiting regions between false detections, divided by the number of false detections. The ARL1 is the sum of the lengths of the detection regions between the correctly detected changepoints and their nearest true changepoints. Note that this excludes the detection regions that are between two true changepoints (missed detections).

| Label | Description |
|---|---|
| EWMA | Exponentially weighted moving average scheme (see Section 2.1.5) |
| CUSUM | Cumulative sum scheme (see Section 2.1.4) |
| AFF | Adaptive forgetting factor scheme |
| CCD | Proportion of changepoints correctly detected |
| DNF | Proportion of detections that are not false detections |
| ARL0 | Average number of observations between false alarms |
| SDRL0 | Standard deviation of ARL0 |
| ARL1 | Average delay in detecting a true changepoint |
| SDRL1 | Standard deviation of ARL1 |

Table 5.1: Explanation of labels used for different change detection schemes and performance metrics.

We also sequentially calculate the variances of ARL0 and ARL1 by recording the sum of squares of the lengths used in ARL0 and ARL1. Care must be exercised in the calculation of the variance of ARL0, however, and we must ensure that we take the square of the *sum* of the lengths of the waiting regions between false detections. The standard deviations of ARL0 and ARL1 denoted by SDRL0 and SDRL1, respectively.

While these definitions of ARL0 and ARL1 are unconventional, as we are averaging the delays that occur for detecting changes of different sizes, these definitions are one way for us to obtain an estimate for the average run lengths of the detectors.

Three algorithms will be compared in the next section: CUSUM, EWMA and AFF. A description of these abbreviations is available in Table 5.1.

## 5.3 Choice of step size $\eta$ in the continuous monitoring context

Although the AFF scheme described in Section 4.1 only has a single control parameter $\alpha$, Section 4.1.3 shows that the value of the step size $\eta$, used in Equation (3.18) to update $\overrightarrow{\lambda}$, affects the performance of the AFF scheme when detecting a single change. However, Table 5.2 shows that the AFF algorithm performs relatively consistently, in the continuous monitoring context, for $\eta = 0.1, 0.01, 0.001$. It is interesting that slightly different estimation procedures give comparable change detection performance. It now appears that, for practical purposes, the AFF scheme only depends on the single control parameter $\alpha$, at least in the continuous monitoring context.

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| AFF | $(\eta, \alpha)$ | (0.100, 0.005) | 0.77 | 0.83 | 19.48 | (20.48) | 175.91 | (175.45) |
| AFF | $(\eta, \alpha)$ | (0.010, 0.005) | 0.77 | 0.81 | 19.68 | (20.55) | 148.16 | (152.94) |
| AFF | $(\eta, \alpha)$ | (0.001, 0.005) | 0.79 | 0.81 | 18.56 | (19.46) | 150.05 | (149.94) |

Table 5.2: Summary of algorithm performance, over $\sim$ 50000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ and burn-in $B$=30. This table shows that AFF has similar performance for $\eta = 0.1, 0.01, 0.001$. CCD is the proportion of changepoints correctly detected, and DNF is the proportion of detections that are not false detection. These performance metrics are introduced in Section 5.1.2.

## 5.4 Experiments and results

Table 5.3 displays exemplar results for the CUSUM, EWMA and AFF algorithms for a choice of parameters, with a burn-in of $B = 30$, with results averaged over a stream containing 50000 changepoints. For all of the algorithms, after a changepoint is detected the mean and variance of the new regime are estimated during the burn-in period. The algorithms then use these estimates to detect the next change. Parameters have been chosen in Table 5.3 to give each algorithm approximately comparable performance in terms of ARL0. Specifically, the CUSUM parameters used were indicated in [110, Section 8.1.3] to be common choices of CUSUM parameter pairs. These are almost identical to those recommended in [63, Table 1]. For EWMA, it is often recommended that $r \in [0.05, 0.25]$ [110, Section 8.2.2] and the parameter pairs used are those recommended in [100]. We choose these default parameter values because we have no other choice in continuous monitoring; when there is no knowledge of the pre- or post-change distribution, or when there will be multiple changepoints between different regimes, there is no opportunity to select the optimal parameter pair. It therefore seems reasonable to try a selection of parameter pairs that have good performance for the single changepoint setting.

In Table 5.3 the AFF parameter $\alpha$ was chosen to be $\alpha = 0.005$ to give comparable ARL0 and ARL1 performance to CUSUM and EWMA. Indeed, comparing AFF with $\alpha = 0.005$ to CUSUM with $(k, h) = (1.00, 2.52)$ (both in **bold**), we see that the AFF has almost the same ARL0, slightly higher ARL1, the same $DNF$ value, and a higher $CCD$. Comparisons with the other two CUSUMs (with $(k, h) = (0.50, 4.77)$ and $(k, h) = (0.25, 8.01)$) are similar, but the latter CUSUM has a higher $CCD$ value than AFF. AFF with $\alpha = (0.01)$

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | **(1.00, 2.52)** | **0.68** | **0.83** | **18.35** | (21.34) | **177.72** | (178.42) |
| CUSUM | $(k, h)$ | (0.50, 4.77) | 0.79 | 0.81 | 17.59 | (19.47) | 163.01 | (160.99) |
| CUSUM | $(k, h)$ | (0.25, 8.01) | 0.85 | 0.82 | 18.63 | (18.62) | 169.36 | (158.73) |
| EWMA | $(r, L)$ | **(0.20, 2.962)** | **0.76** | **0.82** | **17.56** | (20.03) | **169.45** | (172.36) |
| EWMA | $(r, L)$ | (0.25, 2.998) | 0.74 | 0.84 | 17.93 | (20.58) | 197.29 | (194.53) |
| EWMA | $(r, L)$ | (0.30, 3.023) | 0.72 | 0.84 | 18.21 | (20.99) | 202.83 | (203.55) |
| AFF | $(\alpha)$ | **(0.005)** | **0.77** | **0.83** | **19.48** | (20.48) | **175.91** | (175.45) |
| AFF | $(\alpha)$ | **(0.01)** | **0.81** | **0.78** | **18.70** | (19.97) | **119.03** | (120.96) |

Table 5.3: Summary of algorithm performance, over $\sim$ 50000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ and burn-in $B$=30. Parameter values have been chosen to give comparable performance in terms of ARL0. See Table 5.1 for a description of the abbreviations used. Highlighted entries are discussed in the text.

(in **red**) shows that increasing $\alpha$ decreases both the ARL0 and ARL1, increases $CCD$ and decreases the $DNF$.

The comparison of AFF with EWMA in Table 5.3 is similar. Comparing AFF with $\alpha = 0.005$ to EWMA with $(r, L) = (0.20, 2.962)$ (also in **bold**), AFF has almost the same ARL0, slightly higher ARL1, and broadly the same $DNF$ and $CCD$. The situation with the other two EWMAs (with $(r, L) = (0.20, 2.988)$ and $(r, L) = (0.30, 3.023)$) is similar, except that the EWMAs have higher ARL0, but slightly lower $CCD$. Table 5.3 therefore indicates that AFF has broadly the same performance as CUSUM and EWMA. However, the benefit of AFF is that it only requires a single control parameter.

Table 5.4 shows how CUSUM, EWMA and AFF behave with different parameter pairs. First of all, CUSUM with $(k, h) = (1.25, 1.99)$ and $(k, h) = (1.50, 1.61)$ (in **blue**) — two recommended choices of parameter pairs in [63] — have similar performance to those choices in Table 5.3, but with lower $CCD$. If parameter pairs are mixed, as for $(k, h) = (0.25, 2.52)$ or $(k, h) = (1.00, 8.01)$ (in **red**), this results in extreme behaviour; either perfect $CCD$ or $DNF$, but at the expense of poor ARL0, $CCD$ and $DNF$. However, a non-standard choice of a parameter pair $(k, h) = (0.50, 8.01)$ (in **green**), can result in good (even superior) performance — note how this case compares to CUSUM with $(k, h) = (1.00, 2.52)$ in Table 5.3 (in **bold**): comparable $CCD$ and ARL1, but $(k, h) = (0.50, 8.01)$ (in **green**) has far better $DNF$ and ARL0. This shows that setting of the CUSUM parameter pair is non-trivial.

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | **(1.25, 1.99)** | **0.64** | **0.83** | **19.03** | (22.19) | **182.01** | (184.08) |
| CUSUM | $(k, h)$ | **(1.50, 1.61)** | **0.60** | **0.84** | **19.64** | (22.85) | **188.39** | (188.35) |
| CUSUM | $(k, h)$ | **(0.25, 2.52)** | **1.00** | **0.46** | **8.66** | (9.29) | **13.46** | (12.10) |
| CUSUM | $(k, h)$ | **(1.00, 8.01)** | **0.44** | **1.00** | **17.67** | (19.73) | **39722.48** | (31397.78) |
| CUSUM | $(k, h)$ | **(0.50, 8.01)** | **0.69** | **0.96** | **19.47** | (19.77) | **885.03** | (897.55) |
| EWMA | $(r, L)$ | (0.05, 2.615) | 1.00 | 0.35 | 1.10 | (2.15) | 1.12 | (3.58) |
| EWMA | $(r, L)$ | (0.10, 2.814) | 0.99 | 0.35 | 1.70 | (5.17) | 1.79 | (13.51) |
| AFF | $(\alpha)$ | (0.05) | 0.90 | 0.60 | 15.13 | (17.54) | 40.47 | (44.24) |

Table 5.4: Summary of algorithm performance, over $\sim$ 50000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ and burn-in $B$=30. Parameter values have been chosen to show how poorly-chosen parameters can lead to poor performance. Highlighted entries are discussed in the text.

Also in Table 5.4 are two parameter pair choices for EWMA that are recommended in [100], but have very poor performance (very high $CCD$ at the expense of poor performance for the other three metrics). This also shows that setting the EWMA parameter pair is non-trivial. Finally, AFF with $\alpha = 0.05$ is shown to give good $CCD$, but at the expense of the other three metrics. Since increasing $\alpha$ makes the AFF scheme more sensitive to changes, this behaviour is expected. However, since $\alpha$ is the only control parameter, it is relatively easy to adjust the performance of AFF by increasing or decreasing $\alpha$.

Table 5.5 shows how CUSUM, EWMA and AFF behave when the burn-in is $B = 50$, instead of $B = 30$ in Table 5.3. To make comparison fair, the grace period $G$ is set to $G = 50$, otherwise the stream parameters are unchanged. This table shows that the algorithms have comparably the same performance, compared to Table 5.3, except that ARL0 and ARL1 are increased for all algorithms. Note, however, that $CCD$ and $DNF$ are relatively similar, or only slightly increased.

Recall from Section 2.1.5 that the original EWMA paper [128] shows that the EWMA scheme is sensitive to small change sizes. In the continuous monitoring scenario, changes of different sizes are plausible, hence Tables 5.2-5.5 consider streams with change sizes $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ (i.e. both small and large changes). Table 5.6 compares the algorithms when all the changes are relatively large, and shows that, compared to Table 5.3, there is a large increase in $CCD$, a slight decrease in $DNF$, a large decrease in ARL1, and similar ARL0. The increase in $CCD$ and decrease in ARL1 should be expected, since the

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | (1.00, 2.52) | 0.73 | 0.82 | 25.08 | (30.48) | 226.78 | (230.07) |
| CUSUM | $(k, h)$ | (0.50, 4.77) | 0.83 | 0.81 | 22.38 | (27.02) | 215.55 | (217.82) |
| CUSUM | $(k, h)$ | (0.25, 8.01) | 0.88 | 0.82 | 22.81 | (25.10) | 214.90 | (206.51) |
| EWMA | $(r, L)$ | (0.20, 2.962) | 0.80 | 0.83 | 22.60 | (27.90) | 254.12 | (253.01) |
| EWMA | $(r, L)$ | (0.25, 2.998) | 0.78 | 0.84 | 23.12 | (28.56) | 263.92 | (265.30) |
| EWMA | $(r, L)$ | (0.30, 3.023) | 0.76 | 0.84 | 23.98 | (29.45) | 270.02 | (273.78) |
| AFF | $(\alpha)$ | (0.005) | 0.79 | 0.82 | 24.79 | (28.12) | 224.19 | (228.69) |
| AFF | $(\alpha)$ | (0.01) | 0.83 | 0.77 | 23.67 | (27.23) | 148.16 | (153.57) |

Table 5.5: Summary of algorithm performance, over $\sim$ 50000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ and burn-in $B$=50. This table can be compared with Table 5.3 to show how the burn-in length affects performance.

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| CUSUM | $(k, h)$ | (1.00, 2.52) | 0.92 | 0.79 | 7.40 | (11.26) | 178.72 | (182.23) |
| CUSUM | $(k, h)$ | (0.50, 4.77) | 0.96 | 0.78 | 6.86 | (8.31) | 162.85 | (160.12) |
| CUSUM | $(k, h)$ | (0.25, 8.01) | 0.97 | 0.79 | 7.90 | (8.05) | 164.28 | (153.25) |
| EWMA | $(r, L)$ | (0.20, 2.962) | 0.95 | 0.74 | 6.60 | (8.63) | 122.49 | (145.12) |
| EWMA | $(r, L)$ | (0.25, 2.998) | 0.95 | 0.81 | 6.81 | (9.33) | 195.70 | (194.31) |
| EWMA | $(r, L)$ | (0.30, 3.023) | 0.94 | 0.81 | 7.03 | (10.09) | 202.18 | (203.80) |
| AFF | $(\alpha)$ | (0.005) | 0.95 | 0.79 | 8.26 | (10.10) | 172.40 | (174.21) |
| AFF | $(\alpha)$ | (0.010) | 0.95 | 0.73 | 7.92 | (10.01) | 115.17 | (119.68) |

Table 5.6: Summary of algorithm performance, over $\sim$ 50000 changepoints, with $\delta \in \{\pm 1, \pm 2, \pm 3, \pm 4\}$ and burn-in $B$=30. This table corresponds to Table 5.3, but the stream has a different set of jump sizes.

jump sizes are larger.

In conclusion, Tables 5.3, 5.5 and 5.6 show that AFF has similar performance to CUSUM and EWMA when parameters are chosen for all algorithms to have similar ARL0. However, selecting parameter pairs for CUSUM and EWMA is not easy. Table 5.4 shows that standard parameter pair choices can lead to poor performance, and non-standard choices can give better performance. The benefit of AFF is that it only requires a single control parameter $\alpha$, since the adaptive forgetting factor $\overrightarrow{\lambda}$ is automatically tuned. It can be seen that increasing performance for one metric generally decreases performance for other metrics. In particular, it appears that parameter choices that increase $CCD$ lead to a decrease in $DNF$, and vice-versa. Therefore, $\alpha$ can be increased or decreased to adjust for desired $CCD$ or $DNF$ performance for AFF.

## 5.5 Foreign exchange data

As discussed in the introduction, a primary application of continuous monitoring for data streams arises in financial trading. Here, the value of a financial instrument evolves over time, as a result of the behaviour of the market. Individual traders need to determine if the price has made an unexpected change in order to trigger trading actions. However, the data stream continues, uninterrupted, as such trading decisions happen. For illustration, we will consider 5-minute Foreign Exchange (FX) tick data. Specifically, we consider a stream of Swiss Franc (CHF) and Pound Sterling (GBP). Our objective here is simply to detect changes in the price-ratio, that could be used to trigger trading actions.

It is well known that FX streams are non-stationary. The standard approach to address this problem is to transform the data, and analyse the so-called log-returns, $LR_t = \log(x_t) - \log(x_{t-1})$. We perform change detection on the log-returns of the CHF/GBP data for the first 10000 observations. The data is from 07h05 on 21/10/2002 until 21h15 on 10/12/2002 (approximately seven weeks), with one data point every five minutes. We restrict to a short sequence simply for the purposes of clarity. Figure 5.2(a) shows the changepoints (vertical lines) detected on the log-returns superimposed on the raw data stream for the AFF scheme with $\alpha = 0.005$.

Although this section is simply meant to provide an example of the AFF scheme deployed on real data, we also provide a comparison with PELT [87], an optimal *offline* detection algorithm, in order to provide an indication of the “true” changepoints. The changepoints detected by PELT are shown in Figure 5.2(b). PELT also has a single control parameter, which was chosen to be $0.01$ for this figure, in order to detect a comparable number of changepoints. Figure 5.2 shows a high degree of agreement between the AFF scheme and PELT, with more than half of the changepoints common to both (within 3 observations of each other). This is particularly striking since the AFF scheme is an online method while PELT is an offline method. Similar figures are obtained for different parameter values that increase sensitivity and allow more changepoints to be detected. We use the R implementation of PELT provided in the *changepoint* package [86].

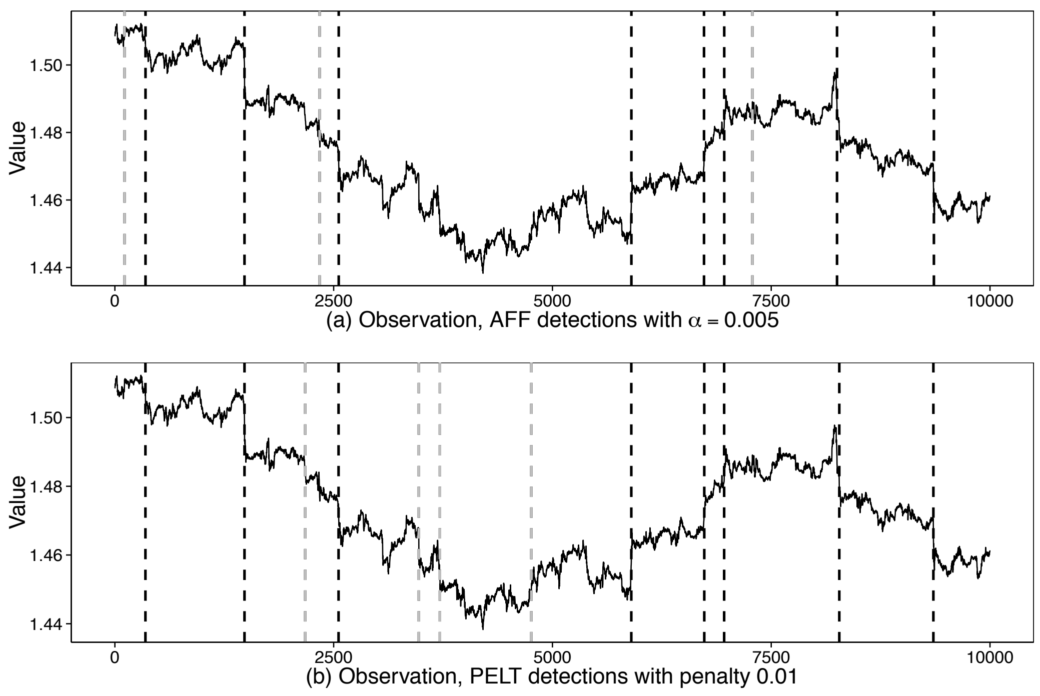

Figure 5.2: Change detection on a CHF/GBP data stream using AFF and PELT. The raw data stream is plotted with the detected changepoints indicated by the vertical dashed lines, black lines indicate that both schemes detect that changepoint (within 3 observations of each other).

## 5.6 Discussion

This chapter has extended the case of detecting of a single changepoint using the forgetting factor framework (Chapter 4) to the context of continuous monitoring, which more closely addresses the challenges of detecting changes in streaming data. Some recent literature is reviewed, and none of the methods seem to satisfy all the requirements of continuous monitoring. Performance metrics for continuous monitoring are then discussed.

It is found that the AFF scheme performs similarly for a wide range of step size values $\eta$, and so the AFF scheme truly requires only a single control parameter, namely the sensitivity $\alpha$.

An extensive simulation study is performed which shows that the AFF scheme has similar change detection performance to CUSUM and EWMA. However, these two methods require two control parameters while the AFF scheme only requires one. This is important because, on the one hand, simply increasing or decreasing the AFF parameter $\alpha$ will either increase or decrease the algorithm's sensitivity to detecting changes. On the other hand, setting the two control parameters of CUSUM and EWMA is non-trivial. While Table 5.3 shows that some recommended settings give good performance, Table 5.4 shows that other recommended choices can lead to poor performance (EWMA), and mixing parameter pairs can lead to poor performance or performance superior to that when recommended settings are used (CUSUM).

Finally, the AFF scheme is applied to detect changes in the mean of a foreign exchange stream. For comparison, an optimal offline method is run over the same stream, and there is good agreement between the changepoints detected by the two methods. This provides some evidence that the AFF scheme can detect changepoints in real-world data streams. Another application in which this forgetting factor framework has been applied is the detection of “relays”, a suspicious kind of behaviour in computer network traffic, by extending the framework to an extreme-value scenario [16].

While the preceding chapters have assumed that the data stream consists of univariate observations, the next chapter extends our forgetting factor framework to the detection of changes in multivariate data.

# Chapter 6

# Multivariate adaptive filtering and change detection

While there are many applications that require the monitoring of a single stream for potential changes, there are situations where it could be desirable to monitor a collection of related streams simultaneously. In a computer network, multiple network traffic ports could be monitored for anomalous behaviour. In the world of finance, a collection of foreign exchange pairs (e.g. see Section 5.5) or a portfolio of share prices could be monitored for an increase in volatility. Indeed, there are scenarios that may not immediately spring to mind, but are no less important. For example, an early reference [72] refers to "sample" bomb sites. In this chapter, multivariate forgetting factor schemes are proposed to sequentially detect multiple changepoints in the mean of a multivariate data stream. It is a natural extension of the work in discussed in Chapters 3, 4 and 5, although new issues have to addressed. As before, a particular concern is managing the dependence on control parameters. In the multivariate case, this is more complicated and we settle on an easier interpretation for setting the control parameters, rather than full automation.

Section 6.1 reviews some of the multivariate change detection literature. Section 6.2 introduces the notation for the multivariate AFF mean and describes two methods for incorporating an adaptive forgetting factor. Section 6.3 describes a decision rule for detecting a change using this AFF framework. Naturally, in the multivariate setting there are more issues to consider for defining such rules. A simulation study in Section 6.4 shows that this method performs as well as, if not better than, some recently proposed methods even though it requires only a single control parameter to be specified. Finally, Section 6.5 applies our multivariate methodology to monitoring the volume of port traffic in a computer network.

## 6.1 Multivariate change detection in the literature

A good overview of multivariate statistical process control can be found in [105]. Several of the univariate charts described in Section 2.1 have been extended to the multivariate setting. For example, there are several multivariate extensions of EWMA [98, 64] and CUSUM [160, 70, 40]. A good comparison of the early methods can be found in [121].

More recently, a multivariate version of the changepoint model has been proposed in [164, 165]. Besides these, there are methods using regression [123, 125, 124, 166] and LASSO [168]. A recent method [162] using generalised likelihood ratio statistics assumes the streams are independent and normally distributed. Self-starting methods have been recently proposed in [147, 65, 101]. However, all of these methods, while sequential, are only designed to detect a *single* changepoint. While there are methods explicitly designed for detecting multiple changepoints in multivariate data [92, 102], these methods are usually offline (non-sequential). Again, as in the discussion in Section 5.1.1, there are no multivariate methods that satisfy all the requirements of continuous monitoring.

In Section 6.4, the SSMEWMA method described in [65] will be used as a basis of comparison for our multivariate forgetting factor methods. Although it was only considered in [65] in the context of detecting a single change, it is the method that can be most easily adapted to the continuous monitoring context. However, its methodology relies on sequential regression on the components of the stream after each new data point has been observed. Consequently, it is computationally expensive when the number of components $d$ is large. The multivariate CUSUM method referred to as $MC1$ in [121] will also used as a benchmark for comparison in Section 6.4.

## 6.2 Multivariate adaptive forgetting factor mean

Suppose that the process being monitored is now multivariate, and that each observation in the stream $\mathbf{x}_1, \mathbf{x}_2, \ldots$ is $d$-dimensional, i.e.

$$\mathbf{x}_i = \begin{pmatrix} x_{i,1} \\ x_{i,2} \\ \vdots \\ x_{i,d} \end{pmatrix}, \qquad i = 1, 2, \ldots.$$

This formulation also allows us to consider the sequences $x_{1,j}, x_{2,j}, \ldots$ for $j = 1, 2, \ldots, d$ to be a collection of $d$ streams that are being observed simultaneously. For an AFF $\overrightarrow{\lambda}$ as defined in Section 3.3, the **multivariate adaptive forgetting factor (MVAFF) mean** $\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}}$ is naturally defined as

$$\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} = \begin{pmatrix} \bar{x}_{N,\overrightarrow{\lambda},1} \\ \bar{x}_{N,\overrightarrow{\lambda},2} \\ \vdots \\ \bar{x}_{N,\overrightarrow{\lambda},d} \end{pmatrix}, \qquad i = 1, 2, \ldots.$$

where each $\bar{x}_{N,\overrightarrow{\lambda},j}$ is the AFF mean as defined in Section 3.3 of the $j$th component of the stream $\mathbf{x}_1, \mathbf{x}_2, \ldots$, for $j = 1, 2, \ldots, d$. To be clear, in this case $\overrightarrow{\lambda}$ is a single scalar forgetting factor for all the component streams. Another formulation is described in Section 6.2.1. The MVAFF mean can be equivalently defined for $N \geq 1$ by the vector equations

$$\begin{aligned} \mathbf{x}_{N,\overrightarrow{\lambda}} &= \left[ I_d \cdot \mathbf{w}_{N,\overrightarrow{\lambda}} \right]^{-1} \cdot \mathbf{m}_{N,\overrightarrow{\lambda}} && (6.1) \\ \mathbf{m}_{N,\overrightarrow{\lambda}} &= \lambda_{N-1} \mathbf{m}_{N-1,\overrightarrow{\lambda}} + \mathbf{x}_N, \qquad \mathbf{m}_{0,\overrightarrow{\lambda}} = \mathbf{0}_d \\ \mathbf{w}_{N,\overrightarrow{\lambda}} &= \lambda_{N-1} \mathbf{w}_{N-1,\overrightarrow{\lambda}} + \mathbf{1}_d, \qquad \mathbf{w}_{0,\overrightarrow{\lambda}} = \mathbf{0}_d \end{aligned}$$

where $I_d$ is the $d$-dimensional identity matrix, $\mathbf{1}_d$ is a vector of length $d$ with all entries equal to 1, and $\mathbf{0}_d$ is a vector of length $d$ with all entries equal to 0. In terms of updating $\overrightarrow{\lambda}$, as in Equation (3.18), one possible cost function would be the multivariate analogue of

$L_{N+1,\overrightarrow{\lambda}}$ (defined in Equation 3.26), defined by

$$\mathbf{L}_{N+1,\overrightarrow{\lambda}} = \left[\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} - \mathbf{x}_{N+1}\right]^T \left[\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} - \mathbf{x}_{N+1}\right].$$

Then the AFF $\overrightarrow{\lambda}$ is updated the according to

$$\lambda_{N+1} = \lambda_N - \eta \frac{\partial}{\partial \overrightarrow{\lambda}} \mathbf{L}_{N+1,\overrightarrow{\lambda}}, \tag{6.2}$$

which is the natural multivariate analogue of Equation (3.18). As before, $\eta$ is the step size Equation (6.2) becomes,

$$\begin{aligned}\lambda_{N+1} &= \lambda_N - \eta \frac{\partial}{\partial \overrightarrow{\lambda}} \left( \left[\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} - \mathbf{x}_{N+1}\right]^T \left[\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} - \mathbf{x}_{N+1}\right] \right) \\ &= \lambda_N - 2\eta \left[ \frac{\partial}{\partial \overrightarrow{\lambda}} \bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} \right]^T \left[\bar{\mathbf{x}}_{N,\overrightarrow{\lambda}} - \mathbf{x}_{N+1}\right].\end{aligned}$$

The FFF scheme is defined as in Equation (6.1), but by setting $\lambda_N = \lambda$, for all $N = 1, 2, \ldots$, for some fixed $\lambda$.

### 6.2.1 Adaptive forgetting factors for each stream

In the formulation above, the same AFF $\overrightarrow{\lambda}$ is used for each component of the stream $\mathbf{x}_1, \mathbf{x}_2, \ldots$, even though the $d$ component streams $x_{1,j}, x_{2,j}, \ldots$ may be in different states of control. While the above formulation may have been the most straightforward, upon reflection it may seem desirable for each component stream to have its own forgetting factor. The $d$-dimensional MVAFF $\overrightarrow{\Lambda}$ is defined as $\overrightarrow{\Lambda} = (\Lambda_1, \Lambda_2, \ldots)$, where $\Lambda_i$ is the diagonal matrix

$$\Lambda_i = \begin{pmatrix} \lambda_{i,1} & 0 & \cdots & 0 \\ 0 & \lambda_{i,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \lambda_{i,d} \end{pmatrix}, \qquad i = 1, 2, \ldots \tag{6.3}$$

and the sequential update equations in Equation (6.1) become

$$\begin{aligned}
\mathbf{x}_{N,\overrightarrow{\lambda}} &= \left[I_p \cdot \mathbf{w}_{N,\overrightarrow{\lambda}}\right]^{-1} \cdot \mathbf{m}_{N,\overrightarrow{\lambda}} && (6.4)\\
\mathbf{m}_{N,\overrightarrow{\lambda}} &= \Lambda_{N-1} \cdot \mathbf{m}_{N-1,\overrightarrow{\lambda}} + \mathbf{x}_N, \qquad \mathbf{m}_{0,\overrightarrow{\lambda}} = \mathbf{0}_d\\
\mathbf{w}_{N,\overrightarrow{\lambda}} &= \Lambda_{N-1} \cdot \mathbf{w}_{N-1,\overrightarrow{\lambda}} + \mathbf{1}_d, \qquad \mathbf{w}_{0,\overrightarrow{\lambda}} = \mathbf{0}_d.
\end{aligned}$$

The $j$th component of $\Lambda_i$ is updated by

$$\lambda_{N+1,j} = \lambda_{N,j} - 2\eta_j \left[\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda},j}\right] \cdot \left[\bar{x}_{N,\overrightarrow{\lambda},j} - x_{N+1}\right], \tag{6.5}$$

where the derivative of $\bar{x}_{N,\overrightarrow{\lambda},j}$ is given by Equation (3.25), and $\eta_j$ is the step-size for the $j$th component. It is possible that this formulation may appear overly complicated due to the number of subscripts involved. However, this formulation can be viewed as giving each component stream $x_{1,j}, x_{2,j}, \dots$ its own AFF $(\lambda_{1,j}, \lambda_{2,j}, \dots)$, for $j = 1, 2, \dots, d$. Furthermore, the separate forgetting factors allows the step-size $\eta$ to be scaled by E $\left[\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}}\right]$ as described in Section 3.3.6, which is a decided advantage, since it removed some of the dependence on the value of $\eta$. Suppose that the variance of $d$ components are estimated to be $\widehat{\sigma}_1, \widehat{\sigma}_2, \dots, \widehat{\sigma}_d$, then the multivariate analogue for Equation (3.31) is

$$\lambda_{N+1,j} = \lambda_{N,j} - 2\frac{\eta_j}{\widehat{\sigma}_j^2} \left[\frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda},j}\right] \cdot \left[\bar{x}_{N,\overrightarrow{\lambda},j} - x_{N+1}\right], \tag{6.6}$$

As can be seen in Table 6.2, this scaling gives similar change detection performance for a range of values of $\eta$. For this reason, the AFF schemes considered in the simulation study in Section 6.4 utilise a separate forgetting factor for each stream. Note that for the multivariate FFF scheme, there is no difference between having a single or separate forgetting factors. In the next section, we discuss different decision rules for detecting a change in a multivariate stream.

## 6.3 Decision rules for multivariate change detection

Suppose the $d$-dimensional stream $\mathbf{x}_1, \mathbf{x}_2, \ldots$ is distributed according to the multivariate normal distribution $\mathrm{N}(\mu_d, \Sigma)$. There are two cases: the component streams can be assumed to be independent, or the covariance can be taken into account.

### 6.3.1 Assuming the streams are independent

If the component streams are considered to be independent, each stream can be considered to be distributed as

$$x_{1,j}, x_{2,j}, \cdots \sim \mathrm{N}(\mu_j, \sigma_j^2), \qquad j = 1, 2, \ldots, d. \tag{6.7}$$

Estimates $\widehat{\mu}_j$ and $\widehat{\sigma}_j^2$ can be obtained during a burn-in period. Then, at time $N$, a value

$$p_j = F_{\mathrm{N}(\widehat{\mu}_j, \widehat{\sigma}_j^2)} \left( \bar{x}_{N, \vec{\lambda}, j} \right) \tag{6.8}$$

can be computed, where $F_{\mathrm{N}(\mu,\sigma^2)}$ is the cdf of $\mathrm{N}(\mu, \sigma^2)$. As in Section 4.1.1, $p_j$ can be turned into a $p$-value by

$$p_j^{'} = 1 - |1 - 2p_j|$$

and we could say that a change has been detected in the $j$th component stream if

$$p_j^{'} < \alpha$$

for some $\alpha \in [0, 1]$. So far, this is the same procedure as for the univariate case considered in Chapters 4 and 5. However, we do not want to only detect a change in a single component, but rather a change in the stream $\mathbf{x}_1, \mathbf{x}_2, \ldots$. Therefore, we compute the $p$-values $p_1^{'}, p_2^{'}, \ldots, p_d^{'}$, and combine these into an overall $p$-value. There are two prominent methods for combining several $p$-values: Fisher's method [49] and Stouffer's method [144] (also known as the Z-method). These two methods are briefly described in Appendix A.1. There has been research [95, 96, 157, 35] comparing the two methods, but they are broadly similar. For Stouffer's method, the $p$-values $p_1^{'}, p_2^{'}, \dots, p_d^{'}$ are combined to give

$$p' = F_{\mathrm{N}(0,1)}\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d} F^{-1}_{\mathrm{N}(0,1)}(1-p_i')\right),$$

where $F_{\mathrm{N}(0,1)}$ is the cdf and $F^{-1}_{\mathrm{N}(0,1)}$ is the inverse of the cdf of the $\mathrm{N}(0,1)$ distribution. A change is then signalled when

$$p' < \alpha.$$

Fisher's method combines the $p$-values via

$$p' = F_{\chi^2_{2d}}\left(-2\sum_{j=1}^{d}\log(1-p_j')\right),$$

where $F_{\chi^2_{2d}}$ is the cdf of the chi-squared distribution with $2d$ degrees of freedom. Again, $p' < \alpha$ signals a change. Both these schemes assume that the $p$-values (and hence the streams) are independent. Next we will consider the case when the streams are not assumed to be independent.

## 6.3.2 Estimating the covariance

The covariance matrix $\Sigma$ of a multivariate stream $\mathbf{x}_1, \mathbf{x}_2, \dots$ can be estimated from the first $N$ observations by $\widehat{\Sigma}_N$, which can be computed sequentially [4] by

$$\begin{aligned}
\bar{\mathbf{x}}_N &= \left(1-\frac{1}{N}\right)\bar{\mathbf{x}}_{N-1} + \frac{1}{N}\mathbf{x}_N, & \bar{\mathbf{x}}_N &= \mathbf{0}_p,\\
\widehat{\Pi}_N &= \left(1-\frac{1}{N}\right)\widehat{\Pi}_{N-1} + \frac{1}{N}\mathbf{x}_N^T\mathbf{x}_N, & \widehat{\Pi}_N &= \mathbf{0}_p,\\
\widehat{\Sigma}_N &= \widehat{\Pi}_N - \mathbf{x}_N^T\mathbf{x}_N.
\end{aligned}$$

It is possible to compute a forgetting factor version of $\widehat{\Sigma}_N$, as in [4], but that raises the question of how to set the value of the forgetting factor. Certainly, the AFF methodology considered in Chapter 3 could be employed with a suitable choice of cost function but, as Chapter 8 will show, implementing AFF estimation for the univariate variance is not straightforward. There are now at least three possibilities:

1. do not estimate the covariance matrix (assume streams are independent)
2. estimate the covariance matrix during the burn-in, and assume the covariance remains the same after the burn-in
3. continue to estimate the covariance matrix continuously

We either assume the streams are independent, or estimate the covariance during a burn-in period. The case where the covariance matrix is continuously estimated is not considered here.

### 6.3.3 Taking the covariance into account: Brown's method

In many cases it is not reasonable to assume the component streams are independent, and so the covariance between the streams needs to be taken into account. Suppose the covariance matrix is estimated during a burn-in period, using the equations given in Section 6.3.2. First, the method of Section 6.3.1 is followed until the one-sided p-values $p_1^{'}, p_2^{'}, \dots, p_p^{'}$ have been computed. Next, an extension of Fisher's method then provides a method for combining the p-values while taking the covariance between the streams into account. This method, originally published by M. B. Brown in [20] and slightly improved in [88], is now briefly described. Start by defining $X^2$ to be

$$X^2 = -2 \sum_{j=1}^{d} (1 - p_j').$$

This is simply Fisher's method described in Appendix A.1.1. If the $p$-values are independent, then $X^2$ follows a chi-squared distribution with $2d$ degrees of freedom. If the $p$-values are not independent, then

$$\mathrm{E}\left[X^2\right] = 2d, \tag{6.9}$$

$$\mathrm{Var}\left[X^2\right] = 4d + 2 \sum_{j=1}^{d} \sum_{i<j} \mathrm{Cov}(-2 \log p_i', -2 \log p_j'). \tag{6.10}$$

The covariance terms can then be approximated using Gaussian quadrature [20] in terms of the correlation values $\rho_{ij}$. Recall that if $(i, j)$th entry of the covariance matrix is $c_{ij}$, then

$$\rho_{ij} = \frac{c_{ij}}{\sqrt{c_{ii}c_{jj}}}.$$

A third-order approximation, given in [88], is then

$$\text{Cov}(-2\log p'_i, -2\log p'_j) = 3.263\rho_{ij} + 0.710\rho_{ij}^2 + 0.027\rho_{ij}^3.$$

This approximation is said to work well [88] as long as $-0.98 \leq \rho_{ij} \leq 0.98$, which is a broad range of values since $\rho_{ij} \in [-1, 1]$. Finally, the first two central moments of $X^2$ given in Equations (6.9) and (6.10) are matched to the first two moments of a $\Gamma(\widehat{k}, \widehat{\theta})$ distribution (see Section 7.3.2, the *Satterthwaite-Welch approximation*) by

$$\widehat{k} = (\text{E}\left[X^2\right])^2/\text{Var}\left[X^2\right], \qquad \widehat{\theta} = \text{Var}\left[X^2\right]/\text{E}\left[X^2\right].$$

Then the combined $p$-value is

$$p' = \Phi_{\Gamma(\widehat{k},\widehat{\theta})}(X^2),$$

where $\Phi_{\Gamma(\widehat{k},\widehat{\theta})}$ is the cdf of the $\Gamma(\widehat{k}, \widehat{\theta})$ distribution and, again, a change is signalled if

$$p' < \alpha.$$

## 6.4 A simulation study

We follow the method described in Section 5.2 for generating a *single univariate* normally-distributed data stream with multiple changepoints, and again use the values

$$\nu = 30, D = 30, G = 30, M = 10000, \tag{6.11}$$

where $\nu$ is the Poisson parameter, $D$ is the period allowed for the algorithm to detect a change, $G$ is the grace period before a change can possibly occur (to give the algorithm time to estimate the stream parameters) and $M$ is the number of changepoints in the stream.

| Label | Description |
|---|---|
| MVFFF-S | Fixed forgetting factor, using Stouffer's method, assuming independent streams |
| MVFFF-F | Fixed forgetting factor, with Fisher's method, assuming independent streams |
| MVFFF-Bcov | Fixed forgetting factor, with Brown's method, taking covariance into account |
| MVAFF-S | Adaptive forgetting factor, using Stouffer's method, assuming independent streams |
| MVAFF-F | Adaptive forgetting factor, with Fisher's method, assuming independent streams |
| MVAFF-Bcov | Adaptive forgetting factor, with Brown's method, taking covariance into account |
| MVCUSUM | Multivariate version of CUSUM, described in [121] as $MC1$ |
| SSMEWMA | Self-starting multivariate EWMA, described in [65] |

Table 6.1: Explanation of labels used for different change detection schemes

Again, the size of the change in the mean for each regime is uniformly sampled from $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$.

Then, we combine this stream with three stationary $\mathrm{N}(0, 1)$-distributed streams (no changepoints) and monitor these four streams for changes using multivariate adaptive estimation as described above. Of course, other formulations are possible, but this formulation is simple and easy to analyse. If there are changes occurring in different streams at different times, difficulties could arise in the analysis; for example, if two changes in different component streams occur close together (in time), and a change is detected soon after the later change, which changepoint is being detected? Therefore, we use the formulation where only a single component stream is changing, as in Chapter 5. Then the changepoint in the multivariate stream is the location of the changepoint in the non-stationary univariate stream. The multivariate AFF and FFF estimation schemes are then used with either

1. Stouffer's method,
2. Fisher's method,
3. Brown's method, i.e. Fisher's method taking covariance into account,

to create multivariate change detection schemes. Consequently, these six schemes are labelled as in Table 6.1. As benchmarks for these forgetting factor methods, multivariate versions of CUSUM and EWMA are implemented. The multivariate CUSUM procedure used is $MC1$ from [121] and is labelled *MVCUSUM*. The multivariate EWMA is a recent and sophisticated self-starting multivariate EWMA [65] that uses regression between the

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| MVAFF-S | $(\eta, \alpha)$ | (0.100, 0.01) | 0.78 | 0.89 | 24.86 | (22.02) | 275.43 | (262.42) |
| MVAFF-S | $(\eta, \alpha)$ | **(0.010, 0.01)** | **0.75** | **0.93** | **28.22** | (24.15) | **441.83** | (413.61) |
| MVAFF-S | $(\eta, \alpha)$ | **(0.001, 0.01)** | **0.75** | **0.94** | **28.63** | (23.32) | **482.67** | (459.08) |
| MVAFF-S | $(\eta, \alpha)$ | (0.100, 0.05) | 0.91 | 0.75 | 22.11 | (19.71) | 89.06 | (76.51) |
| MVAFF-S | $(\eta, \alpha)$ | (0.010, 0.05) | 0.86 | 0.83 | 24.60 | (21.26) | 149.48 | (127.53) |
| MVAFF-S | $(\eta, \alpha)$ | (0.001, 0.05) | 0.85 | 0.85 | 25.74 | (20.89) | 181.77 | (165.49) |
| MVAFF-F | $(\eta, \alpha)$ | (0.100, 0.01) | 0.82 | 0.85 | 22.69 | (21.62) | 186.87 | (176.53) |
| MVAFF-F | $(\eta, \alpha)$ | (0.010, 0.01) | 0.80 | 0.90 | 24.63 | (22.64) | 299.99 | (273.95) |
| MVAFF-F | $(\eta, \alpha)$ | (0.001, 0.01) | 0.82 | 0.91 | 25.76 | (21.34) | 328.24 | (303.27) |
| MVAFF-F | $(\eta, \alpha)$ | (0.100, 0.05) | 0.92 | 0.71 | 19.95 | (18.82) | 74.00 | (62.78) |
| MVAFF-F | $(\eta, \alpha)$ | (0.010, 0.05) | 0.89 | 0.79 | 22.16 | (20.06) | 115.37 | (102.19) |
| MVAFF-F | $(\eta, \alpha)$ | (0.001, 0.05) | 0.88 | 0.83 | 23.73 | (19.64) | 159.46 | (142.53) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.100, 0.01) | 0.82 | 0.85 | 22.52 | (21.60) | 187.56 | (173.79) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.010, 0.01) | 0.79 | 0.89 | 24.44 | (22.63) | 287.44 | (271.81) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.001, 0.01) | 0.82 | 0.91 | 25.56 | (21.18) | 329.07 | (308.05) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.100, 0.05) | 0.92 | 0.72 | 19.86 | (18.89) | 74.97 | (65.18) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.010, 0.05) | 0.88 | 0.79 | 22.25 | (20.12) | 115.03 | (100.15) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.001, 0.05) | 0.89 | 0.84 | 23.77 | (19.83) | 162.85 | (145.93) |

Table 6.2: Summary of algorithm performance, listed, over 10000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ with burn-in $B$=30. This table shows that the different MVAFF schemes have similar performance for $\eta = 0.01$ and $\eta = 0.001$. CCD is the proportion of changepoints correctly detected, and DNF is the proportion of detections that are not false detection. These performance metrics are introduced in Section 5.1.2. Highlighted entries are discussed in the text.

components of the observations, and is labelled *SSMEWMA*. These algorithms are all compared in Sections 6.4.1 and 6.4.2, which consider the two cases where the streams are (a) independent or (b) dependent.

## 6.4.1 Experiments and results: independent streams

In this section the streams are normally-distributed with covariance matrix $\Sigma_{\text{indep}} = I_4$, the $4 \times 4$ identity matrix. Therefore, each stream is independent of the other streams, and each stream has variance $\sigma^2 = 1$. Recall that one of the streams is non-stationary, while the other three streams are stationary.

First it is useful to compare the AFF schemes. Table 6.2 shows that in this scenario, when the streams are independent, for $\eta = 0.010$ and $\eta = 0.001$ the change detection per-

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| MVAFF-S | $(\eta, \alpha)$ | $(0.100, 0.01)$ | 0.78 | 0.89 | 24.86 | (22.02) | 275.43 | (262.42) |
| MVAFF-F | $(\eta, \alpha)$ | $(0.010, 0.01)$ | 0.80 | 0.90 | 24.63 | (22.64) | 299.99 | (273.95) |
| MVAFF-Bcov | $(\eta, \alpha)$ | $(0.010, 0.01)$ | 0.79 | 0.89 | 24.44 | (22.63) | 287.44 | (271.81) |
| MVAFF-S | $(\eta, \alpha)$ | $(0.010, 0.05)$ | 0.86 | 0.83 | 24.60 | (21.26) | 149.48 | (127.53) |
| MVAFF-F | $(\eta, \alpha)$ | **(0.001, 0.05)** | **0.88** | **0.83** | **23.73** | (19.64) | **159.46** | (142.53) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.001, 0.05)** | **0.89** | **0.84** | **23.77** | (19.83) | **162.85** | (145.93) |
| MVFFF-S | $(\lambda, \alpha)$ | (0.99, 0.05) | 0.91 | 0.80 | 24.51 | (19.15) | 115.31 | (96.07) |
| MVFFF-F | $(\lambda, \alpha)$ | (0.99, 0.05) | 0.91 | 0.79 | 23.70 | (19.49) | 109.03 | (91.43) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | (0.99, 0.05) | 0.90 | 0.79 | 23.40 | (19.31) | 112.32 | (93.47) |
| MVAFF-F | $(\eta, \alpha)$ | $(0.01, 0.05)$ | 0.89 | 0.79 | 22.16 | (20.06) | 115.37 | (102.19) |
| MVAFF-Bcov | $(\eta, \alpha)$ | $(0.01, 0.05)$ | 0.88 | 0.79 | 22.25 | (20.12) | 115.03 | (100.15) |

Table 6.3: Summary of algorithm performance, listed, over 10000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ with burn-in $B$=30. This table shows that, when the streams are independent, Fisher's method and Brown's method yield very similar results for the MVAFF and MVFFF schemes. Highlighted entries are discussed in the text.

formance of the MVAFF schemes is very similar. The CCD, DNF and ARL1 are extremely close, and ARL0 is very similar, although slightly larger for $\eta = 0.001$. For example, looking at MVAFF-S with $\alpha = 0.01$ and $\eta = 0.010, 0.001$ (indicated in **bold** on Table 6.2), the CCD, DNF and ARL1 values are almost exactly the same, while the ARL0 values are very similar. For the MVAFF-F and MVAFF-Bcov schemes with $\alpha = 0.01$ there is similar agreement for $\eta = 0.010$ and $\eta = 0.001$. However, for all the MVAFF schemes, using $\eta = 0.100$ results in different behaviour to using $\eta = 0.010$ and $\eta = 0.001$. Therefore, although the value of $\eta$ may be unimportant as long as it is small enough, larger values of $\eta$ will produce different performance. This is not quite as strong as the univariate case in Chapter 5, where Table 5.2 shows that $\eta = 0.1, 0.01, 0.001$ all produce very similar change detection performance. However, the range $[0.001, 0.01]$ still provides some freedom with which to choose $\eta$ and still obtain very similar results. Since $\eta = 0.001$ appears to produce slightly better ARL0, with all other metrics being equal, this is the value used in Table 6.4 when MVAFF and MVFFF are compared to MVCUSUM and SSMEWMA.

Table 6.3 shows that when the streams are independent, using Fisher's method and Brown's method yields almost identical results. For example, for MVAFF-F and MVAFF-Bcov with $(\eta, \alpha) = (0.001, 0.05)$ (indicated in bold), the CCD, DNF, ARL1 and ARL0 values are virtually identical. Interestingly, Stouffer's method also performs similarly, but

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| MVCUSUM | $(k, h)$ | (1.00, 2.490) | 0.78 | 0.69 | 19.49 | (21.35) | 63.71 | (65.01) |
| MVCUSUM | $(k, h)$ | **(0.50, 4.770)** | **0.86** | **0.69** | **18.26** | (19.21) | **65.23** | (63.10) |
| MVCUSUM | $(k, h)$ | (0.25, 8.000) | 0.92 | 0.69 | 19.37 | (17.88) | 62.47 | (54.26) |
| SSMEWMA | $(\lambda, h)$ | **(0.10, 12.907)** | **0.74** | **0.88** | **18.46** | (20.62) | **294.04** | (284.09) |
| SSMEWMA | $(\lambda, h)$ | (0.10, 11.119) | 0.82 | 0.80 | 18.33 | (20.80) | 147.98 | (148.33) |
| SSMEWMA | $(\lambda, h)$ | (0.20, 12.194) | 0.77 | 0.81 | 18.83 | (21.84) | 150.32 | (149.64) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | (0.99, 0.050) | 0.90 | 0.79 | 23.40 | (19.31) | 112.32 | (93.47) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | **(0.95, 0.005)** | **0.86** | **0.77** | **22.02** | (19.78) | **100.76** | (87.09) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | (0.95, 0.010) | 0.88 | 0.73 | 21.03 | (18.82) | 80.66 | (67.19) |
| MVAFF-Bcov | $(\alpha)$ | **(0.01)** | **0.82** | **0.91** | **25.56** | (21.18) | **329.07** | (308.05) |
| MVAFF-Bcov | $(\alpha)$ | (0.05) | 0.89 | 0.84 | 23.77 | (19.83) | 162.85 | (145.93) |
| MVAFF-Bcov | $(\alpha)$ | (0.10) | 0.91 | 0.79 | 22.96 | (18.91) | 111.29 | (91.48) |
| MVAFF-S | $(\alpha)$ | **(0.01)** | **0.75** | **0.94** | **28.63** | (23.32) | **482.67** | (459.08) |
| MVAFF-S | $(\alpha)$ | (0.05) | 0.85 | 0.85 | 25.74 | (20.89) | 181.77 | (165.49) |
| MVAFF-S | $(\alpha)$ | (0.10) | 0.89 | 0.80 | 24.47 | (19.59) | 117.13 | (100.62) |

Table 6.4: Summary of algorithm performance, listed, over 10000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ with burn-in $B$=30. This table compares the forgetting factor methods to MVCUSUM and SSMEWMA when the streams are *independent*. The MVAFF methods use $\eta = 0.001$. Highlighted entries are discussed in the text.

with $\eta = 0.01$ rather than $\eta = 0.001$. Table 6.3 also shows that the MVFFF (fixed forgetting) schemes perform very similarly, for Stouffer's, Fisher's and Brown's method.

Table 6.4 compares the forgetting factor schemes to MVCUSUM and SSMEWMA. First, one notices that MVCUSUM has much lower ARL0 than the other methods, without a great improvement in the other metrics. It is more interesting to compare the MVAFF schemes to SSMEWMA. Comparing SSMEWMA with $(\lambda, h) = (0.10, 12.907)$ to MVAFF-Bcov with $\alpha = 0.01$ (both indicated in **bold**), we see that these two schemes both have similar DNF and ARL0, but the MVAFF-Bcov scheme has higher CCD and ARL1. In fact, this MVAFF-Bcov (in **bold**) can be compared in the same way with all three SSMEWMA schemes (MVAFF-Bcov has the same or higher CCD, DNF and ARL0, but also has higher ARL1).

Comparing the same (**bold**) SSMEWMA scheme to MVAFF-S with $\alpha = 0.1$ (also **bold**), we see that the MVAFF-S has the same CCD, and higher DNF, ARL1 and ARL0. Recall that lower ARL1 indicates better performance, while higher values for all the other metrics indicate better performance. Therefore, while the MVAFF schemes may have the same or better values for CCD, DNF and ARL0, they also have higher ARL1 values, which is not an improvement. Similar comparisons can be made for the other SSMEWMA schemes (different parameter choices). Therefore, it is not clear whether MVAFF or SSMEWMA has better performance, but they can at least be said to have comparable performance.

Table 6.4 also contains values for the fixed forgetting scheme MVFFF-Bcov. While the MVFFF-Bcov method has lower ARL0 than SSMEWMA, and so may not be directly comparable to it, we can compare MVFFF-Bcov to MVCUSUM. Comparing MVFFF-Bcov with $(\lambda, \alpha) = (0.95, 0.005)$ (in **blue**) to MVCUSUM with $(k, h) = (0.50, 4.770)$ (also in **blue**), we see that they have the same CCD, but MVFFF-BCov has much higher DNF and ARL0, at the expense of a slightly higher ARL1. Again, there is no clear winner here, but MVFFF-Bcov is performing well in comparison to MVCUSUM.

To summarise our results when the streams are independent, while the forgetting factor methods may not clearly outperform SSMEWMA and MVCUSUM, they do perform well in comparison. Also, it is relatively easy to set meaningful values for their control parameters, and the MVAFF schemes do not appear to depend on $\eta$ as long as it is small enough. The next section will consider the case when the component streams are dependent.

### 6.4.2 Experiments and results: dependent streams

Suppose that a single univariate stream is generated with changes as described above, but now the other streams are generated so that each 4-dimensional observation is generated as before, but now the normally-distributed observations are generated with mean vector $\mu_{\text{dep}}$ and covariance matrix $\Sigma_{\text{dep}}$:

$$\mu_{\text{dep}} = \begin{pmatrix} \mu \\ 0 \\ 0 \\ 0 \end{pmatrix}, \qquad \Sigma_{\text{dep}} = \begin{pmatrix} 1.00 & 0.32 & 0.54 & 0.27 \\ 0.32 & 1.00 & 0.82 & 0.48 \\ 0.54 & 0.82 & 1.00 & 0.53 \\ 0.27 & 0.48 & 0.53 & 1.00 \end{pmatrix}. \qquad (6.12)$$

This matrix was obtained by randomly generating $\text{N}(0.5, 0.2^2)$-distributed values for the upper-triangular entries of a $4 \times 4$ matrix, adding 1's to the diagonal, symmetrising, and then

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| MVAFF-F | $(\eta, \alpha)$ | (0.100, 0.01) | 0.85 | 0.77 | 22.52 | (21.02) | 100.76 | (93.12) |
| MVAFF-F | $(\eta, \alpha)$ | (0.010, 0.01) | 0.81 | 0.85 | 24.84 | (22.41) | 176.40 | (159.64) |
| MVAFF-F | $(\eta, \alpha)$ | (0.001, 0.01) | 0.80 | 0.88 | 25.66 | (21.36) | 248.88 | (228.71) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.100, 0.05)** | **0.85** | **0.76** | **22.43** | (21.00) | **98.47** | (92.54) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.010, 0.05)** | **0.81** | **0.84** | **24.87** | (22.45) | **168.78** | (152.36) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.001, 0.05)** | **0.80** | **0.88** | **25.35** | (21.04) | **241.07** | (230.05) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.100, 0.01) | 0.69 | 0.89 | 25.31 | (23.32) | 283.05 | (269.83) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.010, 0.01) | 0.67 | 0.93 | 27.82 | (25.23) | 466.15 | (422.47) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.001, 0.01) | 0.66 | 0.94 | 26.11 | (23.16) | 581.70 | (585.96) |
| MVAFF-S | $(\eta, \alpha)$ | (0.100, 0.01) | 0.84 | 0.78 | 24.89 | (21.31) | 100.04 | (90.71) |
| MVAFF-S | $(\eta, \alpha)$ | (0.010, 0.01) | 0.77 | 0.86 | 28.08 | (23.34) | 183.10 | (162.90) |
| MVAFF-S | $(\eta, \alpha)$ | (0.001, 0.01) | 0.73 | 0.89 | 28.60 | (23.12) | 253.93 | (236.43) |

Table 6.5: Summary of algorithm performance, listed, over 10000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ with burn-in $B$=30. This table shows how, for dependent streams, the value of $\eta$ affects the performance of the AFF schemes. Highlighted entries are discussed in the text.

checking that it is positive-definite. The value of $\mu$, the first component in the mean vector $\mu_{\text{dep}}$ changes to $\mu+\delta_i$ at the $i$th changepoint $\tau_i$. As in Section 6.4.1, $\delta_i \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$.

Table 6.5 shows that when the streams are dependent, as above, then the value of $\eta$ does affect how the MVAFF schemes perform. For example, consider AFF-Bcov with $\alpha = 0.05$ (in **bold**), as $\eta$ decreases from $0.100$ to $0.010$ to $0.001$, CCD decreases while DNF, ARL1 and ARL0 all increase, to varying degrees.

The differences may not be great in some cases, such as for AFF-Bcov with $\alpha = 0.01$, which has fairly similar CCD, DNF and ARL1 values for $\eta = 0.100, 0.010, 0.001$, but there is a significant increase in ARL0. This table shows that we need to take some care in setting $\eta$, and again it seems as if smaller $\eta$ values are better, at least in terms of ARL0. For this reason, it is again recommended to set $\eta = 0.001$ for the MVAFF schemes.

Table 6.6 compares the forgetting factors schemes with MVCUSUM and SSMEWMA when the streams are dependent. MVCUSUM has improved CCD, but even worse DNF and ARL0, and so would not be recommended for use in this setting. Again, we focus on comparing the MVAFF schemes to SSMEWMA.

The most favourable match is perhaps SSMEWMA with $(\lambda, h) = (0.20, 12.194)$ (in **bold**), compared to MVAFF-Bcov with $(\eta, \alpha) = (0.001, 0.05)$ (in **bold**). These two

| Algo | Params | Values | CCD | DNF | ARL1 | SDRL1 | ARL0 | SDRL0 |
|---|---|---|---|---|---|---|---|---|
| MVCUSUM | $(k, h)$ | (1.00, 2.490) | 0.90 | 0.55 | 16.57 | (17.74) | 25.35 | (25.84) |
| MVCUSUM | $(k, h)$ | (0.50, 4.770) | 0.91 | 0.61 | 17.53 | (18.05) | 39.58 | (37.95) |
| MVCUSUM | $(k, h)$ | (0.25, 8.000) | 0.93 | 0.66 | 18.85 | (17.56) | 54.29 | (50.03) |
| SSMEWMA | $(\lambda, h)$ | **(0.10, 12.907)** | **0.78** | **0.88** | **17.17** | (20.13) | **296.70** | (295.13) |
| SSMEWMA | $(\lambda, h)$ | **(0.10, 11.119)** | **0.84** | **0.80** | **16.41** | (19.60) | **147.10** | (148.52) |
| SSMEWMA | $(\lambda, h)$ | **(0.20, 12.194)** | **0.79** | **0.80** | **17.48** | (21.33) | **145.73** | (145.40) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | **(0.99, 0.050)** | **0.85** | **0.85** | **25.92** | (20.80) | **175.50** | (159.85) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | (0.95, 0.005) | 0.73 | 0.86 | 24.41 | (21.82) | 202.15 | (194.15) |
| MVFFF-Bcov | $(\lambda, \alpha)$ | (0.95, 0.010) | 0.78 | 0.82 | 23.64 | (21.09) | 151.96 | (136.11) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.001, 0.01) | 0.66 | 0.94 | 26.11 | (23.16) | 581.70 | (585.96) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.001, 0.05)** | **0.80** | **0.88** | **25.35** | (21.04) | **241.07** | (230.05) |
| MVAFF-Bcov | $(\eta, \alpha)$ | (0.001, 0.10) | 0.86 | 0.83 | 24.40 | (20.24) | 153.73 | (140.36) |
| MVAFF-Bcov | $(\eta, \alpha)$ | **(0.010, 0.05)** | **0.81** | **0.84** | **24.87** | (22.45) | **168.78** | (152.36) |

Table 6.6: Summary of algorithm performance, listed, over 10000 changepoints, with $\delta \in \{\pm 0.25, \pm 0.5, \pm 1, \pm 3\}$ with burn-in $B$=30. This table compares the performance of the forgetting factor schemes with MVCUSUM and SSMEWMA, when the streams are *dependent*. Highlighted entries are discussed in the text.

schemes have similar CCD, but MVAFF-BCov has significantly higher DNF and ARL0, but also has higher ARL1. A similar comparison can be made with MVAFF-Bcov with $(\eta, \alpha) = (0.01, 0.05)$ (in **green**). Again, as before, there is no clear winner, but performance is at least comparable.

SSMEWMA with $(\lambda, h) = (0.10, 12.907)$ (in **red**)can be compared to MVAFF-Bcov with $(\eta, \alpha) = (0.001, 0.05)$ (in **bold**), and is found to have similar CCD, DNF, but higher ARL0 and lower ARL1, suggesting this SSMEWMA scheme has better performance. However, it would be difficult to know *a priori* that this choice of parameters may yield better performance.

The (fixed forgetting) MVFFF-BCov scheme performs surprisingly well in comparison to the SSMEWMA schemes for dependent streams, when one considers that its performance for independent streams was not especially good. For example, MVFFF-Bcov with $(\lambda, \alpha) = (0.99, 0.050)$ (in **blue**) compared to SSMEWMA with $(\lambda, h) = (0.10, 11.119)$ (also in **blue**), shows similar CCD, better DNF and ARL0, but significantly higher ARL1.

In summary, for dependent streams it appears that SSMEWMA in most cases has slightly better performance than the forgetting factor schemes, but there are cases where performance is at least comparable, if not better, for MVAFF. Overall, it appears that AFF-

Bcov performs well, whether or not the streams are dependent. In addition, MVAFF-BCov is more efficient than SSMEWMA, because, as mentioned in Section 6.1, SSMEWMA performs sequential linear regression for each new observation, which is at least of order $O(d^2)$, where $d$ is the number of components in the stream. On the other hand, once the covariance matrix has been estimated, MVAFF-Bcov is of order $O(d)$. This will make a difference when $d$ is large.

In the next section the multivariate MVAFF methodology is applied to detecting changes in computer network traffic. This work originally appeared in [13].

## 6.5 Monitoring a computer network

Attacks on computer networks usually cause changes in network traffic that are often only observed in the final stages of the attack. Examples include worm-based attacks [155], distributed denial-of-service (DDoS) attacks [108], and port-scanning [153], [82]. Over the last two decades there has been much research into methods that attempt to detect these attacks in their early stages.

Intrusion detection systems (IDS) have historically been characterised as either signature-detection systems or anomaly-detection systems [84]. Signature-based methods which usually operate at network packet level detect attacks by comparing network behaviour against a database of known attack behaviours, called signatures. Examples of such methods are Bro [119] and Snort [130]. The strength of IDS is that they often operate at a host-level, which distributes the computational burden.

Anomaly-detection methods attempt to detect any unusual activity in the network by monitoring for deviations from the network's standard behaviour [45]. Examples include D-WARD [109] and MULTOPS [56]. The advantage of these methods over signature-based methods is that anomaly-detectors have the potential to detect a wider variety of attacks, and do not require the compilation (and regular updating) of a signature database. There are anomaly-detection methods in the networks literature, but many require offline processing (e.g. [91]).

The analysis in this section is performed using NetFlow data [1]. NetFlow is a protocol for collecting and storing statistics on the packet volumes of IP flows through a router. It is

```
Date flow start         DurationProto Src IP Addr     Dst IP Addr    Src Pt Dst Pt Packets Bytes
2009-04-22 12:04:44.664 13.632   TCP   126.253.5.69    124.195.12.24649882  80     1507    2.1 M
2009-04-22 12:04:42.613 16.768   TCP   126.253.5.69    124.195.12.24649881  80     2179    3.1 M
2009-04-22 12:04:36.404 17.600   TCP   100.253.210.138100.253.192.991104   80     6017    276899
2009-04-22 12:04:58.736 17.344   TCP   126.253.5.69    124.195.12.24649888  80     1708    2.3 M
```

Figure 6.1: An example of NetFlow data.

a much coarser grained representation than packet capture. An (anonymised) example of NetFlow data is given in Figure 6.1. For a specific *flow* (a collection of packets) between two IP addresses, NetFlow data embodies information about protocols, packet numbers and volumes. NetFlow data allows an organisation-wide view of network traffic, but can be large and unwieldy to handle.

Since flows may be characterised by their (source or destination) ports, we apply our AFF multivariate change detector to monitor the volume of network traffic flowing through selected TCP ports. In this way, our anomaly detector will operate at the level of a router, or a computer monitoring a router.

### 6.5.1 Change detection on NetFlow Data

The multivariate change detection methodology developed and tested above is now deployed on real data, specifically, NetFlow data collected on a single router at Imperial College over a 14-day period in 2009. There are numerous options for selecting or designing features for an anomaly detector to run across. In this case, for simplicity, we consider two variables, the volume of traffic on destination Port 80 (http), and the volume of traffic on all other ports. This choice was made partially based on knowledge of the router's role. Since NetFlow data is essentially continuous time, some binning is required for our methodology. In the example we provide, a binning of 100 minutes is used. Thus, the stream consists of sequential 2-vectors reporting the volume of traffic on Port 80 and all other ports. We use a log transformation of both variables since they have only positive support. We do not commit to these being the best or right choices, but rather intend to provide an illustration of the methodology.

Figure 6.2 provides a representation of the raw data, with markers to demonstrate changepoints identified by our methodology (MVAFF-S). For this illustration, no attempt is made to handle the obvious seasonality present in the data. Indeed, while it is possible to attempt to track any data seasonalities, the benefit of doing so is not clear. A deviation from a specific seasonality may or may not be an indication of a changepoint.

We emphasise that this methodology can be applied to detect changes on $d$-dimensional streams, where $d > 2$, and that we have just used $d = 2$ here merely for illustrative purposes. Furthermore, we reiterate that since this is real data, we have no way of knowing the location of the true changepoints (if indeed any occur), and that this section is simply to demonstrate the methodology in action.

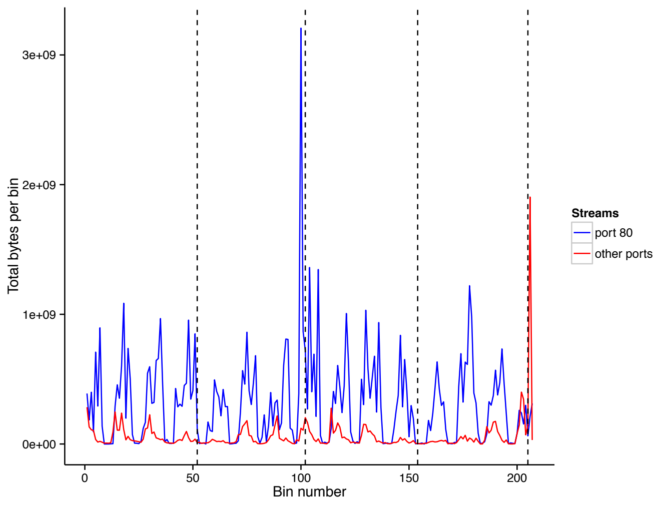

Figure 6.2: Detecting changes in NetFlow data traffic across two ports. Changepoints detected by MVAFF-S are indicated by vertical lines.

In this context of broader application, this analytic is intended to filter NetFlow data in an attempt to reduce information overload on the network analyst. Thus, the analyst would not be routinely concerned with the raw NetFlow data, but would be presented with summaries in relation to detected anomalies. It is worth noting, in the context of organisation-wide network traffic analysis, that triage of detected anomalies is *always* required. For example, an organisation-wide software update would very likely result in a detected anomaly, which is naturally explained by the analyst.

Consider Figure 6.3 as a simple example of what could be provided to an analyst. Each

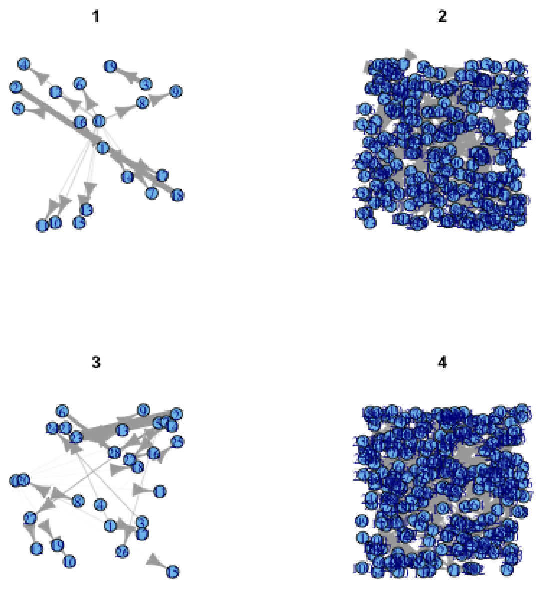

Figure 6.3: Activity diagram of nodes in network when changes are detected. Note that the thicker the arrow, the greater the volume of traffic across that connection.

of the four graphs refers to the flows (source and destination IPs) observed in the four bins which are flagged in Figure 6.2. Additionally, the widths of the edges represent the volume of traffic for those edges. A notable feature is that there are distinct types of anomaly, some involving few nodes, with others on the order of hundreds. It is worth noting that this router handles around 5000 nodes, so this is a significant reduction. Of course, much more refined analytics can be proposed, but such proposals must account for computational aspects, particularly data storage.

## 6.6 Discussion

In this chapter the AFF scheme has been extended from the univariate to the multivariate setting. Two formulations are proposed for an MVAFF framework, one using a single scalar $\overrightarrow{\lambda}$ and the other using an AFF for each component of the multivariate stream. The latter formulation is preferred, since it allows a scaling analogous to that for the univariate case that reduces the dependence on the step size $\eta$. However, when this framework is applied in a change detector, it is shown that there is still some dependence on $\eta$, unless $\eta$ is set to be very small.

Three methods of combining $p$-values are discussed which allow decision rules for multivariate change detection to be specified. Two of the methods assume that the component streams are independent, while the other takes the covariance into account.

An extensive simulation study is performed for the case when the streams are independent, and for when the streams are dependent. The forgetting factor methods are compared to a multivariate CUSUM scheme and a self-starting multivariate EWMA scheme (SSMEMWA). For independent streams, the forgetting factor methods are comparable to SSMEWMA, but for dependent streams SSMEWMA sometimes performs better than the forgetting factor schemes. However, it is not easy to know in advance which parameter settings for SSMEWMA will give the best performance, and SSMEWMA is more computationally complex than the forgetting factor methods.

Finally, the multivariate AFF scheme is applied to NetFlow data to monitor computer network traffic, where the scheme can be used to alert analysts to anomalous network behaviour.

The next chapter evaluates the performance of approximate methods for computing the weighted sum of chi-squared random variables. This will be needed in the context of detecting the change in the variance of a normally-distributed stream, the framework for which will be outlined in Chapter 8.

# Chapter 7

# Approximating the cdf of a weighted sum of chi-squared random variables

Previous chapters have focused on adaptive estimation and change detection for the mean of a data stream. Adaptive estimation of the variance is also possible, and will be explored in Chapter 8. In order to obtain a decision rule for detecting a change in the variance, we could start by assuming that the stream is i.i.d. normal, as was done for the mean. Chapter 8 then shows that, in this case, the forgetting factor variance can then be expressed as a weighted sum of chi-squared random variables. A decision rule then requires the computation of the cumulative distribution function (cdf) of a weighted sum of chi-squared random variables.

Unfortunately, there is no known closed form solution for the cdf of a weighted sum of chi-squared random variables. Numerous approximate methods have been proposed, and Section 7.1 gives a brief review. Section 7.2 discusses why many of these methods are unsuitable for use in a streaming data context. Section 7.3 reviews four moment-matching methods which are suitable for streaming data.

The best approximate method, for a streaming data context, would be one that is both accurate and computationally efficient. Section 7.4 describes how previous studies have analysed the accuracy of approximate methods for computing the cdf of a weighted sum of chi-squared random variables. It is then discussed why these analyses may be regarded as inadequate. Section 7.5 introduces a general framework for evaluating the accuracy of approximate methods for the cdf of weighted sums of *arbitrary* random variables. This framework is then used in Section 7.6 to evaluate the accuracy of the four moment-matching methods described in Section 7.3. Section 7.6.3 analyses their computationally efficiency. Finally, Section 7.7 makes a recommendation for which method should be preferred in a streaming data context.

## 7.1 The cdf of a weighted sum of chi-squared random variables

The cdf $F_{Q_N}$ of a positively-weighted sum of i.i.d. $\chi^2_1$ random variables $Q_N$,

$$Q_N = \sum_{i=1}^{N} d_i W_i^2, \qquad d_i > 0, \qquad W_i \sim \mathrm{N}(0,1), \qquad i = 1, 2, \ldots, N. \tag{7.1}$$

has no known closed-form solution. An approximation of $F_{Q_N}$ is used in goodness-of-fit tests [111, 142] and various other applications [e.g. 167, 75, 10, 34]. When speed of computation is not an issue, Imhof's method [74], which inverts the characteristic function numerically, should be the preferred choice. It can be considered exact [140, 80] since it provides error bounds and can be used to compute $F_{Q_N}(x)$, for some quantile value $x$, to within a desired precision. Other similar numerical methods such as Farebrother's method [47] could also be used, while others [137, 42, 41] lack the precision-bounding feature of Imhof's method. However, Imhof's method and Farebrother's method are both iterative, which affects their speed of computation, as shown in Section 7.6.3.

Perhaps the earliest approximate method, which has come to be known as the Satterthwaite-Welch method [156, 135, 46], involved matching the first two moments of $Q_N$ with the first two moments of a Gamma distribution. See [18, Sec. 3] for a discussion on the history of this method. The Hall-Buckley-Eagleson [59, 24] and Wood $F$ [159] methods match the first three moments of $Q_N$ to other distributions in a similar fashion. The Lindsay-Pilla-Basak method [94] matches the first $n$ moments of $Q_N$ to a mixture distribution. These four moment-matching methods are described in Section 7.3 and an R package implementing these methods is in preparation.

The Solomon-Stephens method [140] takes the Satterthwaite-Welch method a step further, by matching the first three moments of $Q_N$ to a random variable $aX^b$, where $X \sim \chi^2_1$. It is accurate in both the upper and lower tails, but requires the solution of two simultaneous non-linear equations, perhaps via an iterative method. An interesting method using Laguerre polynomials is described in [31], but is also iterative and requires the setting of certain control parameters.

While the methods discussed here have superseded those published previously (e.g. [118, 76]), a good review of older methods can be found in [80]. Although not considered here, a review of the current state-of-the art for weighted sums of *non-central* chi-squared random variables can be found in [43].

## 7.2 Approximations in a streaming data context

If we wished to simply compute a single evaluation of $F_{Q_N}$, for some vector of coefficients $\overrightarrow{d} = (d_1, d_2, \ldots, d_N)$, then we have already described a plethora of methods from which to choose. Amongst these, since Imhof's method is essentially exact it would probably be the preferred choice. There are, however, situations when Imhof's method might not be suitable, and streaming data analysis provides a striking example. For instance, and motivated by the needs of Chapter 8, one might wish to compute $F_{Q_N}(x)$, for $Q_N$ defined in Equation (7.1), and then soon afterwards compute $F_{Q_{N+1}}(x')$, where

$$Q_{N+1} = Q_N + d_{N+1}W_{N+1}.$$

Imhof's method requires the complete vector of weights $\overrightarrow{d}$ in order to compute $F_{Q_{N+1}}(x')$, but in a streaming data context (discussed in the next paragraph) $N$ might be very large, and so storing the entire coefficient vector $(d_1, \ldots, d_N, d_{N+1})$ would be undesirable. Finally, Imhof's method is also iterative, since it runs until a specified precision is obtained. This is also unappealing, since iterative methods have the potential to be slow and computationally expensive.

Streaming data algorithms (e.g. [53, 13]) require methods that are both fast and only require a small, fixed number of parameters to be stored. Amongst the methods discussed above, the moment-matching methods of Satterthwaite-Welch, Hall-Buckley-Eagleson, Wood and Lindsay-Pilla-Basak are the only options that meet these criteria and are described in Section 7.3. The first three of these methods only require a single evaluation of a particular cdf and the storage of a fixed number of parameters that can be easily sequentially updated. The Lindsay-Pilla-Basak method is more computationally intensive, but has the potential to give more accurate results by matching higher-order moments. There are other approximate methods besides these four (e.g. [140]), but they all have shortcomings (e.g. require too much memory, too expensive to compute) that would render them unsuitable for streaming data applications.

## 7.3 Efficient approximate moment-matching methods

As the name suggests, these methods involve matching the moments of $Q_N$ to those of another distribution, and using that distribution's cdf to approximate $F_{Q_N}$. In order to do this, the moments of $Q_N$ need to be computed. However, instead of computing the moments directly, it is easier to first compute the cumulants of $Q_N$ and then obtain the moments from the cumulants. In fact, the first three methods described below directly use the computed cumulants, and do not require computation of the moments. Appendix A.2.2 reviews the definitions of cumulants and moments.

### 7.3.1 Computing cumulants and moments

The cumulants of $Q_N$, a weighted sum of i.i.d. $\chi_1^2$ random variables as in Equation (7.1), are denoted by $\kappa_r(Q_N)$ and can be computed using the formula

$$\kappa_r(Q_N) = 2^{r-1}(r-1)! \sum_{i=1}^{N} (d_i)^r, \qquad r = 1, 2, \ldots. \tag{7.2}$$

where $\overrightarrow{d} = (d_1, d_2, \ldots, d_N)$ are the weighting coefficients. This can easily be shown using the properties of cumulants and recalling (see Appendix A.2.5) that for a $\chi_1^2$ random variable $X$, $\kappa_r(X) = 2^{r-1}(r-1)!$. In a sequential context, when $Q_N$ becomes $Q_{N+1}$, the cumulants can be easily updated by

$$\kappa_r(Q_{N+1}) = \kappa_r(Q_N) + 2^{r-1}(r-1)! \cdot (d_{N+1})^r.$$

For the remainder of this chapter we shall only be concerned with $Q_N$, and so shall write $\kappa_r = \kappa_r(Q_N)$. The moments of $Q_N$, denoted $m_r = m_r(Q_N)$, can be computed from the cumulants using $m_1 = \kappa_1$ and

$$m_r = \kappa_r + \sum_{i=1}^{r-1} \binom{r-1}{i-1} \kappa_i m_{r-i}, \qquad r = 2, 3, \ldots. \tag{7.3}$$

Since the first three methods described below only require the first two or three cumulants of $Q_N$, these are explicitly provided here:

$$\kappa_1 = \sum_{i=1}^{N} d_i, \qquad \kappa_2 = 2\sum_{i=1}^{N} (d_i)^2, \qquad \kappa_3 = 8\sum_{i=1}^{N} (d_i)^3.$$

### 7.3.2 Satterthwaite-Welch approximation

Equating the first two moments of $Q_N$ with a $\Gamma(\widehat{k}, \widehat{\theta})$ variable, yields

$$\widehat{k} = \kappa_1^2/\kappa_2, \qquad \widehat{\theta} = \kappa_2/\kappa_1.$$

If we use $F_{\Gamma(k,\theta)}$ to denote the cdf of a $\Gamma(k, \theta)$ distribution, then the Satterthwaite-Welch approximation uses $F_{\Gamma(\widehat{k},\widehat{\theta})}$ to approximate $F_{Q_N}$. In the references (e.g. [18]), the $\Gamma(k, \theta)$ distribution is often written as a scaled $\chi_1^2$ distribution.

### 7.3.3 Hall-Buckley-Eagleson approximation

We provide a brief outline of the method which is fully described in [24]. First, $Q'_N$ is used to denote $Q_N$ normalised as in:

$$Q'_N = \frac{Q_N - \mathrm{E}[Q_N]}{\sqrt{\mathrm{Var}[Q_N]}} = \kappa_2^{-1/2}(Q_N - \kappa_1).$$

Second, if $\nu$ is defined as

$$\nu = 8\kappa_2^3/\kappa_3^2,$$

and $X_\nu \sim \chi^2_\nu \equiv \Gamma(\nu/2, 2)$, then it can be shown that $Q'_N$ and $(X_\nu - \nu)/\sqrt{2\nu}$ have the same first three central moments. If $Y \sim Q_N$ and $y$ is an observation of $Y$, the Hall-Buckley-Eagleson approximation of $F_{Q_N}(y)$ is obtained by

$$F_{\Gamma(\nu/2,2)}\left(\sqrt{2\nu}\cdot\left[\kappa_2^{-1/2}(y-\kappa_1)\right]+\nu\right).$$

### 7.3.4 Wood $F$ approximation

Wood's $F$ method [159] matches the first three moments of $Q_N$ with another distribution that has a probability density function (pdf) of the form

$$f(x|\alpha_1,\alpha_2,\beta) = \frac{\beta^{\alpha_2}x^{\alpha_1-1}(\beta+x)}{\mathrm{B}(\alpha_1,\alpha_2)}, \qquad \mathrm{B}(\alpha_1,\alpha_2) = \frac{\Gamma(\alpha_1)\Gamma(\alpha_2)}{\Gamma(\alpha_1+\alpha_2)}, \tag{7.4}$$

where $\mathrm{B}(\alpha_1, \alpha_2)$ is the beta function. Although in [159] it is referred to as an $F$ distribution, the density in Equation (7.4) can be better described as that of a G3F or *corrected F* distribution [120, 79]. The parameters $\alpha_1, \alpha_2, \beta$ can be defined in terms of the cumulants $\kappa_1, \kappa_2, \kappa_3$ computed in Equation (7.2) above (e.g. using Gröbner bases [23]; see Appendix A.5):

$$\begin{aligned}
r_1 &= 4\kappa_1\kappa_2^2 + \kappa_3(\kappa_2 - \kappa_1^2), \qquad r_2 = \kappa_1\kappa_3 - 2\kappa_2^2\\
\alpha_1 &= 2\kappa_1\left(\kappa_1\kappa_3 + \kappa_1^2\kappa_2 - \kappa_2^2\right)/r_1\\
\alpha_2 &= 3 + 2\kappa_2\left(\kappa_2 + \kappa_1^2\right)/r_2\\
\beta &= r_1/r_2
\end{aligned} \tag{7.5}$$

It is noted in [159] that if $X$ is distributed according to the density in Equation (7.4), then

$$\frac{\alpha_2}{\alpha_1\beta}X \sim F(2\alpha_1, 2\alpha_2),$$

where $F(2\alpha_1, 2\alpha_2)$ is a standard $F$-distribution with parameters $2\alpha_1$ and $2\alpha_2$. Therefore, if $Y \sim Q_N$, and $y$ is an observation of $Y$, the Wood $F$ approximation of $F_{Q_N}(y)$ is obtained by

$$F_{F(2\alpha_1,2\alpha_2)}\left(\frac{\alpha_2}{\alpha_1\beta}y\right).$$

This approximation can be used as long as both $r_1, r_2 > 0$, which is guaranteed in many cases [159]. When either $r_1 = 0$ or $r_2 = 0$ (it is proved in [159] that neither can be negative), then [159] recommends using either the Satterthwaite-Welch approximation, or another two-moment approximation.

### 7.3.5 Lindsay-Pilla-Basak approximation

This method described in [94] approximates $F_{Q_N}$ using $F_{\widetilde{Q}_N}$, a finite mixture of $n$ Gamma cdfs $F_{\Gamma(k,\theta_i)}$,

$$F_{\widetilde{Q}_N} = \sum_{i=1}^{n} \pi_i F_{\Gamma(k,\theta_i)}, \tag{7.6}$$

where each $\pi_i \geq 0$ and $\sum_i \pi_i = 1$, and the $2n+1$ parameters $k, \theta_1, \theta_2, \ldots, \theta_n, \pi_1, \pi_2, \ldots, \pi_n$ are to be determined. These parameters are computed by following a sequence of steps that make use of results concerning moment matrices (Appendix II in [152]). The sequence in [94] is complicated, so we extract the main steps here (without proofs). The first step is to compute the first $2n$ cumulants $\kappa_1, \kappa_2, \ldots, \kappa_{2n}$ of $Q_N$ using Equation (7.2), and then use the recursive formula in Equation (7.3) to compute the the first $2n$ moments $m_1, m_2, \ldots, m_{2n}$ of $Q_N$. The second step is to define, for a variable $\alpha$, the functions $\delta_r(\alpha)$ as

$$\delta_r(\alpha) = \frac{m_r}{\prod_{i=1}^{r}(1 + (i-1)\alpha)}, \qquad r = 1, 2, \ldots, 2n. \tag{7.7}$$

and $\delta_0(\alpha) = 1$. These functions are then used to create the $(r+1) \times (r+1)$ pseudo-moment matrices $\Delta_r(\alpha)$, defined as

$$\Delta_r(\alpha) = \{\delta_{i+j}(\alpha)\}_{\substack{i=0,1,\ldots r\\ j=0,1,\ldots r}}, \qquad r = 1, 2, \ldots, n. \tag{7.8}$$

For example,

$$\Delta_2(\alpha) = \begin{pmatrix} \delta_0(\alpha) & \delta_1(\alpha) & \delta_2(\alpha) \\ \delta_1(\alpha) & \delta_2(\alpha) & \delta_3(\alpha) \\ \delta_2(\alpha) & \delta_3(\alpha) & \delta_4(\alpha) \end{pmatrix} = \begin{pmatrix} 1 & m_1 & \frac{m_2}{(1+\alpha)} \\ m_1 & \frac{m_2}{(1+\alpha)} & \frac{m_3}{(1+\alpha)(1+2\alpha)} \\ \frac{m_2}{(1+\alpha)} & \frac{m_3}{(1+\alpha)(1+2\alpha)} & \frac{m_4}{(1+\alpha)(1+2\alpha)(1+3\alpha)} \end{pmatrix}.$$

The third step is to find certain roots $\widetilde{\lambda}_1, \widetilde{\lambda}_2, \ldots \widetilde{\lambda}_n$ such that

$$\det \Delta_r(\widetilde{\lambda}_r) = 0, \qquad r = 1, 2, \ldots, n. \tag{7.9}$$

For $r = 1$, there is a unique positive root $\widetilde{\lambda}_1 = m_2/(m_1^2) - 1$. For $r > 1$, one can use a bisection method to solve for the root $\widetilde{\lambda}_r \in [0, \widetilde{\lambda}_{r-1})$ of the equation $\det \Delta_r(\alpha) = 0$. Eventually, $\widetilde{\lambda}_n$ is obtained. The fourth step is to define the matrix $M_n(\widetilde{\lambda}_n, t)$,

$$M_n(\widetilde{\lambda}_n, t) = \begin{pmatrix} 1 & \delta_1(\widetilde{\lambda}_n) & \cdots & \delta_{n-1}(\widetilde{\lambda}_n) & 1 \\ \delta_1(\widetilde{\lambda}_n) & \delta_2(\widetilde{\lambda}_n) & \cdots & \delta_n(\widetilde{\lambda}_n) & t \\ \delta_2(\widetilde{\lambda}_n) & \delta_3(\widetilde{\lambda}_n) & \cdots & \delta_{n+1}(\widetilde{\lambda}_n) & t^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \delta_n(\widetilde{\lambda}_n) & \delta_{n+1}(\widetilde{\lambda}_n) & \cdots & \delta_{2n-1}(\widetilde{\lambda}_n) & t^n \end{pmatrix}. \tag{7.10}$$

Note that $M_n(\widetilde{\lambda}_n, t)$ is the same as $\Delta_n(\widetilde{\lambda}_n)$ but with the last column replaced by $(1, t, \ldots, t^n)'$. This matrix is used to compute the $n$th degree polynomial $S_n(\lambda, t)$,

$$S_n(\lambda, t) = \det M_n(\widetilde{\lambda}_n, t) = \sum_{j=0}^{n} c_j t^j, \qquad c_j \in \mathbb{R}, j = 0, 1, \ldots, n \tag{7.11}$$

In order to obtain the value of the coefficient $c_j$, one can replace the last column of $M_n(\widetilde{\lambda}_n, t)$ (the powers of $t$), with the basis vector $e_{j+1}$ (the $(j+1)$th component equals one, all others are zero), and compute the determinant of this modified matrix. With the coefficients computed, the $n$ roots of $S_n(\lambda, t) = 0$, denoted $\mu_1, \mu_2, \ldots, \mu_n$, can be found (the roots are real and distinct [152, Appendix II.4]). The fifth step is to use these roots $\mu_i$ to solve the system of linear equations

$$\begin{pmatrix} 1 & 1 & \cdots & 1 \\ \mu_1 & \mu_2 & \cdots & \mu_n \\ \vdots & \vdots & \vdots & \vdots \\ \mu_1^{n-1} & \mu_2^{n-1} & \cdots & \mu_n^{n-1} \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{pmatrix} = \begin{pmatrix} 1 \\ \delta_1(\widetilde{\lambda}_n) \\ \vdots \\ \delta_{n-1}(\widetilde{\lambda}_n) \end{pmatrix} \tag{7.12}$$

to compute the mixture proportions $\pi_1, \pi_2, \ldots, \pi_n$. Since the matrix on the left of Equation (7.12) is a Vandermonde matrix, it is non-singular, and so this system of linear equations has a unique solution. Finally, we define $k = (\widetilde{\lambda}_n)^{-1}$ and $\theta_i = \widetilde{\lambda}_n \cdot \mu_i$, for $i = 1, 2, \ldots, n$, and now can compute the approximate cdf $F_{\widetilde{Q}_N}$ in Equation (7.6). Note that the Lindsay-Pilla-Basak method agrees with the Satterthwaite-Welch method for $n = 1$.

It should be remarked that [127] also attempts to obtain an approximation using a method of mixtures, but by computing the characteristic function rather than using the method of moments.

## 7.4 The evaluation of approximate methods for computing $F_{Q_N}$ in the literature

In previous work on approximations for computing the cdf $F_{Q_N}$ of weighted sums of chi-squared random variables $Q_N$ [74, 140, 159, 94, 31], it was common to estimate the performance of an approximate method by demonstrating its accuracy for a selected sample of $M$ distributions $Q_{N,\vec{d}_1}, Q_{N,\vec{d}_2}, \ldots, Q_{N,\vec{d}_M}$, where

$$Q_{N,\vec{d}_k} = \sum_{i=1}^{N} d_{i,k} W_i^2 \qquad W_i \sim \mathrm{N}(0,1), \qquad k = 1, 2, \ldots, M,$$

and $\vec{d}_k = (d_{1,k}, d_{2,k}, \ldots, d_{N,k})$. Recall that the cdf of a random variable $X$ is defined by

$$F_X(x) = \Pr(X \leq x).$$

In this thesis, values $x$ in the domain of the cdf $F_X$ will be called *quantile values*, and values $F_X(x)$ will be called *probability values*. For each $Q_{N,\vec{d}_k}$ the quantile values $x_{j,k}$ are found such that for $j = 1, 2, \ldots, L$,

$$F_{Q_{N,\vec{d}_k}}(x_{j,k}) = p_j, \tag{7.13}$$

for a specific set of probability values $p_j$. Then a table of errors $\epsilon_{j,k}$, where

$$\epsilon_{j,k} = |G(x_j) - F_{Q_{N,\vec{d}_k}}(x_j)|, \tag{7.14}$$

is presented for one or more approximate methods, where $G$ is the cdf produced by the approximate method. According to the literature, the method with the smallest set of errors is then considered to be the best approximate method.

This may seem to be a reasonable approach, but the execution in previous works leaves something to be desired. In [74, 140, 159, 94, 31] each analysis only considers a selection of between $M = 8$ and $M = 18$ distributions $Q_N$ for varying coefficients and numbers of terms. Results established for an approximation procedure based on the analysis of such a small selection should be viewed with caution. So, while previous works may have established the accuracy for the particular selections considered, those results cannot reasonably be assumed to hold for all possible $Q_N$. Moreover, previous works only considered $Q_N$ with fewer than $N = 10$ terms, so it is natural to wonder how approximate methods perform for distributions $Q_N$ with significantly larger $N$.

There is a possible explanation for why previous works only consider a limited selection of distributions $Q_N$ in their analyses. When these approximate methods were first considered in the 1950s and 1960s (e.g. [18, 74]), calculating the probability values may have been difficult, especially with computing in its infancy. Therefore, only a limited table of results was produced. When later methods in the 1970s and 1980s (e.g. [140, 159]) were developed, it would have been natural to use the performance analysis of earlier methods as the benchmark, and so a table of errors $\epsilon_{j,k}$ was again compiled for a small (in some cases the same) sample of distributions. Unfortunately, this method of evaluating performance has continued unchanged (e.g. [94, 31]), even though computers able to complete a much more thorough analysis are now readily available. In Section 7.5 we outline such an analysis, which will seem natural following the discussion in this section.

It should be mentioned that while we shall use Farebrother's method in combination with a bisection procedure to compute the exact quantile values (i.e. Equation (7.13)) in Section 7.6, it was not indicated in previous works how the exact quantile values were obtained for performance calculations.

## 7.5 Evaluating the performance of an approximate method for a cdf of a weighted sum of arbitrary random variables

This section discusses the issue of evaluating the performance of approximation methods for the cdf of a weighted sum of random variables. This procedure is then used in Section 7.6 to analyse the performance of approximate methods for the cdf of a weighted sum of chi-squared random variables. In this section $R_N$ is a weighted sum of random variables from an *arbitrary* distribution (not necessarily chi-squared). It is assumed that a method exists for computing the true probability value $F_{R_N}(x)$ for quantile value $x$, to arbitrary accuracy. However, the method may be too computationally or memory intensive for routine application.

### 7.5.1 Performance of an approximate method for a particular distribution

Suppose a method provides approximate probability values $G(x)$ for a weighted sum of random variables $R_N$. Suppose further that we wish to determine how close $G$ is to the true cdf $F_{R_N}$, for a particular distribution $R_{N,\overrightarrow{d}}$ with weights $\overrightarrow{d} = (d_1, d_2, \ldots, d_N)$. For a set of probability values

$$\{p_1, p_2, \ldots, p_L\},$$

suppose that the "exact" quantile values

$$\{x_1, x_2, \ldots, x_L\}$$

can be computed to an arbitrary precision $\xi$, perhaps at a practically unacceptable computational cost, so that

$$|F_{R_N}(x_j) - p_j| < \xi, \qquad j = 1, 2, \ldots, L, \qquad \xi \ll 1.$$

The errors of the approximate method $G$, denoted by $\epsilon_j$, are then defined as

$$\epsilon_j = |G(x_j) - F_{R_N}(x_j)|.$$

The smaller the $\epsilon_j$, the better that $G$ approximates $F_{R_N}$ for the probability values $p_j$. By a simple application of the triangle inequality,

$$|G(x_j) - p_j| < \epsilon_j + \xi$$

is obtained. Therefore, if the $x_j$ can be computed to ensure $\xi \ll \epsilon_j$ for all $j$, it is then only necessary to look at the values $|G(x_j) - p_j|$ to obtain a good approximation for $\epsilon_j$.

### 7.5.2 Estimating the accuracy of an approximate method for a specific $N$

The first step to more comprehensively evaluating the performance of an approximate method for distributions with $N$ terms is to randomly generate a large sample of $M$ coefficient vectors $\overrightarrow{d}_k = (d_{1,k}, d_{2,k}, \ldots, d_{N,k})$, where

$$d_{1,k}, d_{2,k}, \ldots, d_{N,k} \sim D, \qquad k = 1, 2, \ldots, M, \tag{7.15}$$

for some distribution $D$, so that $F_{R_{N,\overrightarrow{d}_k}}$ is the cdf of

$$R_{N,\overrightarrow{d}_k} = \sum_{i=1}^{N} d_{i,k} Y_i, \qquad Y_i \sim Y, \qquad k = 1, 2, \ldots, M,$$

for some distribution $Y$. The next step is to select a wide range of probability values $\{p_1, p_2, \ldots, p_L\}$, and then to compute the quantile values

$$\{x_{1,k}, x_{2,k}, \ldots, x_{L,k}\}, \qquad k = 1, 2, \ldots, M,$$

so that for some precision $\xi$,

$$|F_{R_{N,\overrightarrow{d}_k}}(x_{j,k}) - p_j| < \xi, \qquad j = 1, 2, \ldots, L, \qquad k = 1, 2, \ldots, M. \tag{7.16}$$

Finally, the errors $\epsilon_{j,k}$ are computed as

$$\epsilon_{j,k} = |G(x_j) - F_{R_{N,\overrightarrow{d}_k}}(x_j)|, \qquad j = 1, 2, \ldots, L, \qquad k = 1, 2, \ldots, M.$$

The set of errors for probability value $p_j$ is defined as

$$E_j = \{\epsilon_{j,k} | k = 1, 2, \ldots, M\}.$$

While it would now be easy to compute $\max E_j$ and declare this to be a reasonable upper bound for the error when computing $p_j$, provided that $M$ is large, the following procedure is preferable because it establishes a probabilistic result. Define $\bar{\epsilon}_j$ to be the sample mean, $s^2_{\epsilon_j}$ the sample variance, and $q^2_{\epsilon_j}$ the scaled sample variance of $E_j$ by the equations:

$$\begin{aligned}
\bar{\epsilon}_j &= \frac{1}{M}\sum_{k=1}^{M} \epsilon_{j,k}, \\
s^2_{\epsilon_j} &= \frac{1}{M-1}\sum_{k=1}^{M} \left[\epsilon_{j,k} - \bar{\epsilon}_j\right]^2, \\
q^2_{\epsilon_j} &= \left(\frac{M+1}{M}\right) s^2_{\epsilon_j}.
\end{aligned}$$

Suppose that $\epsilon^*_j$ is the error for $F_{R_{N,\overrightarrow{d}^*}}$, with coefficient vector $\overrightarrow{d}^*$ generated as in Equation (7.15). If we assume that the errors in $E_j$ are i.i.d. according to some distribution, then Chebyshev's inequality with the sample mean and variance [136] gives us, for any $\delta > 0$,

$$\Pr\left(|\epsilon^*_j - \bar{\epsilon}_j| > \delta q_{\epsilon_j}\right) \leq \frac{1}{\delta^2} + \frac{1}{M}\left(1 - \frac{1}{\delta^2}\right). \tag{7.17}$$

If we set the the right-hand side of Equation (7.17) to be

$$\alpha_{\delta,M} = \frac{1}{\delta^2} + \frac{1}{M}\left(1 - \frac{1}{\delta^2}\right), \tag{7.18}$$

then Equation (7.17) implies

$$\begin{aligned}
&\Pr\left(\epsilon^*_j > \bar{\epsilon}_j + \delta q_{\epsilon_j}\right) \leq \alpha_{\delta,M}, \\
\Rightarrow &\Pr\left(\epsilon^*_j \leq \bar{\epsilon}_j + \delta q_{\epsilon_j}\right) > 1 - \alpha_{\delta,M}.
\end{aligned} \tag{7.19}$$

Then $\bar{\epsilon}_j + \delta q_{\epsilon_j}$ provides an upper bound for $100(1 - \alpha_{\delta,M})\%$ of all possible errors obtained when computing $p_j$ using the approximate method. In other words, the probability that the error exceeds the upper bound is less than $\alpha_{\delta,M}$. For example, when $\delta = 10$ and $M = 10000$ then $\alpha_{\delta,M} \approx 0.01$, or $\delta = 32$ and $M = 10000$ gives $\alpha_{\delta,M} \approx 0.001$.

The same procedure could be followed to obtain a bound for the error of computing $p_j$ for every $p_j \in \{p_1, p_2, \dots, p_L\}$, and so an estimate of the error for an approximate method of computing probability values for distributions $Q_N$ is obtained, for a particular $N$.

The assumption that the errors in $E_j$ are i.i.d. may seem restrictive, but in fact the errors need only be weakly exchangeable. Finally, although [136] gives a slightly sharper bound for the inequality in Equation (7.17), its expression is far more complicated and does not significantly change the bound for our purposes here.

## 7.6 Results

A simulation is performed by computing $M = 10000$ sets of coefficients $d_{i,k} \sim U(0,1)$ for cases where $N = 10, 20, 50, 100$, and then computing the quantile values $x_{j,k}$ corresponding to probability values

$$p_j \in \{0.001, 0.05, 0.01, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99, 0.999\}\,.$$

Assuming that the coefficients are sampled from $U(0,1)$ is not particularly restrictive; if a particular application uses coefficients that are known to be bounded, they can be rescaled to the range $(0,1)$. Farebrother's method is used to ensure the quantiles are accurate to $\xi = 10^{-8}$ as in Equation (7.16). Imhof's method could also have used, but the implementation of Farebrother's method in the R package CompQuadForm [89] appears to allow a greater precision to be specified. The analysis is then performed using $\delta = 32$ to obtain an upper bound with confidence $\alpha_{\delta,M} \approx 0.001$ (see Equation (7.18)). The accuracy of each of the four moment-matching methods in Section 7.3 is computed, and the methods are compared side by side in Section 7.6.1. The Lindsay-Pilla-Basak method is computed for $n = 4$ (that is for the first four moments), and so will be abbreviated to LPB4. In Section 7.6.3 we then investigate the relative speeds of each method. Note that none of the sampled coefficient vectors $\mathbf{d}_k$ yielded degenerate cases (as mentioned in Section 7.3.4) for the Wood F approximation.

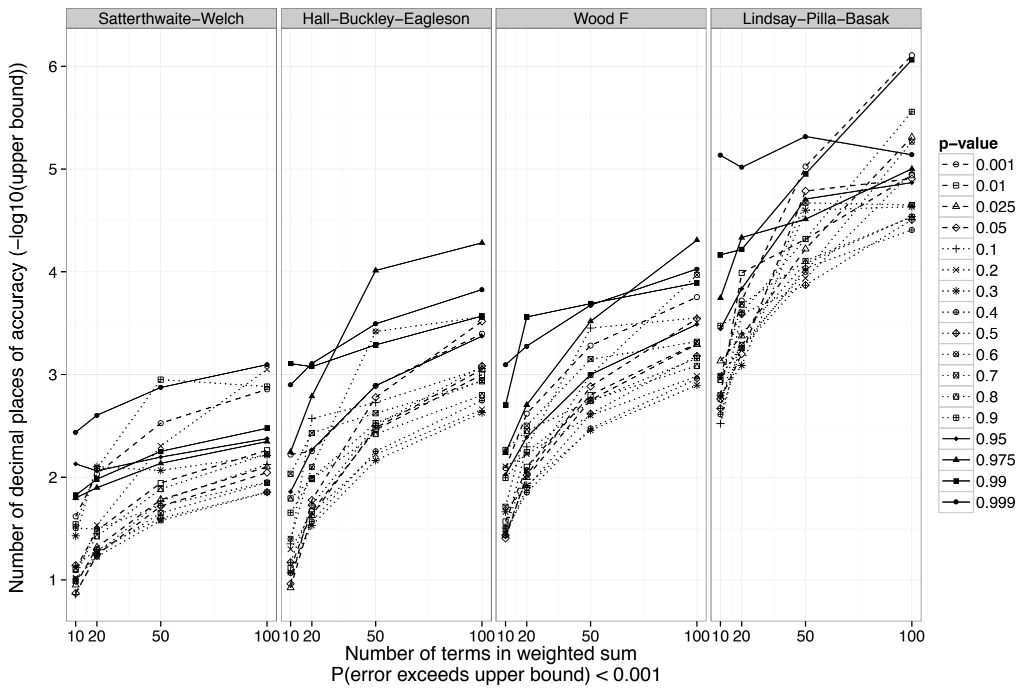

Figure 7.1: Error of Satterthwaite-Welch, Hall-Buckley-Eagleson, Wood F and Lindsay-Pilla-Basak approximations for varying number of terms $N$, grouped by method. The coefficients are distributed according to $U[0, 1]$.

### 7.6.1 Accuracy

The accuracy of the Satterthwaite-Welch (SW), Hall-Buckley-Eagleson (HBE), Wood F (WF) and Lindsay-Pilla-Basak with $n = 4$ (LPB4) approximate methods is shown in Figures 7.1 and 7.2, for a wide selection of probability values and values of $N$. The horizontal axis indicates the value of $N$, while the vertical axis shows *number of digits of accuracy*; the value shown is $-\log_{10}(\bar{\epsilon}_j + \delta q_{\epsilon_j})$ (see Equation (7.19)). Figure 7.1 groups the values by method, while Figure 7.2 groups the values by probability value. For the purposes of discussion below, let us define the lower tail to be the probability values $\{0.001, 0.01, 0.025, 0.05\}$ and the upper tail to be $\{0.95, 0.975, 0.99, 0.999\}$. Values that are neither in the lower nor upper tails will be referred to as middle probability values.

Figure 7.1 illustrates several points. The first feature of interest is that the methods generally increase in accuracy as $N$ increases. There are a couple of exceptions (e.g. $p_j =$

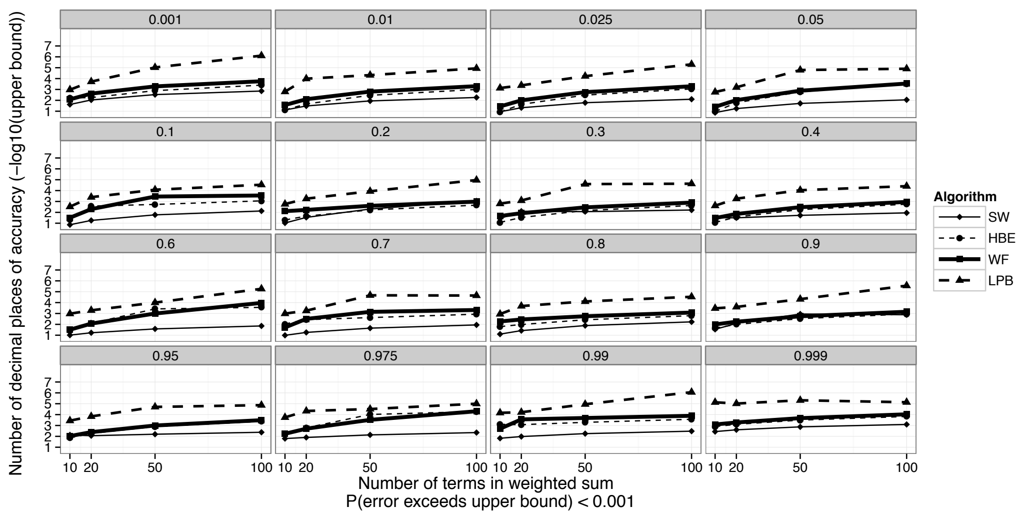

Figure 7.2: Error of Satterthwaite-Welch, Hall-Buckley-Eagleson, Wood F and Lindsay-Pilla-Basak approximations for varying number of terms $N$, grouped by probability value. The coefficients are distributed according to $U[0, 1]$.

$0.999$ for the LPB4), but any decreases are minor. This seems to suggest a trend which would continue for $N \geq 100$. Following this observation, if method $A$ has number of digits of accuracy $y$ for number of terms $N'$, we shall say that method $A$ is accurate to $y$ decimal places for $N \geq N'$. As far as we are aware, this observation that the accuracy of these approximate methods generally increases, as the value of $N$ increases, has not been noted before and is not apparent or implied from the construction of the methods. As already mentioned, previous analyses only focused on distributions $Q_N$ for a limited range of $N$.

If the results for each individual method are now examined, it can be seen that SW is accurate in the upper and lower tails to at least two decimal places for $N \geq 100$. The HBE method is accurate to two decimal places for all $p_j$ for $N \geq 50$. The WF method is also accurate to two decimal places for $p_j$ for $N \geq 50$, and is accurate in the upper tail to 3 decimal places for $N \geq 50$. The LPB4 method is accurate to 4 decimal places for almost all probability values (only exceptions are a few middle probability values) for $N \geq 50$, and has close to 5 digits of accuracy for the upper and lower tails for $N \geq 100$.

Figure 7.2 shows that over the different probability values, SW is the least accurate,

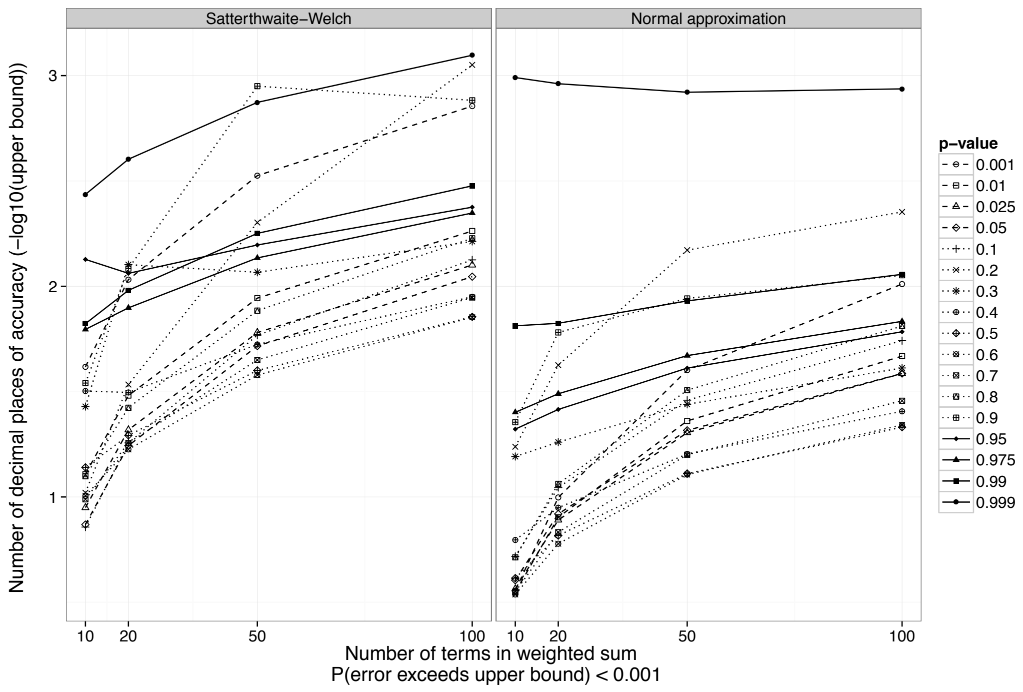

Figure 7.3: Error of the Normal approximation, compared to the Satterthwaite-Welch approximation, grouped by method

while LPB4 is clearly the most accurate, and WF and HBE appear to be essentially matched, although for most probability values WF has a slightly better accuracy than HBE (one exception is for $p_j = 0.975$ and $N = 50$).

### 7.6.2 Comparison to the normal approximation

Although the normal approximation is not considered to be as good as the four approximations considered above, it is interesting to investigate how it compares to SW, the simplest of the approximations above.

The normal approximation is computed in a similar manner to SW. Equating the first two moments of $Q_N$ with a N$(\widehat{\mu}, \widehat{\sigma}^2)$ variable, yields

$$\widehat{\mu} = \kappa_1, \qquad \widehat{\sigma} = \sqrt{\kappa_2}$$

| Method | Time | Relative speed (to HBE) |
|---|---:|---:|
| Normal | 1.18 | 0.45 |
| SW | 1.56 | 0.60 |
| HBE | 2.62 | 1.00 |
| WF | 3.14 | 1.19 |
| Imhof | 110.18 | 42.05 |
| LPB4 | 892.62 | 340.69 |
| Farebrother | 28793.98 | 10990.07 |

Table 7.1: The time taken (in seconds) for each method to compute $M = 17 \times 10000$ probability values for $Q_N$ with $N = 100$.

following the definition of the cumulants $\kappa_1$ and $\kappa_2$ in Section 7.3.1. Figure 7.3 shows that SW appears to be one decimal place more accurate than the normal approximation. The only exception is for $p_j = 0.999$, for which probability value the two methods appear to have similar accuracy. Even though both methods are two-moment approximations, and the computational complexity is virtually the same, SW's use of a Gamma cdf appears to provide a significant increase in accuracy over the normal approximation.

### 7.6.3 Speed of computation

Table 7.1 shows that while the SW, HBE and WF methods have similar speeds (of the same order), LPB4 is significantly slower. This could be due to the iterative methods needed in steps 3 and 4 of the algorithm (as described in Section 7.3.5) and the matrix algebra in several steps. Besides the matrix operations, the LPB method needs to employ root-finding algorithms (which can be very efficient, but are still iterative). For comparison purposes, the speeds of the normal approximation, Imhof's method and Farebrother's method have also been included. The normal approximation is slightly faster than SW, but is much less accurate. Surprisingly, Imhof's method is faster than LPB4, but is still over 40 times slower than HBE. LPB4 is over 300 times slower than HBE. Farebrother's method is much slower than any of the methods.

The four algorithms (SW, HBE, WF, LPB) and the normal approximation were written in R, while Imhof's method and Farebrother's are implemented in the R package *CompQuadForm* [89]. The speed test was done on an Apple iMac with an Intel Core i5

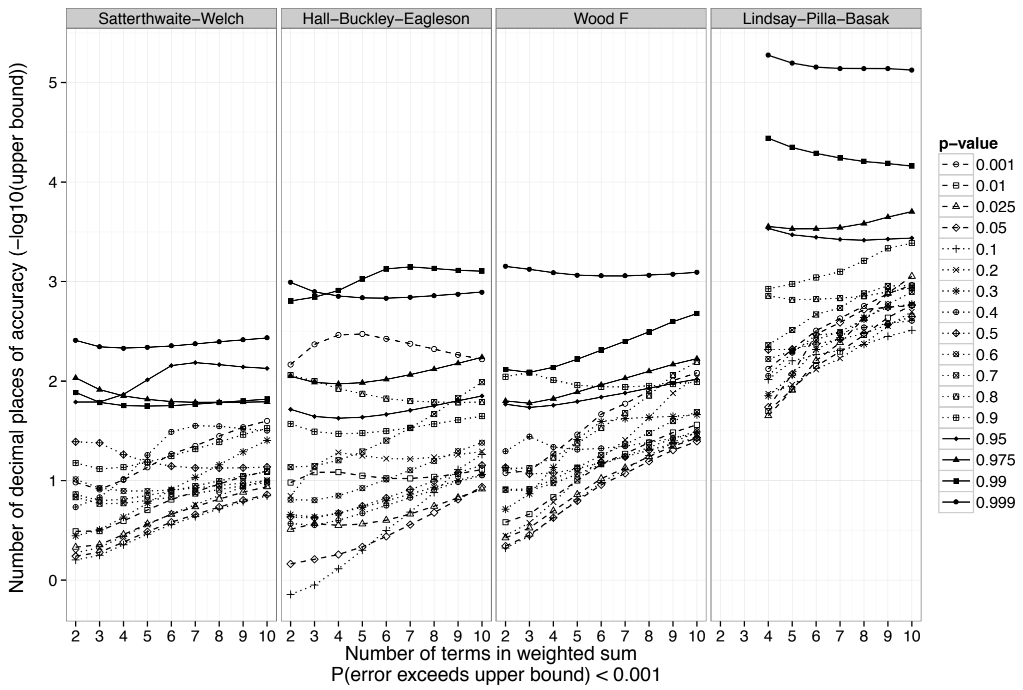

Figure 7.4: Error of Satterthwaite-Welch, Hall-Buckley-Eagleson, Wood F and Lindsay-Pilla-Basak approximations for a small number of terms $N$, grouped by method. Note that results are not provided for LPB for $N = 2, 3$.

(3.2 GHz) processor (4 cores) and 8 GB of RAM.

### 7.6.4 Weighted sums with a small number of terms

It is worth investigating the accuracy of these methods for the cases where $N \in \{2, 3, ..., 10\}$. The results of this investigation are shown in Figure 7.4, and show that SW, HBE and WF will generally give between 0 and 2 digits of accuracy, while LPB4 generally gives at least 2 digits of accuracy. These results suggest that when $N < 10$, these methods should be used with caution. Note that for $N = 2, 3$ there are choices of coefficient vector $\mathbf{d}_k$ which result in the LPB4 method not being able to provide an approximation (fails to find roots $\widetilde{\lambda}_r$ for Equation (7.9)), so values for $N = 2, 3$ for LPB4 are omitted.

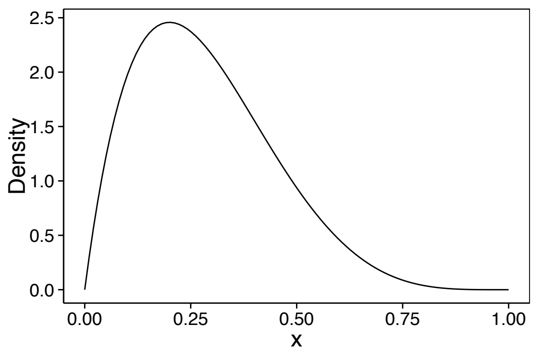

Figure 7.5: The probability density function of a Beta$(2, 5)$ random variable.

### 7.6.5 Results for another choice of coefficients

It is natural to wonder if the distribution of the coefficients $\overrightarrow{d}$ plays a role in the accuracy of the methods. Let us consider the case when coefficients $d_{i,k}$ are distributed according to a beta distribution, e.g. $d_{i,k} \sim$ Beta$(2, 5)$, which has pdf shown in Figure 7.5. This distribution was chosen because the coefficients of $Q_{N,\overrightarrow{d}}$ may follow a similar distribution if these methods were applied in a forgetting factor framework where each $d_{i_k} = \lambda^{N-i}/w_{N,\lambda}$. If the experiment were re-run as above but with coefficients distributed according to Beta$(2, 5)$ (instead of $U[0, 1]$), the resulting plots are very similar to Figures 7.1 and 7.2. Therefore, the choice of coefficients does not seem to be too critical to our results.

## 7.7 The preferred approximate method in a streaming data context

While Imhof's method is essentially exact, there are situations in which, as discussed in Section 7.2, it is necessary to (a) not store all the coefficients of $Q_N$, and (b) have efficient computation. In such situations, moment-matching methods such as Satterthwaite-Welch, Hall-Buckley-Eagleson, Wood F and Lindsay-Pilla-Basak may be very useful.

Choosing between these methods is not a simple matter of choosing the most accurate. One also needs to consider the speed of computation, and, to a lesser extent, the ease of implementation. While Figures 7.1 and 7.2 show the Lindsay-Pilla-Basak method to be extremely accurate, it is also significantly slower to compute (Table 7.1, Section 7.6.3) and laborious to implement (Section 7.3.5). If it is not essential to have four decimal places of accuracy, other methods could be used.

The Satterthwaite-Welch method is easy to implement, but two decimal places of accuracy (achieved in both tails after $N = 100$ terms; see Figure 7.1) may not be sufficient for many applications.

The Hall-Buckley-Eagleson method is fast, easy to implement, and fairly accurate (for $N = 50$ it is already accurate to at least two decimal places; see Figure 7.1). The Wood F method displays (in Figure 7.2) a slight increase in accuracy over the Hall-Buckley-Eagleson method, but it is slightly more difficult to implement, and one may need to be careful of degenerate cases (Section 7.3.4). For this reason, the Hall-Buckley-Eagleson method is recommended for most practitioners.

## 7.8 Discussion

This chapter reviews and describes methods for approximating the cdf of a weighted sum of chi-squared random variables. Obtaining such an approximation is necessary in the context of change detection for the variance, under certain assumptions, as will be shown in the next chapter. A general framework is introduced for comparing approximations of weighted sums, and this is applied to evaluating the performance of moment-matching methods that approximate the cdf of a weighted sum of chi-squared random variables. This performance analysis is far more extensive than previous attempts. An interesting observation revealed by the analysis in Section 7.6.1 is that the accuracy of the moment-matching methods generally increases as the number of terms $N$ increases. As far as we are aware, this has not been shown before. Also, the relative computational efficiency of the algorithms, shown in Section 7.6.3, has not been shown before. After considering various factors, the Hall-Buckley-Eagleson method is recommended for most practitioners, even though it may not be the most accurate method.

# Chapter 8

# Variance Change Detection

So far, this research has concentrated on the mean of a data stream. There are applications in which the variance is also of interest, in fields as diverse as finance [33] and medicine [143]. This chapter takes the first steps in developing a forgetting factor estimation framework for the variance. There are subtle issues to contend with, particularly whether the mean is estimated adaptively and whether the forgetting factor is shared. A first contribution in this chapter is a change detector for the variance of a normally-distributed stream. This uses the moment-matching approximation methods analysed in the previous chapter. A second contribution is development of update equations for the adaptive forgetting factor variance scheme, for both fixed and variable forgetting. Note that this development requires extensive algebraic manipulation, the details of which can be found in Appendix A.6. A final contribution revisits the F-AFF scheme of Section 4.2.2 and reformulates the method using an adaptive estimate of the variance.

## 8.1 Adaptive estimation of the variance

In this section the adaptive forgetting factor (AFF) variance $s^2_{N,\overrightarrow{\lambda}}$ is described, building on the derivation of the AFF mean in Chapter 3. The development here follows that given in Chapter 3; first the FFF variance $s^2_{N,\lambda}$ is defined, then sequential updating equations are derived, and finally the AFF variance $s^2_{N,\overrightarrow{\lambda}}$ will be defined.

### 8.1.1 The fixed forgetting factor variance

Suppose, as before, that the stream $x_1, x_2, \ldots$ is generated by the random variables $X_1, X_2, \ldots$ and that $N$ observations have been observed so far. The sample variance of the first $N$ observations is

$$s_N^2 = \frac{1}{N-1} \sum_{k=1}^{N} \left[x_k - \bar{x}_N\right]^2,$$

where $\bar{x}_N$ is the sample mean. As described in Section 3.2, the FFF mean is given by

$$\bar{x}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} x_i,$$

which motivates the definition of the FFF variance $s_{N,\lambda}^2$ by

$$s_{N,\lambda}^2 = \frac{1}{v_{N,\lambda}} \sum_{k=1}^{N} \lambda^{N-i} \left[x_k - \bar{x}_{N,\lambda}\right]^2, \tag{8.1}$$

where

$$v_{N,\lambda} = w_{N,\lambda}(1 - u_{N,\lambda}).$$

Note that we are using the forgetting factor mean $\bar{x}_{N,\lambda}$ rather than the sample mean $\bar{x}_N$ in Equation (8.1). The choice of $v_{N,\lambda}$ for $s_{N,\lambda}^2$ is analogous to the choice of $w_{N,\lambda}$ for $\bar{x}_{N,\lambda}$ in Section 3.2. It is shown in Appendix A.6.1 that for $X_1, \ldots, X_N$ i.i.d. with variance $\sigma^2$,

$$\mathrm{E}\left[\frac{1}{v_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} \left[X_i - \bar{X}_{N,\lambda}\right]^2\right] = \sigma^2.$$

### 8.1.2 Sequential update equations for $s_{N,\lambda}^2$

Defining

$$S_{N,\lambda} = \sum_{i=1}^{N} \lambda^{N-i} \left[x_i - \bar{x}_{N,\lambda}\right]^2, \tag{8.2}$$

so that

$$s_{N,\lambda}^2 = \frac{1}{v_{N,\lambda}} S_{N,\lambda}, \tag{8.3}$$

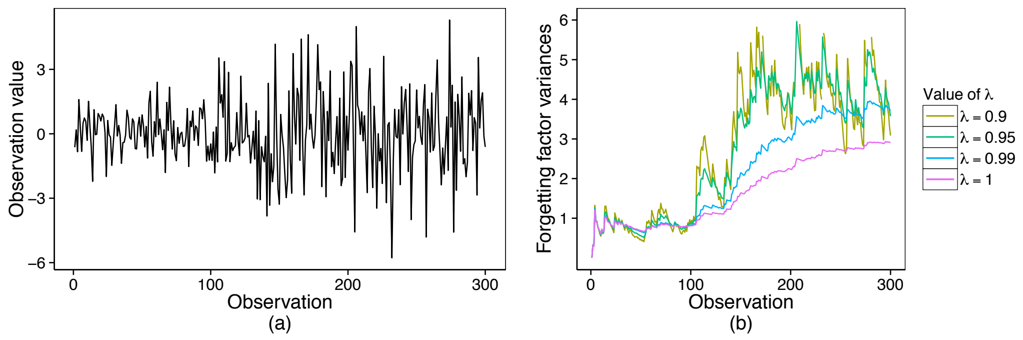

Figure 8.1: (a) a stream $x_1, x_2, \dots, x_{300}$ sampled from $X_1, \dots X_{100} \sim \mathrm{N}(0,1)$ and $X_{101}, \dots X_{300} \sim \mathrm{N}(0,2)$ $s^2_{N,\lambda}$, and (b) the value of the fixed forgetting factor variance $s^2_{N,\lambda}$ (on this stream) for different values of $\lambda$.

a sequential update equation for $S_{N,\lambda}$ is derived in Appendix A.6.2,

$$S_{N+1,\lambda} = \lambda S_{N,\lambda} + \left(\frac{w_{N+1,\lambda} - 1}{w_{N+1,\lambda}}\right) \left(\bar{x}_{N,\lambda} - x_{N+1}\right)^2 . \tag{8.4}$$

Since $v_{N,\lambda}$ is defined in terms of $w_{N,\lambda}$ and $u_{N,\lambda}$, for which we already have sequential update equations (see Section 3.2.4), Equations (8.3) and (8.4) enable $s^2_{N,\lambda}$ to be computed sequentially.

### 8.1.3 The relationship between $s^2_{N,\lambda}$ and $\lambda$

Figure 8.1 shows the value of $s^2_{N,\lambda}$ for different values of $\lambda$ on a single stream, while Figure 8.2 shows the average behaviour of $s^2_{N,\lambda}$, just as Figure 3.2 displays the average behaviour for $\bar{x}_{N,\lambda}$. Error bars are displayed at intervals, and have a width of one standard deviation. For all $\lambda$, $s^2_{1,\lambda}$ is displayed as 0 (since $s^2_{N,\lambda}$ is only defined for $N = 2, 3, \dots$). As for $\bar{x}_{N,\lambda}$, for smaller values of $\lambda$, $s^2_{N,\lambda}$ reacts to the change in the variance more quickly than when $\lambda$ is closer to 1. For example, Figure 8.2 shows that when $\lambda = 0.9$, $s^2_{N,\lambda}$ is close to the true value of $\sigma^2_2 = 4$ soon after the changepoint at $\tau = 100$. However, the error bars are larger for smaller values of $\lambda$.

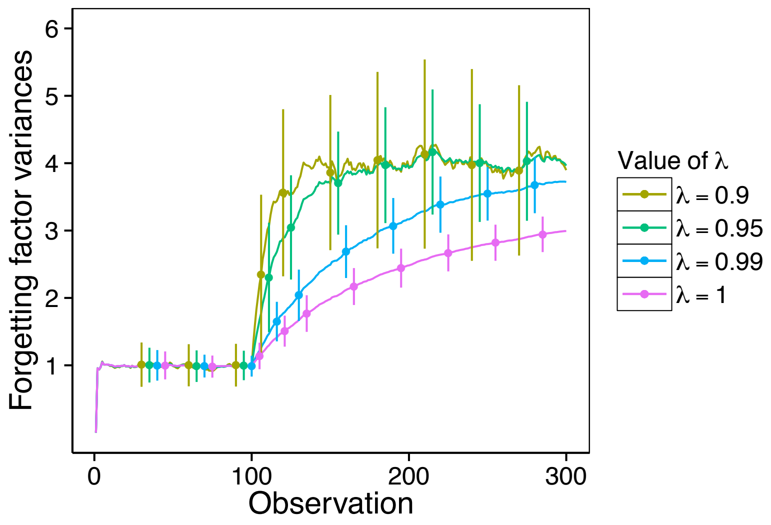

Figure 8.2: The average behaviour of $s^2_{N,\lambda}$, where $x_1, x_2, \ldots$ is sampled from $X_1, \ldots X_{100} \sim \mathrm{N}(0,1)$ and $X_{101}, \ldots X_{200} \sim \mathrm{N}(0,2)$, averaged over 100 simulations. Error bars (width of one standard deviation) are displayed at intervals.

### 8.1.4 Change detection for $s^2_{N,\lambda}$ assuming normality

Supposing that the observations $x_1, x_2, \ldots, x_N$ are sampled from the random variables $X_1, X_2, \ldots, X_N$, where

$$X_i \sim \mathrm{N}(\mu, \sigma^2) \tag{8.5}$$

and the fixed forgetting factor variance $s^2_{N,\lambda}$ is given by

$$s^2_{N,\lambda} = \frac{1}{v_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} \left[X_i - \bar{X}_{N,\lambda}\right]^2 .$$

If we define the random variables $Y_i$ by

$$Y_i = X_i - \bar{X}_{N,\lambda} \tag{8.6}$$

then the covariance for $i \neq j$ is given (see Appendix A.6.3) by

$$\mathrm{Cov}(Y_i, Y_j) = \left( u_{N,\lambda} - \frac{1}{w_{N,\lambda}} \left( \lambda^{N-i} + \lambda^{N-j} \right) \right) \sigma^2 \tag{8.7}$$

and the variance of $Y_i$ is given (see Appendix A.6.4) by

$$\mathrm{Var}\left[Y_i\right] = \left( 1 + u_{N,\lambda} - \frac{2}{w_{N,\lambda}} \lambda^{N-i} \right) \sigma^2. \tag{8.8}$$

Note the slight difference between the covariance of diagonal terms (variance) and the off-diagonal terms, an extra term with value $\sigma^2$. Now the FFF variance could be written in terms of the $Y_i$ variables as

$$s^2_{N,\lambda} = \sum_{i=1}^{N} \frac{\lambda^{N-i}}{v_{N,\lambda}} Y_i^2$$

and defining the random variables $Z_i$ as

$$Z_i = \sqrt{\frac{\lambda^{N-i}}{v_{N,\lambda}}} Y_i,$$

the FFF variance can be expressed simply as

$$s^2_{N,\lambda} = \sum_{i=1}^{N} Z_i^2. \tag{8.9}$$

Now $V_{ij}$, the covariance matrix of the $Z_i$ variables, can be expressed

$$V_{ij} = \begin{cases} \gamma_{ij} & \text{if } i \neq j \\ \gamma_{ii} + \alpha_i & \text{if } i = j \end{cases} \tag{8.10}$$

where

$$\gamma_{ij} = \frac{1}{v_{N,\lambda}} \sqrt{\lambda^{N-i}} \sqrt{\lambda^{N-j}} \left( u_{N,\lambda} - \frac{1}{w_{N,\lambda}} (\lambda^{N-i} + \lambda^{N-j}) \right) \tag{8.11}$$

$$\alpha_i = \frac{1}{v_{N,\lambda}} \tag{8.12}$$

and so

$$\gamma_{ii} = \frac{1}{v_{N,\lambda}} \lambda^{N-i} \left( u_{N,\lambda} - \frac{2}{w_{N,\lambda}} \lambda^{N-i} \right). \tag{8.13}$$

This is shown in Appendix A.6.5. The FFF variance $s^2_{N,\lambda}$ can be expressed as a weighted sum of *independent* chi-squared variables by using an extension of Cochran's Theorem, due to G. E. P. Box [18], which we quote here:

**Theorem 8.1.1** *If $z$ denotes a column vector of $N$ random normal variables $z_1, \ldots, z_N$ having expectation zero and distributed in a multivariate normal distribution with $N \times N$ variance-covariance matrix $V$, and if $Q = z'Mz$ is any real quadratic form of rank $R \leq N$, then $Q$ is distributed like a quantity*

$$X = \sum_{j=1}^{R} \nu_j \xi_j^2,$$

*where each $\xi_j^2 \sim \chi^2$ variate is distributed independently of every other, and the $\nu_k$ are the $R$ real nonzero eigenvalues of the matrix*

$$U = VM.$$

In our formulation of $s^2_{N,\lambda} = \sum_{i=1}^{N} Z_i^2$ in Equation (8.9), the matrix $M$ referred to in the theorem is simply the identity matrix, and so the matrix $U$ (referred in the theorem) is simply $V$, the covariance matrix of the $Z_1, Z_2, \ldots, Z_N$ random variables.

Therefore, the weights $\nu_j$ are simply the non-zero eigenvalues of the covariance matrix $V$, which is specified by Equations (8.10)-(8.13). Chapter 7 describes methods for approximating the cdf of $X = \sum_{j=1}^{R} \nu_j \xi_j^2$ that only require the first two or three cumulants $\kappa_r(X)$, where

$$\kappa_n(X) = 2^{n-1}(n-1)! \sum_{j=1}^{R} \nu_j^n.$$

If $N$ were large, computing the eigenvalues of an $N \times N$ matrix would be computationally intensive. Fortunately, it is not necessary to compute the values of the individual $\nu_j$ — we only need to compute the sums

$$\sum_{j=1}^{R} \nu_j^n,$$

which, according to [24], are simply given by

$$\text{tr}(V^n) = \sum_{j=1}^{R} \nu_j^n. \tag{8.14}$$

In order to use the Hall-Buckley-Eagleson method [59, 24] described in Section 7.3.3, the first three cumulants are needed, and so the traces of the matrices $V$, $V^2$ and $V^3$ need to be computed.

## Computing the cumulants of $s^2_{N,\lambda}$

In Appendix A.6.5 the following computations are given in detail. Briefly, using Equations (8.10)-(8.13), we can write the traces as

$$\begin{aligned}
\text{tr}(V) &= \sum_{i=1}^{N} V_{ii}, \\
\text{tr}(V^2) &= \sum_{k=1}^{N}\sum_{j=1}^{N} V_{kj}V_{jk}, \\
\text{tr}(V^3) &= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N} V_{qk}V_{kt}V_{tq}.
\end{aligned}$$

The first trace can be easily computed as

$$\text{tr}(V) = \sigma. \tag{8.15}$$

For the second trace, the symmetry of $V$ can be exploited to simplify calculations, but some effort is required to first rewrite the summation over $V_{ij}$ as a summation over $\gamma_{ij}$ (because of the split definition, and the terms on the diagonal). Eventually,

$$\text{tr}(V^2) = \frac{\sigma^2}{v^2}\left[u^2w^2 - \frac{2}{w}(w_{N,\lambda^3}) + uw^2\right] \tag{8.16}$$

is obtained. The third trace is computed in a similar manner to the second, but the computation is a bit more complex. Still, eventually one arrives at

$$\begin{aligned} \text{tr}(V^3) = \frac{\sigma^3}{v^3}\bigg[ &- \big(u^3w^3\big) + (w_{N,\lambda^3})(3u^2+6u+2) \\ &+ \frac{1}{w}(w_{N,\lambda^4})(-9-12u) + \frac{12}{w^2}(w_{N,\lambda^5})\bigg]. \end{aligned} \tag{8.17}$$

Now, the first three cumulants of $s^2_{N,\lambda}$ can be expressed as

$$\kappa_1(s^2_{N,\lambda}) = \sigma, \tag{8.18}$$

$$\kappa_2(s^2_{N,\lambda}) = \frac{2\sigma^2}{v^2}\left[u^2w^2 - \frac{2}{w}(w_{N,\lambda^3}) + uw^2\right], \tag{8.19}$$

$$\begin{aligned} \kappa_3(s^2_{N,\lambda}) = \frac{8\sigma^3}{v^3}\bigg[ &- \big(u^3w^3\big) + (w_{N,\lambda^3})(3u^2+6u+2) \\ &+ \frac{1}{w}(w_{N,\lambda^4})(-9-12u) + \frac{12}{w^2}(w_{N,\lambda^5})\bigg], \end{aligned} \tag{8.20}$$

by using our calculations in Equations (8.15), (8.16) and (8.16), and

$$\kappa_n(s^2_{N,\lambda}) = 2^{n-1}(n-1)! \cdot \text{tr}(V^n).$$

It is important to note that only the following seven quantities need to be sequentially updated in order to compute the first three cumulants at any time $N$:

$$u_{N,\lambda},\ v_{N,\lambda},\ w_{N,\lambda},\ w_{N,\lambda^2},\ w_{N,\lambda^3},\ w_{N,\lambda^4},\ w_{N,\lambda^5}. \tag{8.21}$$

This makes sequential change detection of the FFF variance possible in a streaming data context.

It should be noted that [104] had a similar approach using the combination of a EWMA scheme for the variance and the Satterthwaite-Welch method described in Section 7.3.2. However, only asymptotic values were used for the first two cumulants, rather than the exact values. The next section presents a simulation study illustrating how the behaviour of this change detector.

## 8.2 Simulation study for change detection with FFF variance

This section describes a simulation study illustrating how the change detector based on $s^2_{N,\lambda}$ performs. First, Section 8.2.1 shows how the methodology developed in Section 8.1 can be combined with an approximate method for computing the cdf of a weighted sum of chi-squared random variables (several of which are described in Chapter 7) to arrive at a decision rule for whether or not the variance is in control. Section 8.2.2 then shows how the change detector performs for a single changepoint in the variance, with the pre-change parameters assumed known.

### 8.2.1 Performance of the approximate cdf with the FFF variance

Suppose that the stream $x_1, x_2, \ldots$ is generated from the random variables $X_1, X_2, \ldots$, where

$$\begin{aligned} X_1, X_2, \ldots, X_{100} &\sim \mathrm{N}(0,1), \\ X_{101}, X_{102}, \ldots &\sim \mathrm{N}(2,1). \end{aligned}$$

For $\lambda = 0.95$, three methods are compared for computing $F(s^2_{N,\lambda})$, where $F$ is the cdf of the weighted sum of independent chi-squared random variables with coefficients

$$\nu_1, \nu_2, \ldots, \nu_{N-1}.$$

These coefficients for the independent weighted sum are obtained from the coefficients

$$\frac{1}{v_{N,\lambda}}, \frac{\lambda}{v_{N,\lambda}}, \ldots, \frac{\lambda^{N-1}}{v_{N,\lambda}}, \tag{8.22}$$

by using Box's extension of Cochran's Theorem (Theorem 8.1.1 in Section 8.1.4) for the dependent weighted sum

$$s^2_{N,\lambda} = \frac{1}{v_{N,\lambda}} \sum_{i=1}^{N} \lambda^{N-i} \left[x_i - \bar{x}_{N,\lambda}\right]^2. \tag{8.23}$$

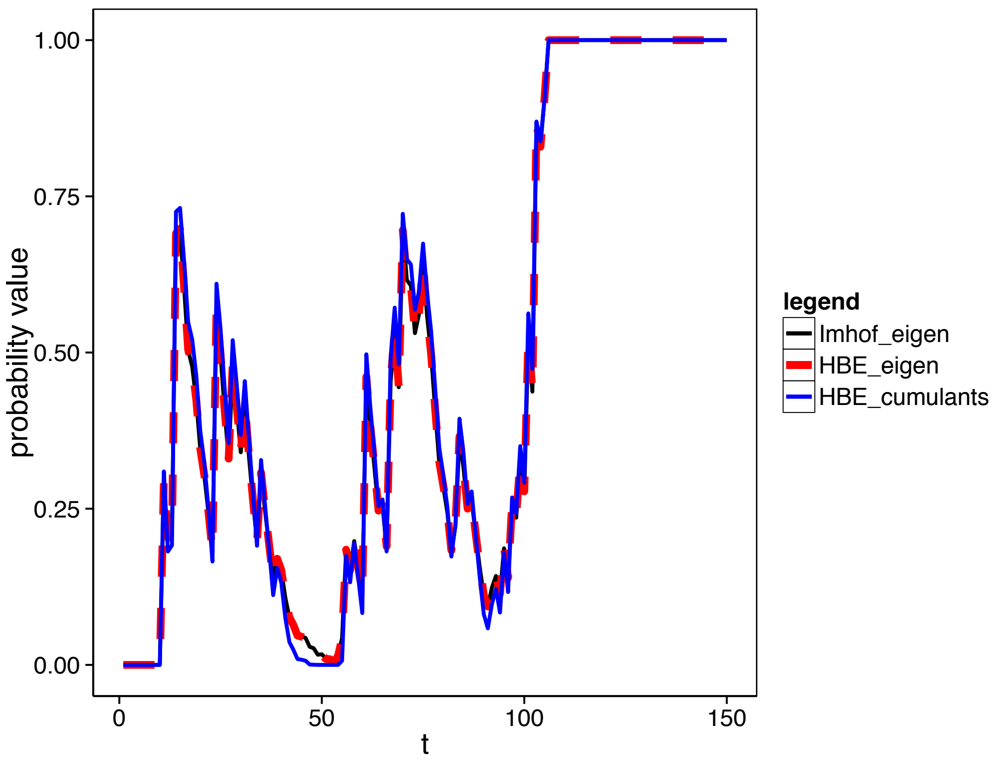

Figure 8.3: The value of $F(s^2_{N,\lambda})$ for a single stream, for three methods of computation, with $\lambda = 0.95$, for $X_1, \ldots, X_{100} \sim \mathrm{N}(0,1)$, $X_{101}, \ldots, X_{300} \sim \mathrm{N}(0,2^2)$.

Note the difference in the number of coefficients: there are $N$ coefficients for the dependent sum, but $N-1$ coefficients for the independent sum. Three methods for computing the cdf $F$ are now compared in Figures 8.3 and 8.4. The first method, indicated by the solid black line, uses Cochran's theorem to obtain the "independent" coefficients in Equation (8.22) from the "dependent" coefficients in (8.23). This step involves computing the eigenvalues of an $N \times N$ matrix. Then Imhof's method uses these "independent" coefficients to compute $F(s^2_{N,\lambda})$. Recall that Imhof's method requires the vector of coefficients to compute the cdf, and can be considered an exact method. Therefore, this first method — labelled in the figures as *Imhof_eigen* — is essentially exact.

The second method, indicated by the dashed red line, also computes the "independent" coefficients, but then uses the Hall-Buckley-Eagleson (HBE) method described in

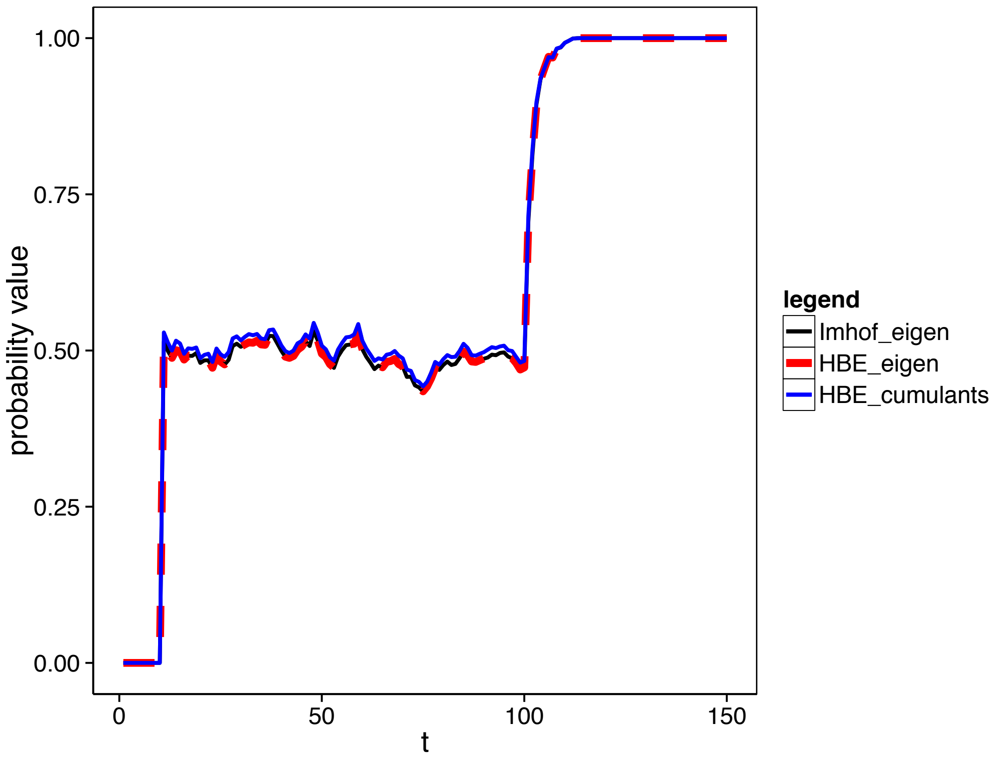

Figure 8.4: The value of $F(s^2_{N,\lambda})$ for a single stream, for three methods of computation, with $\lambda = 0.95$, for $X_1, \ldots, X_{100} \sim \text{N}(0,1)$, $X_{101}, \ldots, X_{300} \sim \text{N}(0,2^2)$, averaged over 100 simulations.

Section 7.3.3 to compute $F(s^2_{N,\lambda})$. This method is labelled in the figures as *HBE_eigen*. Recall that the HBE method does not require the vector of coefficients, only the first three moments, and is an approximate method. Figure 8.3 shows that it returns essentially the same values as the first method, which is essentially exact. However, this method is still not suitable for deployment on a stream (since it also uses Theorem 8.1.1).

The third method, indicated by the solid blue line, sequentially computes the first three moments of $s^2_{N,\lambda}$, as described in Section 8.1.4, and then uses the HBE method to compute $F(s^2_{N,\lambda})$. This method, labelled in the figures as *HBE_cumulants* — is completely online, yet Figure 8.3 shows that it returns almost exactly the same values as the first two methods which are essentially exact.

Figure 8.3 shows there is a high degree of agreement between the three methods for

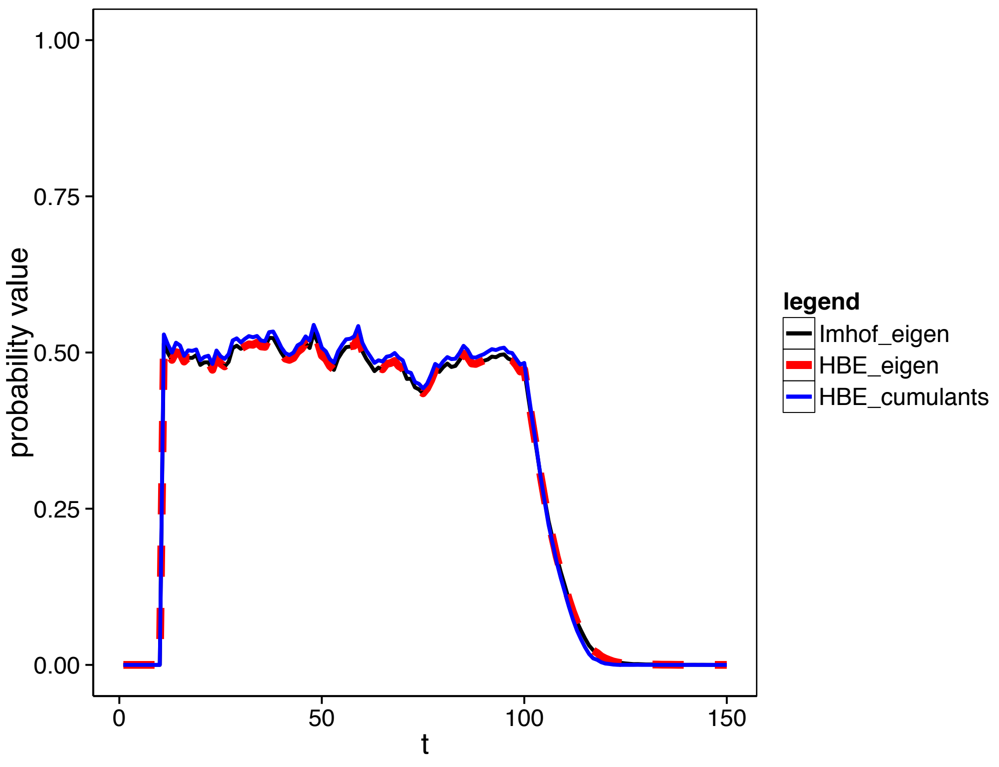

Figure 8.5: The value of $F(s^2_{N,\lambda})$ for a single stream, for three methods of computation, with $\lambda = 0.95$, for $X_1, \dots, X_{100} \sim \text{N}(0, 1)$, $X_{101}, \dots, X_{300} \sim \text{N}(0, 0.5^2)$, averaged over 100 simulations.

a single stream. Before the changepoint at $\tau = 100$, these is some degree of stochastic variation for $F(s^2_{N,\lambda})$. This should be expected, since $F(s^2_{N,\lambda})$ is essentially a $U(0, 1)$ random variable. However, very soon after the changepoint the value of $F(s^2_{N,\lambda})$ increases to a value close to 1, and remains close to 1 for the rest of the stream. This behaviour, and the high degree of agreement between the three methods, shows that our approach of computing the cumulants has some merit.

Figure 8.4 shows the average behaviour of $p = F(s^2_{N,\lambda})$ over 100 streams. The average value of $F(s^2_{N,\lambda})$ is close to 0.5 before the changepoint, which would make sense if $F(s^2_{N,\lambda})$ is a $U(0, 1)$ random variable. Again, after the changepoint at $\tau = 100$, the average value of $p$ is very close to 1. Error bars have been omitted from Figure 8.4, because change detection performance will be compared in Section 8.2.2. Note that $F(s^2_{N,\lambda})$ is only computed for

$N = 11, 12, \ldots$, since Imhof's method is very slow for small values of $N$, and in practice a burn-in period greater than $B = 10$ would be used, so these values ($F(s^2_{N,\lambda})$ for $N = 2, \ldots, 10$) are not of interest. More importantly, it may be desirable to allow $s^2_{N,\lambda}$ to "settle down" before computing $F(s^2_{N,\lambda})$ ($s^2_{N,\lambda}$ may have a large variance for small $N$). Therefore, in Figures 8.3, 8.4 and 8.5, $F(s^2_{N,\lambda}) = 0$ for $N = 2, \ldots, 10$.

While Figures 8.3 and 8.4 show the behaviour of $F(s^2_{N,\lambda})$ for an increase in the variance, a similar figure, Figure 8.5, is obtained for a *decrease* in the variance. However, $F(s^2_{N,\lambda})$ in this case is close to 0 after the changepoint, rather than close to 1, which is the case for an increase in the variance.

### 8.2.2 Change detection using the FFF variance

Suppose that the stream $x_1, x_2, \ldots$ is generated from the random variables $X_1, X_2, \ldots$, where

$$\begin{aligned} X_1, X_2, \ldots, X_{100} &\sim \mathrm{N}(0, 1), \\ X_{101}, X_{102}, \ldots &\sim \mathrm{N}(0, 2^2). \end{aligned}$$

Suppose for the moment that we are only interested in detecting an *increase* in the mean. This would correspond to the case when $F(s^2_{N,\lambda})$ is very close to 1. A change detector could be constructed by specifying $p_{max}$, so that a change is signalled by

$$F(s^2_{N,\lambda}) > p_{max}. \tag{8.24}$$

An interesting adjustment to this scheme is to introduce another parameter $R$, which specifies that Equation (8.24) must be satisfied by $R$ consecutive values of $N$, e.g.

$$N = \hat{\tau}, \hat{\tau} + 1, \ldots \hat{\tau} + R - 1,$$

before a change is signalled at $\hat{\tau} + R - 1$. This is a type of **run rule**; see [113, 32] for change detection schemes based on run rules. This change detection scheme is named *FFFvar*, and some results are shown in Table 8.1 for $p_{max} = 0.999$ and different choices

| Algo. | Param. | Param. Val. | ARL0 | SDRL0 | ARL1 | SDRL1 |
|---|---|---|---|---|---|---|
| FFFvar | $(\lambda, p_{max}, R)$ | (0.95, 0.999, 1) | 15223.8 | (21892.2) | 10.7 | (12.3) |
| FFFvar | $(\lambda, p_{max}, R)$ | (0.90, 0.999, 1) | 5465.8 | (7760.15) | 10.1 | (12.1) |
| FFFvar | $(\lambda, p_{max}, R)$ | (0.95, 0.999, 3) | 32117.0 | (38119.6) | 13.9 | (15.5) |
| FFFvar | $(\lambda, p_{max}, R)$ | (0.90, 0.999, 3) | 35627.8 | (45578.0) | 14.5 | (17.0) |

Table 8.1: An ARL table for FFFvar over 1000 trials. For ARL1, $X_1, \dots X_{100} \sim \text{N}(0,1)$ and $X_{101}, \dots X_{200} \sim \text{N}(0, 2^2)$. Normal streams, pre-change variance assumed known.

of $\lambda$ and $R$. Note that although no burn-in is necessary to estimate the parameters (since for this simulation study the pre-change mean and variance are assumed known), $F(s^2_{N,\lambda})$ is not computed for $N = 2, \dots, 10$, because $s^2_{N,\lambda}$ may have a large variance for small $N$. Table 8.1 shows that this scheme has high ARL0 compared to ARL1, for all parameter choices considered, which provides some evidence that the methodology developed in this chapter can be successfully incorporated into a change detector. The next section presents equations for the adaptive forgetting factor variance.

## 8.3 The adaptive forgetting factor variance

One might consider the natural definition of the adaptive forgetting factor variance to be to modify Equation (8.1) to

$$s^2_{N,\overrightarrow{\lambda}} = \frac{1}{v_{N,\overrightarrow{\lambda}}} \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \left[ x_k - \bar{x}_{N,\overrightarrow{\lambda}} \right]^2 . \tag{8.25}$$

However, when it comes to defining the derivative of $s^2_{N,\overrightarrow{\lambda}}$ with respect to $\overrightarrow{\lambda}$, we have

$$s^2_{N,\overrightarrow{\lambda}+\epsilon} = \frac{1}{v_{N,\overrightarrow{\lambda}}} \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p + \epsilon \right) \left[ x_k - \bar{x}_{N,\overrightarrow{\lambda}+\epsilon} \right]^2 . \tag{8.26}$$

The key observation is that Equation (8.26) now has two instances of $\epsilon$, in the terms

$$\lambda_p + \epsilon, \qquad \bar{x}_{N,\overrightarrow{\lambda}+\epsilon}.$$

This makes the derivation of the equations for the derivative of $s^2_{N,\overrightarrow{\lambda}}$ extremely difficult. The derivation is possible, though, and a summary of the update equations for this formulation is given in Appendix A.6.8. However, it may not even make sense to use a single forgetting factor for both the mean and the variance. For example, if there is a change in the variance, but not the mean, the decrease in $\overrightarrow{\lambda}$ may affect the subsequent estimation of the mean. Therefore, the approach in this chapter is to use separate forgetting factors for the mean and variance.

Suppose we use the adaptive forgetting factor

$$\overrightarrow{B} = (B_1, B_2, \ldots, B_N), \tag{8.27}$$

for the mean,

$$\bar{x}_{N,\overrightarrow{B}} = \frac{1}{w_{N,\overrightarrow{B}}} \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} B_p \right) x_k, \tag{8.28}$$

while $\overrightarrow{\lambda}$,

$$\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_N), \tag{8.29}$$

is the adaptive forgetting factor for the variance, which is defined as

$$s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{1}{V_{N,\overrightarrow{\lambda},\overrightarrow{B}}} \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \left[ x_k - \bar{x}_{N,\overrightarrow{B}} \right]^2. \tag{8.30}$$

We can calculate sequential update equations for $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, as well as sequential update equations for the derivative of $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ with respect to $\overrightarrow{\lambda}$. Extensive manipulation in Appendix A.6.6 derives these equations. Now, the adaptive forgetting factor variance can be defined as

$$s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{1}{V_{N,\overrightarrow{\lambda},\overrightarrow{B}}} S_{N,\overrightarrow{\lambda},\overrightarrow{B}}, \tag{8.31}$$

where

$$S_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \left[ x_k - \bar{x}_{N,\overrightarrow{B}} \right]^2, \tag{8.32}$$

and $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ is defined so that

$$\mathrm{E}[s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}] = \sigma^2.$$

A non-sequential derivation of $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ is shown in Appendix A.6.6 to yield

$$V_{N,\overrightarrow{\lambda},\overrightarrow{B}} = (1 + u_{N,\overrightarrow{B}})w_{N,\overrightarrow{\lambda}} - \frac{2}{w_{N,\overrightarrow{B}}} \sum_{i=1}^{N} \Big( \prod_{p=i}^{N-1} B_p \Big) \Big( \prod_{q=i}^{N-1} \lambda_q \Big). \tag{8.33}$$

Now, for the following update equations for the $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, we assume that for time $N+1$ we have the following quantities available:

$$x_{N+1}, \lambda_N, m_{N,\overrightarrow{\lambda}}, w_{N,\overrightarrow{\lambda}}, w_{N+1,\overrightarrow{\lambda}}, B_N, w_{N+1,\overrightarrow{B}}, u_{N+1,\overrightarrow{B}}, \bar{x}_{N,\overrightarrow{B}}, \bar{x}_{N+1,\overrightarrow{B}}. \tag{8.34}$$

The quantities in Equation (8.34) are all defined in Chapter 3. Now, the sequential update equations for $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ are

$$P_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = B_N \lambda_N \Big( P_{N,\overrightarrow{\lambda},\overrightarrow{B}} \Big) + 1, \qquad P_{0,\overrightarrow{\lambda},\overrightarrow{B}} = 0 \tag{8.35}$$

$$V_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = (1 + u_{N+1,\overrightarrow{B}})w_{N+1,\overrightarrow{\lambda}} - \frac{2}{w_{N+1,\overrightarrow{B}}} P_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}, \tag{8.36}$$

and the sequential update equations for $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ are given by

$$\xi_{N+1,\overrightarrow{B}} = \frac{1}{w_{N+1,\overrightarrow{B}}} \Big( \bar{x}_{N,\overrightarrow{B}} - x_{N+1} \Big), \tag{8.37}$$

$$H_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = \xi_{N+1,\overrightarrow{B}} \big[ 2m_{N,\overrightarrow{\lambda}} - w_{N,\overrightarrow{\lambda}} (2\bar{x}_{N,\overrightarrow{B}} - \xi_{N+1,\overrightarrow{B}}) \big], \tag{8.38}$$

$$S_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = \lambda_N \Bigg[ S_{N,\overrightarrow{\lambda},\overrightarrow{B}} + H_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} \Bigg] + \big[ x_{N+1} - \bar{x}_{N+1,\overrightarrow{B}} \big]^2. \tag{8.39}$$

Then, Equation (8.31) can be used to compute

$$s^2_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{1}{V_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}} S_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}.$$

### 8.3.1 Cost functions for the AFF variance

Following the development of methodology for the AFF mean in Section 3.3, we update $\lambda_N \to \lambda_{N+1}$ by using gradient descent and the derivative of a chosen cost function. The choice of cost function for the variance would probably involve the variance $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, which is composed of the weighted sum of squares $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ defined in Equation (8.32) and the weight $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ defined in Equation (8.33). The derivative of $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ is defined as

$$Z_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{\partial}{\partial \overrightarrow{\lambda}} S_{N,\overrightarrow{\lambda},\overrightarrow{B}}, \tag{8.40}$$

and can be sequentially defined by

$$\begin{aligned} F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} &= \left[\xi_{N+1,\overrightarrow{B}}\right]\left[2\Delta_{N+1,\overrightarrow{\lambda}} - \Omega_{N+1,\overrightarrow{\lambda}}\left(2\bar{x}_{N,\overrightarrow{B}} - \xi_{N+1,\overrightarrow{B}}\right)\right] \\ Z_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} &= \lambda_N Z_{N,\overrightarrow{\lambda},\overrightarrow{B}} + S_{N,\overrightarrow{\lambda},\overrightarrow{B}} + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}, \qquad Z_{1,\overrightarrow{\lambda},\overrightarrow{B}} = 0, \end{aligned}$$

where $\xi_{N+1,\overrightarrow{B}}$ has been defined in Equation (8.37), and the quantities

$$\lambda_N,\ \Delta_{N+1,\overrightarrow{\lambda}},\ \Omega_{N+1,\overrightarrow{\lambda}},\ \bar{x}_{N,\overrightarrow{B}} \tag{8.41}$$

are additionally required. A sketch of the derivation of these update equations is given in Appendix A.6.7, which also contains the update equations for the derivative of $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$. With all of these equations, the AFF $\overrightarrow{\lambda}$ can be sequentially updated using

$$\lambda_{N+1} = \lambda_N - \zeta \frac{\partial}{\partial \overrightarrow{\lambda}} J_{N+1,\overrightarrow{\lambda},\overrightarrow{B}},$$

where $\zeta$ is some step-size and $J_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}$ is a cost function which is a function of $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$. One possible candidate for a cost function is

$$J_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = \left[s^2_{N,\overrightarrow{B},\overrightarrow{\lambda}} - (x_{N+1} - x_N)^2\right]^2,$$

which attempts to mimic the cost function $L_{N+1,\overrightarrow{\lambda}}$ for the mean. Since $J_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}$ is a function of $s^2_{N,\overrightarrow{B},\lambda}$, it can be computed sequentially using the equations provided in this section. Of course, other cost functions could be used. Instead of using a function of $s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}}$ or $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, it may be preferable to use

$$E_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \left[ x_k - \bar{x}_{N-1,\overrightarrow{B}} \right]^2, \tag{8.42}$$

which is the same as $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, except that the mean used here is $\bar{x}_{N-1,\overrightarrow{B}}$ rather than $\bar{x}_{N,\overrightarrow{B}}$. Sequential update equations can be derived as in Appendix A.6.7. Finally, our cost function for the AFF mean

$$L_{N+1,\overrightarrow{\lambda}} = \left[ \bar{x}_{N,\overrightarrow{\lambda}} - x_{N+1} \right]^2, \tag{8.43}$$

may also have some merit, since it is itself a residual, being the last term in $E_{N,\overrightarrow{\lambda},\overrightarrow{B}}$.

## 8.4 Distribution-free methods utilising the AFF variance

Having an adaptive estimator for the variance provides the opportunity to enhance change detectors for the mean. This section outlines one such approach. Recalling the F-AFF method described in Section 4.2.2, which compares the AFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ with the FFF mean $\bar{x}_{N,\overrightarrow{\lambda}}$ using the decision rule,

- $\bar{x}_{N,\overrightarrow{\lambda}} \in (a_N, b_N) \Rightarrow$ stream is in-control,
- $\bar{x}_{N,\overrightarrow{\lambda}} \notin (a_N, b_N) \Rightarrow$ changepoint has occurred.

where $a_N$ and $b_N$ were defined by

$$a_N = \bar{x}_{N,\lambda} - \beta\sigma,$$
$$b_N = \bar{x}_{N,\lambda} + \beta\sigma.$$

In these equations, $\sigma^2$ is the variance of the stream which is assumed to be known, or could be estimated during a burn-in period. Now with the development of the forgetting factor variance $s^2_{N,\lambda}$, the knowledge or estimation of $\sigma^2$ during a burn-in period is no longer necessary. The control limits $a_N$ and $b_N$ can be enhanced by replacing $\sigma$ with $s_{N,\lambda}$, i.e.

$$\begin{aligned} a_N &= \bar{x}_{N,\lambda} - \beta s_{N,\lambda}, \\ b_N &= \bar{x}_{N,\lambda} + \beta s_{N,\lambda}. \end{aligned}$$

This would provide change detection sensitive to processes which are varying in both mean and variance. Of course, as with previous chapters, issues remain with setting control parameters.

As a final remark, we can use Chebyshev's inequality with $s^2_{N,\lambda}$ as in Section 4.2.1. However, as for the mean, there is no clear way to set the control parameter $\delta$. For this reason, we omit these equations.

## 8.5 Discussion

This chapter introduces the mathematical background for a forgetting factor framework for adaptively estimating the variance. A decision rule is proposed for detecting a change in the variance of a normally-distributed stream. The simulation study in Section 8.2 shows that the framework can be successfully incorporated into a change detector. The sequential update equations for the adaptive forgetting factor variance scheme are determined and summarised. Finally, an enhancement of a distribution-free change detector for the mean is suggested.

# Chapter 9

# Conclusion

Detecting changes in streaming data raises challenges not usually encountered in the traditional SPC setting. Potentially unending streams of observations may be arriving at a very high rate, meaning that online, computationally-efficient methods are required for timely detection. Furthermore, data streams are expected to have multiple changepoints, leading to multiple regimes, and so any selection of control parameters should apply to a wide range of scenarios. This is because it is not possible to intervene after a change is detected and re-calibrate parameters — the observations of a data stream will continue to flow, regardless of the action taken. The goal should therefore be to reduce the number of control parameters, or at least let have them be automatically set.

Our change detection methodology can be broadly split into two parts: obtaining up-to-date adaptive estimates of the mean and variance of the stream, and defining a decision rule for signalling a change. Adaptive estimation provides a tool for handling time-varying sequences. The mathematical formulation is interesting, and when the forgetting factor is *adaptive*, this leads to a reduction in the dependence on control parameters. There are many ways to choose a decision rule, some based on distributional assumptions, while others are distribution-free, and a variety of options are explored.

To summarise the contributions of this thesis:

- We describe an adaptive forgetting factor (AFF) framework, and define a derivative with respect to the AFF $\overrightarrow{\lambda}$ which formalises previous heuristic derivations. The derivative is also used to update the value of the $\overrightarrow{\lambda}$ using gradient descent. Sequential update equations for the AFF mean are derived; these are crucial for streaming data analysis. A study of the optimal fixed and adaptive forgetting factor shows that, when encountering a change in the mean, $\overrightarrow{\lambda}$ performs similarly to the optimal fixed forgetting factor (FFF) $\lambda$.

- We introduce the *continuous monitoring* setting where multiple changepoints occur, and define additional performance metrics required for meaningful comparison of change detection methods in this context. The need to continuously monitor data streams arises with modern technology, and we cannot simply use existing methods that monitor for only a single changepoint. The AFF scheme for detecting a change in the mean was shown to perform similarly to CUSUM and EWMA, two methods which are regarded as benchmarks in the single changepoint context. Crucially, the AFF scheme only relies on the setting of a single control parameter, while CUSUM and EWMA both require two parameters to be selected, and selecting these two parameters to obtain optimal performance in the multiple changepoint scenario is not straightforward. The AFF scheme was applied to monitoring a real foreign exchange stream, and *sequentially* detects the location of changepoints strikingly similar to an optimal *offline* method.

- We extend our framework to the multivariate setting; indeed, many examples of continuous monitoring may be more natural in the multivariate context. Two different approaches to defining the multivariate AFF mean are discussed, and three change detection rules are defined based on the univariate formulation and different methods for combining $p$-values. Our MVAFF-BCov method is shown to perform comparably to a state-of-the-art self-starting multivariate method for both independent and dependent streams. However, there is more dependence on the step size $\eta$ when there is covariance between the components; remedying this issue will be the subject of future work. The methodology is then applied to detecting changes in computer network traffic, and exemplar results are provided.

- Besides monitoring the mean, it is also of interest to monitor the variance of a stream. Defining a decision rule requires more work, and under certain assumptions requires the cdf of a weighted sum of chi-squared random variables, for which there is no analytical solution. While there are many approximate methods to compute such a cdf, it is not clear which should be the preferred choice in a streaming data context. This motivates the introduction of a general framework for evaluating the performance of methods for approximately computing the cdf of a weighted sum of random variables, which are all sampled from the same arbitrary distribution. This framework is applied to evaluate moment-matching methods for approximately computing the cdf of a weighted sum of *chi-squared* random variables in terms of accuracy and computational efficiency. An interesting discovery is that these methods all become more accurate as the number of terms in the weighted sum increases.

- We extend the framework for the FFF mean to define the FFF variance. A decision rule is derived under the assumption that the observations follow a normal distribution, which then requires a method for approximately computing the cdf of a weighted sum of chi-squared random variables. In order to implement one of the moment-matching methods earlier described, the first three cumulants of the FFF variance are derived. Next, the AFF variance is defined, utilising a separate AFF for the mean. Sequential update equations are derived in detail for the corresponding AFF formulation. Finally, the AFF mean, FFF mean and FFF variance are combined to form a distribution-free control chart for detecting a change in the mean.

## Future work

While the work in this thesis has already led to the consideration of several new directions for future work, the following three topics represent the most immediate areas to be addressed.

### Issues concerning the step size $\eta$

The definition of the adaptive forgetting factor in Chapter 3 requires the setting of a step size $\eta$ for the gradient descent update equation, Equation (3.18). Figure 3.8 shows that when a change in the mean occurs, the value of $\overrightarrow{\lambda}$ drops dramatically, regardless of the value of

$\eta$, but recovery to pre-change levels is slower for smaller values of $\eta$. However, this does not seem to affect the value of the AFF mean, as shown in Figure 3.9, nor does it affect the change detection performance of the AFF scheme in the continuous monitoring context, as shown in Table 5.2, nor the MVAFF-BCov scheme when the streams are independent, as shown in Table 6.2. However, it *does* affect the change detection performance in the *single* changepoint setting, as shown in Table 4.3, and the multivariate continuous monitoring setting when the streams are *dependent*, as shown in Table 6.5.

One future avenue of research is to ensure faster recovery of the adaptive forgetting factor to pre-change levels, regardless of the setting. A potentially fruitful direction worth exploration is the use of second-order gradient descent methods. For the multivariate setting, incorporating the full covariance matrix may address the issue when the component streams are dependent.

## Variance change detection

Chapter 8 provided preliminary steps towards constructing a change detector. What remains is to conduct an empirical study of variance change detection. Extensive simulations are required, as first the single change case and continuous monitoring cases should be considered. For the AFF variance, there are several candidates for the choice of cost function to be considered and compared. In Chapter 6 it was noted that a multivariate AFF variance may be useful for adaptively estimating the covariance of a stream. Change detection in the covariance matrix could also be explored.

## Distribution-free schemes

Many of the change detection rules considered in this thesis make the assumption that the underlying distribution of the stream is normal, although the parameters may be unknown. Some distribution-free methods were proposed, but this invariably introduced additional control parameters, without obvious guidance on how these parameters should be set. These methods are certainly worth revisiting, and principled methods to set the control parameters should be sought.

Although the construction of the prediction interval implicitly assumed that the underlying distribution was symmetric, recent preliminary work has suggested a way to take the skewness of the the distribution into account, in a similar manner to work done in [37].

# Appendix A

# Derivations

## A.1 Methods for combining $p$-values

In this section the formulations for Fisher's method [49] and Stouffer's method [144] for combining $p$-values are briefly outlined. Suppose that the $p$-values $p'_1, p'_2, \ldots, p'_d$ are provided. Recall that the smaller the value of $p'_i$, the more significant it is.

### A.1.1 Fisher's method

Fisher's method [49] combines the $p$-values, $p'_1, p'_2, \ldots, p'_d$, into a single $p$-value,

$$p' = F_{\chi^2_{2d}}\left(-2\sum_{j=1}^{d}\log(1-p'_j)\right), \tag{A.1}$$

where $F_{\chi^2_{2d}}$ is the cdf of the chi-squared distribution with $2d$ degrees of freedom. Again, the smaller the value of $p'$, the more significant it is.

### A.1.2 Stouffer's method

Provided with $p$-values $p'_1, p'_2, \ldots, p'_d$, Stouffer's method [144] first defines

$$Z_i = F^{-1}_{\mathrm{N}(0,1)}(1-p'_i),$$

where $F^{-1}_{\mathrm{N}(0,1)}$ is the *inverse* of the cdf of the $\mathrm{N}(0,1)$ distribution, and then defines

$$p' = F_{\mathrm{N}(0,1)}\left(\frac{1}{\sqrt{d}}\sum_{i=1}^{d} Z_i\right), \tag{A.2}$$

where $F_{\mathrm{N}(0,1)}$ is the cdf of the $\mathrm{N}(0,1)$ distribution. Again, the smaller the value of $p'$, the more significant the result.

## A.2 Distribution theory

This section provides necessary background on distribution theory, primarily for Chapter 7. A standard reference is [134].

### A.2.1 Normal, Gamma and Chi-squared distributions

A normal random variable $X \sim \mathrm{N}(\mu, \sigma^2)$ follows a distribution with probability density function (pdf)

$$f(x;\mu,\sigma) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[\frac{(x-\mu)^2}{2\sigma^2}\right].$$

A Gamma random variable $Y \sim \Gamma(k,\theta)$, where $k$ and $\theta$ are the shape and scale parameters, respectively, has pdf

$$g(x;k,\theta) = \frac{1}{\Gamma(k)\theta^k}x^{k-1}e^{-\frac{x}{\theta}},$$

for $x \geq 0$ (and 0 otherwise) where $\Gamma$ is the Gamma function. If we consider the random variable $W = Y^2$, where $Z \sim \mathrm{N}(0,1)$ is a standard normal random variable, we can derive (Appendix A.2.3) the pdf of $W$ to be

$$h(w) = \frac{1}{\Gamma(1/2)2^{1/2}}w^{1/2-1}e^{-\frac{w}{2}}, \tag{A.3}$$

for $w \geq 0$ (and 0 otherwise) which shows that $W \sim \Gamma(1/2, 2)$. We call this a *chi-squared* random variable with one degree of freedom, and write $W \sim \chi^2(1)$. In other words, $\chi^2(1) \equiv \Gamma(1/2,2)$. Using a result about Gamma variables (Appendix A.2.4), we can show that if $W_i \sim \chi^2(1)$, for $i = 1, 2, \ldots, n$, then $\sum_{i=1}^{n} W_i \sim \chi^2(n) \equiv \Gamma(n/2, 2)$.

## A.2.2 Cumulants and moments

The $r$th **moment** of a random variable $X$ is defined for $k = 1, 2, \dots$ as

$$m_r = m_r(X) = \mathrm{E}[X^k] = \int_S x^r \, \mathrm{d}F_X(x), \tag{A.4}$$

where $S$ is the support of the random variable $X$ [117]. The **moment generating function** $\mathbb{M}_X$ can be defined as

$$\mathbb{M}_X(t) = \mathrm{E}[e^{tX}], \qquad t \in \mathbb{R}. \tag{A.5}$$

The $r$th **central moment** is defined as

$$\mu_r = \mu_r(X) = \mathrm{E}\left[(X - m_1)^r\right], \tag{A.6}$$

where $m_1 = \mathrm{E}[X]$. If two random variables have the same moments $m_1, m_2, \dots$, then they share a common distribution.

The **cumulant generating function** $\mathbb{K}_X$ is then defined as

$$\mathbb{K}_X(t) = \log \mathbb{M}_X(t), \tag{A.7}$$

and the $r$-th **cumulant** is then defined as the $r$-th derivative of $\mathbb{K}_X(t)$ evaluated at $t = 0$, that is

$$\kappa_r = \kappa_r(X) = \frac{\mathrm{d}^r}{\mathrm{d}t^r} \mathbb{K}_X(t) \bigg|_{t=0}. \tag{A.8}$$

It is of note the the first three cumulants are equal to the first three central moments,

$$\kappa_1 = \mu_1, \qquad \kappa_2 = \mu_2, \qquad \kappa_3 = \mu_3. \tag{A.9}$$

This property is exploited in Chapter 7. Also, it is convenient for Chapter 7 to utilise invariance properties of moments and cumulants; specifically, the moments of $X$ determine the cumulants of $X$, and vice versa. The relationship between the values of moments and cumulants of a random variable is expressed by $m_1 = \kappa_1$ and

$$m_r = \kappa_r + \sum_{i=1}^{r-1} \binom{r-1}{i-1} \kappa_i m_{r-i}, \qquad r = 2, 3, \ldots. \tag{A.10}$$

Finally, it is necessary to know that the $r$-th cumulant of a chi-squared random variable $W \sim \chi_1^2$ is given by the formula

$$\kappa_r(W) = 2^{r-1}(r-1)!, \tag{A.11}$$

as shown in Appendix A.2.5.

### A.2.3 The square of a standard normal variable is chi-squared

Suppose that $X \sim N(0,1)$, and $Y = X^2$. Further suppose that $F_X$ is the cdf of $X$ and $F_Y$ is the cdf of $Y$. We now wish to find the pdf of $Y$. For $y < 0$,

$$F_Y(y) = \text{Pr}(Y < y) = \text{Pr}(X^2 < y) = 0$$

For $y \geq 0$,

$$\begin{aligned} F_Y(y) = \text{Pr}(Y < y) &= \text{Pr}(X^2 < y) \\ &= \text{Pr}(-\sqrt{y} < X < \sqrt{y}) \\ &= F_X(\sqrt{y}) - F_X(-\sqrt{y}) \\ &= F_X(\sqrt{y}) - (1 - F_X(\sqrt{y})) \\ &= 2F_X(\sqrt{y}) - 1 \end{aligned}$$

Therefore, the pdf of $Y$ for $y \geq 0$ is

$$\begin{aligned} f_Y(y) &= \frac{\partial}{\partial y}\left[2F_X(\sqrt{y}) - 1\right] \\ &= 2\frac{\partial}{\partial y}F_X(\sqrt{y}) - 0 \\ &= 2\frac{\partial}{\partial y}\left[\int_{-\infty}^{\sqrt{y}} \frac{1}{\sqrt{2\pi}} \exp[-t^2/2]\, \mathrm{d}t\right] \\ &= 2\frac{1}{\sqrt{2\pi}}\left[\exp[-y/2]\frac{\partial}{\partial y}\sqrt{y} - 0\right] \\ &= 2\frac{1}{\sqrt{2\pi}} \exp[-y/2] \cdot \frac{1}{2}y^{-1/2} \\ &= \frac{1}{2^{1/2}\Gamma(1/2)}y^{1/2-1} \exp[-y/2] \end{aligned}$$

and 0 otherwise, which shows that $Y \sim \Gamma(1/2, 2) \equiv \chi^2(1)$.

### A.2.4 Sum of Gamma variables with same scale is again Gamma

Suppose $Y_i \sim \Gamma(k_i, \theta)$, for $i = 1, 2, \ldots, n$. Then, the characteristic function of $Y_i$ is

$$\phi_i(t) = (1 - it\theta)^{k_i}.$$

If $Y = \sum_{i=1}^n Y_i$, the characteristic function of $Y$ is

$$\phi(t) = \prod_{i=1}^n \phi_i(t) = \prod_{i=1}^n (1 - it\theta)^{k_i} = (1 - it\theta)^{\sum_{i=1}^n k_i},$$

which implies that

$$Y = \sum_{i=1}^n Y_i \sim \Gamma\left(\sum_{i=1}^n k_i, \theta\right).$$

## A.2.5 The cumulants of a chi-squared random variable

Suppose

$$Y \sim \Gamma(k,\theta), \qquad f_Y(y;k,\theta) = \frac{1}{\Gamma(k)\theta^k} y^{k-1} e^{-y/\theta}.$$

It is then straightforward to show

$$\mathbb{M}_Y(t) = \mathrm{E}[e^{tY}] = \int_0^\infty e^{ty} f_Y(y;k,\theta)\,\mathrm{d}y = (1-\theta t)^{-k},$$

for $t < 1/\theta$. If $X \sim \chi_1^2$, which is the same as saying $X \sim \Gamma(1/2, 2)$, then

$$\mathbb{M}_X(t) = (1-2t)^{-1/2},$$

which is defined for $t < 1/2$. The cumulant generating function of $X$ is then

$$\mathbb{K}_X(t) = -\frac{1}{2}\log(1-2t), \qquad t < \frac{1}{2}.$$

Calculating its derivatives with respect to $t$:

$$\begin{aligned}
\mathbb{K}_X'(t) &= -\frac{1}{2}\frac{-2}{1-2t}\\
\mathbb{K}_X''(t) &= -\frac{1}{2}\frac{(-2)^2}{(1-2t)^2}(-1)\\
\mathbb{K}_X^{(3)}(t) &= -\frac{1}{2}\frac{(-2)^3}{(1-2t)^3}(-1)(-2)\\
&\vdots\\
\mathbb{K}_X^{(r)}(t) &= -\frac{1}{2}\frac{(-2)^r}{(1-2t)^r}(-1)(-2)\dots(-(r-1))\\
&= (-1)\cdot\frac{1}{2}\cdot(-1)^r\cdot 2^r\cdot\left(\frac{1}{(1-2t)^r}\right)\cdot(-1)^{r-1}\cdot(r-1)!\\
&= 2^{r-1}\cdot\left(\frac{1}{(1-2t)^r}\right)\cdot(r-1)!\cdot(-1)^{2r}\\
\Rightarrow \mathbb{K}_X^{(r)}(0) &= 2^{r-1}(r-1)!
\end{aligned}$$

Then, by definition, the $r$-th cumulant of a $\chi_1^2$ random variable $X$ is simply

$$\begin{aligned}\kappa_r(X) &= \frac{\mathrm{d}^r}{\mathrm{d}t^r}\mathbb{K}_X(t)\bigg|_{t=0} \\ &= \mathbb{K}_X^{(r)}(0) \\ \Rightarrow \kappa_r(X) &= 2^{r-1}(r-1)! \end{aligned} \tag{A.12}$$

## A.3 Derivations for the adaptive forgetting factor mean

### A.3.1 Non-sequential definitions for $\bar{x}_{N,\overrightarrow{\lambda}}$

The adaptive forgetting factor mean is defined as

$$\bar{x}_{N,\overrightarrow{\lambda}} = \frac{m_{N,\overrightarrow{\lambda}}}{w_{N,\overrightarrow{\lambda}}}$$

where the update equation for $m_{N,\overrightarrow{\lambda}}$ is defined by

$$m_{N,\overrightarrow{\lambda}} = \lambda_{N-1} m_{N-1,\overrightarrow{\lambda}} + x_N. \tag{A.13}$$

Similarly,

$$w_{N,\overrightarrow{\lambda}} = \lambda_{N-1} w_{N-1,\overrightarrow{\lambda}} + 1. \tag{A.14}$$

The non-sequential definition of $m_{N,\overrightarrow{\lambda}}$ is now obtained for a sequence of observations $x_1, x_2, \ldots, x_N$. The easiest method to do this is to observe that the function

$$f(N) \equiv f(N, \lambda_1, \ldots, \lambda_{N-1}, x_1, \ldots, x_N) = \sum_{k=1}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right] \tag{A.15}$$

satisfies the relation

$$f(N) = \lambda_{N-1} f(N-1) + x_N. \tag{A.16}$$

This is shown by:

$$\begin{aligned}
f(N) &= \sum_{k=1}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right]\\
&= \sum_{k=1}^{N-1}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right] + \sum_{k=N}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right]\\
&= \sum_{k=1}^{N-1}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right] + \Big(\prod_{p=N}^{N-1}\lambda_p\Big)x_N\\
&= \sum_{k=1}^{N-1}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right] + x_N\\
&= \sum_{k=1}^{N-1}\left[\lambda_{N-1}\Big(\prod_{p=k}^{N-2}\lambda_p\Big)x_k\right] + x_N\\
&= \lambda_{N-1}\sum_{k=1}^{N-1}\left[\Big(\prod_{p=k}^{N-2}\lambda_p\Big)x_k\right] + x_N\\
&= \lambda_{N-1}f(N-1) + x_N
\end{aligned}$$

Note that we have used the fact that the empty product is 1, i.e.

$$\prod_{k=M}^{M-1}(x_k) = 1.$$

Therefore, $f(N)$ agrees with $m_{N,\overrightarrow{\lambda}}$, and so

$$m_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right]. \tag{A.17}$$

Similarly, we obtain

$$w_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}\lambda_p\Big). \tag{A.18}$$

### A.3.2 Alternate non-sequential equation of $\Delta_{N,\overrightarrow{\lambda}}$

Recall from Section 3.3.2 that

$$\Delta_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] \tag{A.19}$$

Equation (A.19) is equivalent to:

$$\Delta_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N-1} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg], \tag{A.20}$$

(where the first summation over $k$ runs over $k = 1, \dots, N$ or $k = 1, \dots, N-1$) since

$$\begin{aligned}
\sum_{k=1}^{N} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] &= \sum_{k=1}^{N-1} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] + \sum_{k=N}^{N} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] \\
&= \sum_{k=1}^{N-1} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] + \Bigg[ \sum_{t=N}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] \\
&= \sum_{k=1}^{N-1} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] + 0 \\
\Rightarrow \sum_{k=1}^{N} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] &= \sum_{k=1}^{N-1} \Bigg[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N-1} \lambda_p \Big) x_k \Bigg] \qquad \text{(A.21)}
\end{aligned}$$

where we have used the fact that an empty summation

$$\sum_{t=N+1}^{N} \alpha_t = 0 \tag{A.22}$$

(for example, a sum from a higher to a lower index) must be zero.

### A.3.3 Proof of Lemma 1

We now prove the lemma

**Lemma 1**
$$\prod_{p=k}^{N}\big(\lambda_p+\epsilon\big)=\prod_{p=k}^{N}\lambda_p+\epsilon\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)+O(\epsilon^2)$$

using induction. First, prove the lemma is true for $N=k$:

$$\begin{aligned}
\text{LHS}&=\prod_{p=k}^{k}\big(\lambda_p+\epsilon\big)\\
&=\lambda_k+\epsilon\\
&=\lambda_k+\epsilon+O(\epsilon^2)\\
\text{RHS}&=\prod_{p=k}^{k}\lambda_p+\epsilon\left(\sum_{t=k}^{k}\Big(\prod_{\substack{p=k\\p\neq t}}^{k}\lambda_p\Big)\right)+O(\epsilon^2)\\
&=\lambda_k+\epsilon\left(\Big(\prod_{\substack{p=k\\p\neq k}}^{k}\lambda_p\Big)\right)+O(\epsilon^2)\\
&=\lambda_k+\epsilon\Big(1\Big)+O(\epsilon^2)\\
&=LHS
\end{aligned}\tag{A.23}$$

Note that for the LHS, we used $x=x+O(\epsilon^2)$, and for the RHS, we again used the property of product notation that an "empty product" is 1, and not zero. Next, assume the lemma holds for $N=M$, i.e.

$$\prod_{p=k}^{M}\big(\lambda_p+\epsilon\big)=\prod_{p=k}^{M}\lambda_p+\epsilon\left(\sum_{t=k}^{M}\Big(\prod_{\substack{p=k\\p\neq t}}^{M}\lambda_p\Big)\right)+O(\epsilon^2)\tag{A.24}$$

To complete the inductive proof, we prove the lemma true for $N = M + 1$.

$$\begin{aligned}
\text{LHS} &= \prod_{p=k}^{M+1} \big(\lambda_p + \epsilon\big) \\
&= \big(\lambda_{M+1} + \epsilon\big) \prod_{p=k}^{M} \big(\lambda_p + \epsilon\big) \\
&\text{(and using (A.24) )} \\
&= \big(\lambda_{M+1} + \epsilon\big) \left[ \prod_{p=k}^{M} \lambda_p + \epsilon \left( \sum_{t=k}^{M} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M} \lambda_p \Big) \right) + O(\epsilon^2) \right] \\
&= \lambda_{M+1} \prod_{p=k}^{M} \lambda_p + \epsilon\lambda_{M+1} \left( \sum_{t=k}^{M} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M} \lambda_p \Big) \right) + \epsilon \prod_{p=k}^{M} \lambda_p + O(\epsilon^2) \\
&= \prod_{p=k}^{M+1} \lambda_p + \epsilon \left( \sum_{t=k}^{M} \lambda_{M+1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M} \lambda_p \Big) \right) + \epsilon \prod_{p=k}^{M} \lambda_p + O(\epsilon^2) \\
&= \prod_{p=k}^{M+1} \lambda_p + \epsilon \left( \sum_{t=k}^{M} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M+1} \lambda_p \Big) + \prod_{p=k}^{M} \lambda_p \right) + O(\epsilon^2) \\
&= \prod_{p=k}^{M+1} \lambda_p + \epsilon \left( \sum_{t=k}^{M} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M+1} \lambda_p \Big) + \prod_{\substack{p=k \\ p\neq M+1}}^{M+1} \lambda_p \right) + O(\epsilon^2) \\
&= \prod_{p=k}^{M+1} \lambda_p + \epsilon \left( \sum_{t=k}^{M+1} \Big( \prod_{\substack{p=k \\ p\neq t}}^{M+1} \lambda_p \Big) \right) + O(\epsilon^2)
\end{aligned}$$

which proves the lemma true for $N = M + 1$, and so proves the lemma by induction.

A very convenient form of the lemma is

$$\prod_{p=k}^{N} \big(\lambda_p + \epsilon\big) - \prod_{p=k}^{N} \lambda_p = \epsilon \left( \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p\neq t}}^{N} \lambda_p \Big) \right) + O(\epsilon^2) \qquad \text{(A.25)}$$

## A.3.4 Sequential update equation for $\Delta_{N,\overrightarrow{\lambda}}$

We now prove the update equation $\Delta_{N+1,\overrightarrow{\lambda}} = \lambda_N \Delta_{N,\overrightarrow{\lambda}} + m_{N,\lambda}$.

$$
\begin{aligned}
\text{LHS} = \Delta_{N+1,\overrightarrow{\lambda}} &= \sum_{k=1}^{N+1} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) x_k \right] \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) x_k \right]
\end{aligned}
$$

$$
\begin{aligned}
\text{RHS} &= \lambda_N \Delta_{N,\overrightarrow{\lambda}} + m_{N,\lambda} \\
&= \lambda_N \sum_{k=1}^{N} \left[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) x_k \right] + \sum_{k=1}^{N} \left[ \Big( \prod_{p=k}^{N-1} \lambda_p \Big) x_k \right] \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N-1} \lambda_N \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) x_k \right] + \sum_{k=1}^{N} \left[ \Big( \prod_{p=k}^{N-1} \lambda_p \Big) x_k \right] \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) x_k \right] + \sum_{k=1}^{N} \left[ \Big( \prod_{p=k}^{N-1} \lambda_p \Big) x_k \right] \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) + \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \right] x_k \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) - \sum_{t=N}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) + \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \right] x_k \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) - \Big( \prod_{\substack{p=k \\ p \neq N}}^{N} \lambda_p \Big) + \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \right] x_k \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) - \Big( \prod_{p=k}^{N-1} \lambda_p \Big) + \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \right] x_k \\
&= \sum_{k=1}^{N} \left[ \sum_{t=k}^{N} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N} \lambda_p \Big) \right] x_k \\
&= \text{LHS}
\end{aligned}
$$

which proves the relation

$$\Delta_{N+1,\overrightarrow{\lambda}} = \lambda_N \Delta_{N,\overrightarrow{\lambda}} + m_{N,\lambda}.$$

## A.3.5 Non-sequential and sequential definitions for $\Omega_{N,\overrightarrow{\lambda}}$

By following the derivation of $\Delta_{N,\overrightarrow{\lambda}}$ in Section 3.3.2 and Appendix A.3.2, and by setting $x_k = 1$ for $k = 1, \ldots, N$, we can similarly derive the non-sequential equations for $\Omega_{N,\overrightarrow{\lambda}}$ to be

$$\frac{\partial}{\partial \overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}} = \Omega_{N,\overrightarrow{\lambda}}$$

$$\Omega_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) \tag{A.26}$$

$$\Omega_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N-1} \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) \tag{A.27}$$

Similarly, following the derivation for the update equation for $\Delta_{N,\overrightarrow{\lambda}}$ in Appendix A.3.4, the sequential update equation for $\Omega_{N+1,\overrightarrow{\lambda}}$

$$\Omega_{N+1,\overrightarrow{\lambda}} = \lambda_N \Omega_{N,\overrightarrow{\lambda}} + w_{N,\overrightarrow{\lambda}}$$

## A.3.6 Non-sequential definition for $u_{N,\overrightarrow{\lambda}}$

Recalling from Appendix A.3.1 that for a sequence $X_1, X_2, \ldots, X_N$,

$$m_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \Big[ \Big( \prod_{p=k}^{N-1} \lambda_p \Big) X_k \Big].$$

Assuming that the $X_i$ are independent, and $\text{Var}[X_i] = \sigma^2$, the variance of $\bar{X}_{N,\overrightarrow{\lambda}}$ is first computed to show:

$$\begin{aligned}
\text{Var}\big[\bar{X}_{N,\overrightarrow{\lambda}}\big] &= \text{Var}\left[\frac{m_{N,\overrightarrow{\lambda}}}{w_{N,\overrightarrow{\lambda}}}\right] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \text{Var}\Big[m_{N,\overrightarrow{\lambda}}\Big] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \text{Var}\left[\sum_{k=1}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right]\right] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \sum_{k=1}^{N}\text{Var}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)x_k\right] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \sum_{k=1}^{N}\left[\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right]^2 \text{Var}\big[x_k\big] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}(\lambda_p)^2\Big)\sigma^2
\end{aligned}$$

By defining,

$$u_{N,\overrightarrow{\lambda}} = \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}(\lambda_p)^2\Big)$$

we have

$$\text{Var}\big[\bar{x}_{N,\overrightarrow{\lambda}}\big] = u_{N,\overrightarrow{\lambda}}\sigma^2.$$

### A.3.7 Sequential update equation for $u_{N,\overrightarrow{\lambda}}$

In Appendix A.3.6 it is shown that

$$u_{N,\overrightarrow{\lambda}} = \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\right)^2 \sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}(\lambda_p)^2\Big), \tag{A.28}$$

now define

$$\begin{aligned}\tilde{u}_{N,\overrightarrow{\lambda}} &= \sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}(\lambda_p)^2\Big),\\ \Rightarrow u_{N,\overrightarrow{\lambda}}\big(w_{N,\overrightarrow{\lambda}}\big)^2 &= \tilde{u}_{N,\overrightarrow{\lambda}}\end{aligned}$$

Now,

$$\begin{aligned}\tilde{u}_{N+1,\overrightarrow{\lambda}} &= \sum_{k=1}^{N+1}\Big(\prod_{p=k}^{N}(\lambda_p)^2\Big)\\ &= \sum_{k=1}^{N}\Big(\prod_{p=k}^{N}(\lambda_p)^2\Big) + \Big(\prod_{p=N+1}^{N}(\lambda_p)^2\Big)\\ &= \sum_{k=1}^{N}\Big(\prod_{p=k}^{N}(\lambda_p)^2\Big) + 1\\ &= (\lambda_N)^2\sum_{k=1}^{N}\Big(\prod_{p=k}^{N-1}(\lambda_p)^2\Big) + 1\\ &= (\lambda_N)^2\tilde{u}_{N,\overrightarrow{\lambda}} + 1\end{aligned}$$

And so,

$$\begin{aligned}u_{N+1,\overrightarrow{\lambda}} &= \left(\frac{1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2\tilde{u}_{N+1,\overrightarrow{\lambda}}\\ &= \left(\frac{1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2\Big[(\lambda_N)^2\tilde{u}_{N,\overrightarrow{\lambda}} + 1\Big]\\ &= \left(\frac{1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2\Big[(\lambda_N)^2u_{N,\overrightarrow{\lambda}}\big(w_{N,\overrightarrow{\lambda}}\big)^2 + 1\Big]\\ &= \left(\frac{\lambda_N w_{N,\overrightarrow{\lambda}}}{w_{N+1,\overrightarrow{\lambda}}}\right)^2 u_{N,\overrightarrow{\lambda}} + \left(\frac{1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2\\ \Rightarrow u_{N+1,\overrightarrow{\lambda}} &= \left(\frac{w_{N+1,\overrightarrow{\lambda}}-1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2 u_{N,\overrightarrow{\lambda}} + \left(\frac{1}{w_{N+1,\overrightarrow{\lambda}}}\right)^2\end{aligned}$$

When $N = 1$, then from Equation (A.28) it can be seen that $u_{1,\overrightarrow{\lambda}} = 1$.

### A.3.8 Expectation of cost function $L_{N,\overrightarrow{\lambda}}$

Recall the definition of the cost function $L_{N,\overrightarrow{\lambda}}$ from Equation (3.26),

$$L_{N,\overrightarrow{\lambda}} = \left[X_N - \bar{X}_{N,\overrightarrow{\lambda}}\right]^2$$

Since $X_N$ and $\bar{X}_{N,\overrightarrow{\lambda}}$ are independent, the expectation of the cost function $L_{N,\overrightarrow{\lambda}}$ is:

$$\begin{aligned}
\mathrm{E}\left[L_{N,\overrightarrow{\lambda}}\right] &= \mathrm{E}\left[\left[X_N - \bar{X}_{N,\overrightarrow{\lambda}}\right]^2\right] \\
&= \mathrm{E}\left[X_N^2 - 2X_N\bar{X}_{N-1,\overrightarrow{\lambda}} + \left(\bar{X}_{N-1,\overrightarrow{\lambda}}\right)^2\right] \\
&= \mathrm{E}\left[X_N^2\right] - 2\mathrm{E}\left[X_N\bar{X}_{N-1,\overrightarrow{\lambda}}\right] + \mathrm{E}\left[\left(\bar{X}_{N-1,\overrightarrow{\lambda}}\right)^2\right] \\
&= \left(\mathrm{Var}[X_N] + (\mathrm{E}[X_N])^2\right) - 2\mathrm{E}\left[X_N\right]\mathrm{E}\left[\bar{X}_{N-1,\overrightarrow{\lambda}}\right] + \\
&\quad \left(\mathrm{Var}[\bar{X}_{N-1,\overrightarrow{\lambda}}] + \left(\mathrm{E}[\bar{X}_{N-1,\overrightarrow{\lambda}}]\right)^2\right) \\
&= \left(\sigma^2 + \mu^2\right) - 2\mu\cdot\mu + \left(u_{N-1,\overrightarrow{\lambda}}\sigma^2 + \mu^2\right) \\
\Rightarrow \mathrm{E}\left[L_{N,\overrightarrow{\lambda}}\right] &= \sigma^2\left(u_{N-1,\overrightarrow{\lambda}} + 1\right) \qquad \text{(A.29)}
\end{aligned}$$

which shows that $\mathrm{E}\left[L_{N,\overrightarrow{\lambda}}\right] \sim O(\sigma^2)$.

### A.3.9 Expectation of the derivative of the cost function $L_{N,\overrightarrow{\lambda}}$

We expect the derivative of $L_{N+1,\overrightarrow{\lambda}}$ to also be $O(\sigma^2)$, but investigate this formally. The derivative is:

$$\begin{aligned}
\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}} &= 2\left[x_{N+1} - \bar{x}_{N,\overrightarrow{\lambda}}\right] \cdot (-1) \cdot \frac{\partial}{\partial \overrightarrow{\lambda}} \bar{x}_{N,\overrightarrow{\lambda}} \\
&= 2\left[\bar{x}_{N,\overrightarrow{\lambda}} - x_{N+1}\right] \left( \frac{\Delta_{N,\overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}} - m_{N,\overrightarrow{\lambda}} \Omega_{N,\overrightarrow{\lambda}}}{\left(w_{N,\overrightarrow{\lambda}}\right)^2} \right) \\
&= \frac{2}{w_{N,\overrightarrow{\lambda}}} \left[\bar{X}_{N,\overrightarrow{\lambda}} - x_{N+1}\right] \left(\Delta_{N,\overrightarrow{\lambda}} - \bar{X}_{N,\overrightarrow{\lambda}} \Omega_{N,\overrightarrow{\lambda}}\right) \\
&= \frac{2}{w_{N,\overrightarrow{\lambda}}} \Big[\bar{X}_{N,\overrightarrow{\lambda}} \Delta_{N,\overrightarrow{\lambda}} - X_{N+1} \Delta_{N,\overrightarrow{\lambda}} \\
&\qquad - (\bar{x}_{N,\overrightarrow{\lambda}})^2 \Omega_{N,\overrightarrow{\lambda}} + X_{N+1} \bar{x}_{N,\overrightarrow{\lambda}} \Omega_{N,\overrightarrow{\lambda}}\Big]
\end{aligned} \tag{A.30}$$

The expectation of $(w_{N,\overrightarrow{\lambda}}/2)\frac{\partial}{\partial \overrightarrow{\lambda}} L_{N+1,\overrightarrow{\lambda}}$ is split into four terms:

$$\mathrm{E}\left[\bar{X}_{N,\overrightarrow{\lambda}} \Delta_{N,\overrightarrow{\lambda}}\right] \tag{A.31}$$

$$\mathrm{E}\left[X_{N+1} \Delta_{N,\overrightarrow{\lambda}}\right] \tag{A.32}$$

$$\mathrm{E}\left[(\bar{X}_{N,\overrightarrow{\lambda}})^2 \Omega_{N,\overrightarrow{\lambda}}\right] \tag{A.33}$$

$$\mathrm{E}\left[X_{N+1} \bar{X}_{N,\overrightarrow{\lambda}} \Omega_{N,\overrightarrow{\lambda}}\right] \tag{A.34}$$

The last three terms are easily calculated. The second term (Equation (A.32)) is:

$$\begin{aligned}
\mathrm{E}\left[X_{N+1}\Delta_{N,\overrightarrow{\lambda}}\right] &= \mathrm{E}[X_{N+1}]\mathrm{E}[\Delta_{N,\overrightarrow{\lambda}}] \\
&= \mu \cdot \mathrm{E}\left[\sum_{k=1}^{N-1}\left[\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)x_k\right]\right] \\
&= \mu \cdot \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\mathrm{E}\left[x_k\right] \\
&= \mu \cdot \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\mu \\
&= \mu^2 \cdot \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big) \\
\mathrm{E}\left[X_{N+1}\Delta_{N,\overrightarrow{\lambda}}\right] &= \mu^2\Omega_{N,\overrightarrow{\lambda}} \qquad \text{(A.35)}
\end{aligned}$$

The third term (Equation (A.33)) is:

$$\begin{aligned}
\mathrm{E}\left[(\bar{x}_{N,\overrightarrow{\lambda}})^2\Omega_{N,\overrightarrow{\lambda}}\right] &= \Omega_{N,\overrightarrow{\lambda}}\mathrm{E}\left[(\bar{x}_{N,\overrightarrow{\lambda}})^2\right] \\
&= \Omega_{N,\overrightarrow{\lambda}}\left[\mathrm{Var}\left[\bar{x}_{N,\overrightarrow{\lambda}}\right] + \left(\mathrm{E}\left[\bar{x}_{N,\overrightarrow{\lambda}}\right]\right)^2\right] \\
\Rightarrow \mathrm{E}\left[(\bar{x}_{N,\overrightarrow{\lambda}})^2\Omega_{N,\overrightarrow{\lambda}}\right] &= \Omega_{N,\overrightarrow{\lambda}}\left[u_{N,\lambda}\sigma^2 + \mu^2\right] \qquad \text{(A.36)}
\end{aligned}$$

The fourth term (Equation (A.34)) is:

$$\begin{aligned}
\mathrm{E}\left[x_{N+1}\bar{x}_{N,\overrightarrow{\lambda}}\Omega_{N,\overrightarrow{\lambda}}\right] &= \Omega_{N,\overrightarrow{\lambda}}\mathrm{E}\left[x_{N+1}\right]\mathrm{E}\left[\bar{x}_{N,\overrightarrow{\lambda}}\right] \\
&= \Omega_{N,\overrightarrow{\lambda}}\mu \cdot \mu \\
\Rightarrow \mathrm{E}\left[x_{N+1}\bar{x}_{N,\overrightarrow{\lambda}}\Omega_{N,\overrightarrow{\lambda}}\right] &= \mu^2\Omega_{N,\overrightarrow{\lambda}} \qquad \text{(A.37)}
\end{aligned}$$

The first term takes some work, but we start with:

$$\mathrm{E}\left[\bar{x}_{N,\vec{\lambda}}\Delta_{N,\vec{\lambda}}\right] = \frac{1}{w_{N,\vec{\lambda}}}\mathrm{E}\left[m_{N,\vec{\lambda}}\Delta_{N,\vec{\lambda}}\right]$$

We recall from Equations (A.17) and (A.20) that

$$\begin{aligned} m_{N,\vec{\lambda}} &= \sum_{j=1}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_j \\ \Delta_{N,\vec{\lambda}} &= \sum_{k=1}^{N-1}\left[\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)x_k\right] \end{aligned}$$

Before proceeding further, define a new quantity $\psi_{N,\vec{\lambda}}$ by

$$\psi_{N,\vec{\lambda}} = \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\left(\prod_{q=k}^{N-1}\lambda_q\right). \tag{A.38}$$

$$= \sum_{k=1}^{N}\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\left(\prod_{q=k}^{N-1}\lambda_q\right) \tag{A.39}$$

Equation (A.38) and (A.39) because $k$ can sum to either $N$ or $N-1$, and the result is the same (an "empty" sum will be zero). However, it will be convenient to have both forms available. Returning to the computation of the first term, we find:

$$\begin{aligned} \mathrm{E}\left[\Delta_{N,\vec{\lambda}}m_{N,\vec{\lambda}}\right] &= \mathrm{E}\left[\left(\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)x_k\right)\left(\sum_{j=1}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_j\right)\right] \\ &= \mathrm{E}\left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{j=1}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_kx_j\right], \end{aligned}$$

which we need to split up into two cases, $j = k$ and $j \neq k$. Therefore,

$$
\begin{aligned}
\mathrm{E}\left[\Delta_{N,\vec{\lambda}} m_{N,\vec{\lambda}}\right] &= \mathrm{E}\Bigg[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j=k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_k x_j + \\
&\quad \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_k x_j\Bigg] \\
&= \mathrm{E}\left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\left(\prod_{q=k}^{N-1}\lambda_q\right)x_k^2\right] + \\
&\quad \mathrm{E}\left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)x_k x_j\right] \\
&= \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\left(\prod_{q=k}^{N-1}\lambda_q\right)\mathrm{E}\left[x_k^2\right] + \\
&\quad \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)\mathrm{E}\left[x_k\right]\mathrm{E}\left[x_j\right] \\
&= \left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\left(\prod_{q=k}^{N-1}\lambda_q\right)\right]\left(\mu^2+\sigma^2\right) + \\
&\quad \left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)\right]\mu^2 \\
&= \psi_{N,\vec{\lambda}}\left(\mu^2+\sigma^2\right) + \\
&\quad \left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)\right]\mu^2
\end{aligned}
\tag{A.40}
$$

We look at the coefficient of the second term ($\mu^2$) in the sum:

$$\begin{aligned}
\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j\neq k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right) &= \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{j=1}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right) - \\
&\quad \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\sum_{\substack{j=1\\j=k}}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right) \\
&= \left[\sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right]\left[\sum_{j=1}^{N}\left(\prod_{q=j}^{N-1}\lambda_q\right)\right] - \\
&\quad \sum_{k=1}^{N-1}\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\left(\prod_{q=k}^{N-1}\lambda_q\right) \qquad \text{(A.41)} \\
&= \Omega_{N,\vec{\lambda}}w_{N,\vec{\lambda}} - \psi_{N,\vec{\lambda}} \qquad \text{(A.42)}
\end{aligned}$$

using Equations (A.27) and (A.18) for $\Omega_{N,\vec{\lambda}}$ and $w_{N,\vec{\lambda}}$, respectively, for the first term, and Equation (A.39) for the second term in Equation (A.41). Substituting the expression in Equation (A.42) into Equation (A.40), we obtain an expression for Equation (A.32):

$$\begin{aligned}
\mathrm{E}\left[\Delta_{N,\vec{\lambda}}m_{N,\vec{\lambda}}\right] &= \psi_{N,\vec{\lambda}}\left(\mu^2+\sigma^2\right) + \left(\Omega_{N,\vec{\lambda}}w_{N,\vec{\lambda}} - \psi_{N,\vec{\lambda}}\right)\mu^2 \\
&= \psi_{N,\vec{\lambda}}\sigma^2 + \Omega_{N,\vec{\lambda}}w_{N,\vec{\lambda}}\mu^2 \\
\Rightarrow \mathrm{E}\left[\Delta_{N,\vec{\lambda}}\bar{x}_{N,\vec{\lambda}}\right] &= \frac{1}{w_{N,\vec{\lambda}}}\psi_{N,\vec{\lambda}}\sigma^2 + \Omega_{N,\vec{\lambda}}\mu^2 \qquad \text{(A.43)}
\end{aligned}$$

Adding the four terms in Equations (A.35), (A.36), (A.37) and (A.43) together, we have (taking the signs into account):

$$\begin{aligned}
\mathrm{E}\left[\left(\frac{w_{N,\overrightarrow{\lambda}}}{2}\right)\frac{\partial}{\partial\overrightarrow{\lambda}}L_{N+1,\overrightarrow{\lambda}}\right] &= \mathrm{E}\left[\bar{x}_{N,\overrightarrow{\lambda}}\Delta_{N,\overrightarrow{\lambda}}\right] - \mathrm{E}\left[x_{N+1}\Delta_{N,\overrightarrow{\lambda}}\right] \\
&\quad - \mathrm{E}\left[(\bar{x}_{N,\overrightarrow{\lambda}})^2\Omega_{N,\overrightarrow{\lambda}}\right] + \mathrm{E}\left[x_{N+1}\bar{x}_{N,\overrightarrow{\lambda}}\Omega_{N,\overrightarrow{\lambda}}\right] \\
&= \left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\psi_{N,\overrightarrow{\lambda}}\sigma^2 + \Omega_{N,\overrightarrow{\lambda}}\mu^2\right) - \left(\mu^2\Omega_{N,\overrightarrow{\lambda}}\right) \\
&\quad - \left(\Omega_{N,\overrightarrow{\lambda}}\left[u_{N,\lambda}\sigma^2 + \mu^2\right]\right) + \left(\mu^2\Omega_{N,\overrightarrow{\lambda}}\right) \\
&= \frac{1}{w_{N,\overrightarrow{\lambda}}}\psi_{N,\overrightarrow{\lambda}}\sigma^2 - \Omega_{N,\overrightarrow{\lambda}}u_{N,\lambda}\sigma^2 \\
\Rightarrow \mathrm{E}\left[\frac{\partial}{\partial\overrightarrow{\lambda}}L_{N+1,\overrightarrow{\lambda}}\right] &= \frac{2}{w_{N,\overrightarrow{\lambda}}}\left(\frac{1}{w_{N,\overrightarrow{\lambda}}}\psi_{N,\overrightarrow{\lambda}} - \Omega_{N,\overrightarrow{\lambda}}u_{N,\lambda}\right)\sigma^2 \qquad \text{(A.44)}
\end{aligned}$$

which shows that, as expected, $\frac{\partial}{\partial\overrightarrow{\lambda}}L_{N+1,\overrightarrow{\lambda}} \sim O(\sigma)^2$. Note that an alternative proof starts by showing

$$\mathrm{E}\left[\frac{\partial}{\partial\overrightarrow{\lambda}}L_{N+1,\overrightarrow{\lambda}}\right] = \left[\frac{\partial}{\partial\overrightarrow{\lambda}}u_{N,\overrightarrow{\lambda}}\right]\sigma^2,$$

which is enough to show $\mathrm{E}\left[\frac{\partial}{\partial\overrightarrow{\lambda}}L_{N+1,\overrightarrow{\lambda}}\right] \sim O(\sigma^2)$, but then arriving at Equation (A.44) is more arduous.

## A.4 Derivations for the optimal fixed forgetting factor

We assume that we have $N$ observations ($D = N - \tau$):

$$x_1, x_2, \ldots, x_\tau, x_{\tau+1}, \ldots, x_{\tau+D}$$

where

$$\begin{aligned}
x_1, x_2, \ldots, x_\tau &\sim \mathrm{N}(\mu_1, \sigma_1^2) \\
x_{\tau+1}, \ldots, x_{\tau+D} &\sim \mathrm{N}(\mu_2, \sigma_2^2)
\end{aligned}$$

We would like to know, given this information, what the optimal value $\widehat{\lambda}_{\tau,N}$ is for each $N$, where we define

$$\widehat{\lambda}_{\tau,N} = \begin{cases} \inf_{\lambda\in[0,1]} \mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_0)^2\big] & \text{for } N \leq \tau, \\ \inf_{\lambda\in[0,1]} \mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_1)^2\big] & \text{for } N > \tau. \end{cases} \tag{A.45}$$

we shall often abbreviate this to

$$\widehat{\lambda}_{\tau,N} = \mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_{0,1,N})^2\big],$$

In other words, we are trying to find the forgetting factor that minimises the (squared-)difference between the forgetting factor mean and the actual mean. Furthermore, we would like to find this value for each $N$. There are two cases, $N \leq \tau$ and $N > \tau$.

## Case 1 :$N \leq \tau$

In the case $N \leq \tau$,

$$\begin{aligned} \mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_{0,1,N})^2\big] &= \mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_1)^2\big] \\ &= \mathrm{E}\big[(\bar{x}_{N,\lambda})^2 - 2\mu_1\bar{x}_{N,\lambda} + \mu_1^2\big] \\ &= \mathrm{E}\big[(\bar{x}_{N,\lambda})^2\big] - 2\mu_1\mathrm{E}\big[\bar{x}_{N,\lambda}\big] + \mathrm{E}\big[\mu_1^2\big] \\ &= \big(\mathrm{Var}\big[\bar{x}_{N,\lambda}\big] + (\mathrm{E}\big[\bar{x}_{N,\lambda}\big])^2\big) - 2\mu_1\mathrm{E}\big[\bar{x}_{N,\lambda}\big] + \mathrm{E}\big[\mu_1^2\big] \\ &= \big(u_{N,\lambda}\sigma_1^2 + \mu_1^2\big) - 2\mu_1(\mu_1) + \mu_1^2 \\ &= \big(u_{N,\lambda}\big)\sigma_1^2 \end{aligned}$$

## Case 2 :$N > \tau$

Now assuming $N > \tau$,

$$\begin{aligned}
(\bar{x}_{N,\lambda} - \mu_2)^2 &= (\bar{x}_{N,\lambda})^2 - 2\mu_2\bar{x}_{N,\lambda} + \mu_2^2 \\
&= A_1 + A_2 + A_3 \\
A_1 &= (\bar{x}_{N,\lambda})^2 \\
A_2 &= -2\mu_2\bar{x}_{N,\lambda} \\
A_3 &= \mu_2^2
\end{aligned}$$

Then,

$$\begin{aligned}
\mathrm{E}\big[(\bar{x}_{N,\lambda} - \mu_2)^2\big] &= \mathrm{E}\big[A_1 + A_2 + A_3\big] \\
&= \mathrm{E}\big[A_1\big] + \mathrm{E}\big[A_2\big] + \mathrm{E}\big[A_3\big]
\end{aligned}$$

And we find these expectations now. First we find $\mathrm{E}\big[\bar{x}_{N,\lambda}\big]$:

$$\begin{aligned}
\mathrm{E}\big[\bar{x}_{N,\lambda}\big] &= \frac{1}{w_{N,\lambda}}\mathrm{E}\left[\sum_{k=1}^{\tau}\lambda^{N-k}x_k + \sum_{k=\tau+1}^{N}\lambda^{N-k}x_k\right] \\
&= \frac{1}{w_{N,\lambda}}\mathrm{E}\left[\lambda^D\sum_{k=1}^{\tau}\lambda^{\tau-k}x_k + \sum_{j=1}^{D}\lambda^{D-k}x_{j+\tau}\right] \\
&= \frac{1}{w_{N,\lambda}}\left(\lambda^D\sum_{k=1}^{\tau}\lambda^{\tau-k}\mu_1 + \sum_{j=1}^{D}\lambda^{D-k}\mu_2\right) \\
\Rightarrow \mathrm{E}\big[\bar{x}_{N,\lambda}\big] &= \frac{1}{w_{N,\lambda}}\left(\lambda^D(w_{\tau,\lambda})\mu_1 + (w_{D,\lambda})\mu_2\right) \qquad \text{(A.46)}
\end{aligned}$$

Now we calculate $\mathrm{E}\big[(\bar{x}_{N,\lambda})^2\big]$:

$$\begin{aligned}
(\bar{x}_{N,\lambda})^2 &= \left(\frac{1}{w_{N,\lambda}}\sum_{k=1}^{N}\lambda^{N-k}x_k\right)^2 \\
&= \left(\frac{1}{w_{N,\lambda}}\right)^2\Big[B_1 + B_2 + B_3\Big] \\
B_1 &= \lambda^{2D}\left(\sum_{k=1}^{\tau}\lambda^{\tau-k}x_k\right)^2 \\
B_2 &= 2\lambda^{D}\left(\sum_{k=1}^{\tau}\lambda^{\tau-k}x_k\right)\left(\sum_{j=1}^{D}\lambda^{D-k}x_{j+\tau}\right) \\
B_3 &= \left(\sum_{j=1}^{D}\lambda^{D-k}x_{j+\tau}\right)^2
\end{aligned}$$

In general:

$$\begin{aligned}
\left(\sum_{k=1}^{M}a_kx_k\right)^2 &= \left(\sum_{k=1}^{M}a_kx_k\right)\left(\sum_{j=1}^{M}a_jx_j\right) \\
&= \sum_{k=1}^{M}a_k^2x_k^2 + \sum_{k=1}^{M}a_kx_k\left(\sum_{\substack{j=1\\ j\neq k}}^{M}a_jx_j\right) \\
&= \sum_{k=1}^{M}a_k^2x_k^2 + \sum_{k=1}^{M}\sum_{\substack{j=1\\ j\neq k}}^{M}a_ka_jx_kx_j
\end{aligned}$$

Now, it can be shown that:

$$\begin{aligned}
\mathrm{E}\left[\left(\sum_{k=1}^{M}a_kx_k\right)^2\right] &= \mathrm{E}\left[\sum_{k=1}^{M}a_k^2x_k^2 + \sum_{k=1}^{M}\sum_{\substack{j=1\\ j\neq k}}^{M}a_ka_jx_kx_j\right] \\
&= \left(\sum_{k=1}^{M}a_k^2\right)(\sigma^2) + \left(\sum_{k=1}^{M}a_k\right)^2(\mu^2)
\end{aligned}$$

If, in particular, we define:

$$\begin{aligned}
a_k &= \lambda^{\tau-k}, \\
\Rightarrow \sum_{k=1}^{\tau} a_k &= \sum_{k=1}^{\tau} \lambda^{\tau-k} \\
&= w_{\tau,\lambda} \\
\Rightarrow \sum_{k=1}^{\tau} a_k^2 &= \sum_{k=1}^{\tau} \left(\lambda^{\tau-k}\right)^2 \\
&= \sum_{k=1}^{\tau} \left(\lambda^2\right)^{\tau-k} \\
&= w_{\tau,\lambda^2}
\end{aligned}$$

Now using the general result above for $\mathrm{E}\big[B_1\big]$:

$$\begin{aligned}
\mathrm{E}\big[B_1\big] &= \mathrm{E}\left[\lambda^{2D}\left(\sum_{k=1}^{\tau} \lambda^{\tau-k} x_k\right)^2\right] \\
&= \lambda^{2D}\left[\left(\sum_{k=1}^{\tau}(\lambda^{\tau-k})^2\right)\sigma_1^2 + \left(\sum_{k=1}^{\tau} \lambda^{\tau-k}\right)^2 \mu_1^2\right] \\
&= \lambda^{2D}\left[\big(w_{\tau,\lambda^2}\big)\sigma_1^2 + \big(w_{\tau,\lambda}\big)^2 \mu_1^2\right]
\end{aligned}$$

Now for $\mathrm{E}\big[B_2\big]$:

$$\begin{aligned}
\mathrm{E}\big[B_2\big] &= \mathrm{E}\left[2\lambda^{D}\left(\sum_{k=1}^{\tau} \lambda^{\tau-k} x_k\right)\left(\sum_{j=1}^{D} \lambda^{D-k} x_{j+\tau}\right)\right] \\
&= 2\lambda^{D}\left[\sum_{k=1}^{\tau}\sum_{j=1}^{D} \lambda^{\tau-k}\lambda^{D-k}\big[\mu_1\big]\big[\mu_2\big]\right] \\
&= 2\lambda^{D}\mu_1\mu_2\left[\left(\sum_{k=1}^{\tau} \lambda^{\tau-k}\right)\left(\sum_{j=1}^{D} \lambda^{D-k}\right)\right] \\
&= 2\lambda^{D}\mu_1\mu_2 w_{\tau,\lambda} w_{D,\lambda}
\end{aligned}$$

Now for $\mathrm{E}\big[B_3\big]$ (will be similar to $\mathrm{E}\big[B_1\big]$):

$$\begin{aligned}\mathrm{E}\big[B_3\big] &= \mathrm{E}\Bigg[\bigg(\sum_{j=1}^{D}\lambda^{D-k}x_{j+\tau}\bigg)^2\Bigg]\\ &= \Big[\big(w_{D,\lambda^2}\big)\sigma_2^2 + \big(w_{D,\lambda}\big)^2\mu_2^2\Big]\end{aligned}$$

We can put this all together, and simplify (completing the square) to get:

$$\begin{aligned}(w_{N,\lambda})^2\cdot\mathrm{E}\big[\big(\bar{x}_{N,\lambda}\big)^2\big] &= \mathrm{E}\Big[B_1 + B_2 + B_3\Big]\\ &= \mathrm{E}\big[B_1\big] + \mathrm{E}\big[B_2\big] + \mathrm{E}\big[B_3\big]\\ &= \lambda^{2D}\Big[\big(w_{\tau,\lambda^2}\big)\sigma_1^2 + \big(w_{\tau,\lambda}\big)^2\mu_1^2\Big] + 2\lambda^D\mu_1\mu_2 w_{\tau,\lambda}w_{D,\lambda} +\\ &\quad + \Big[\big(w_{D,\lambda^2}\big)\sigma_2^2 + \big(w_{D,\lambda}\big)^2\mu_2^2\Big]\\ &= \Big[\lambda^D w_{\tau,\lambda}\mu_1 + w_{D,\lambda}\mu_2\Big]^2 + \Big[\lambda^{2D}w_{\tau,\lambda^2}\sigma_1^2 + w_{D,\lambda^2}\sigma_2^2\Big]\end{aligned}$$

Finally, we have

$$\mathrm{E}\big[\big(\bar{x}_{N,\lambda}\big)^2\big] = \left(\frac{1}{w_{N,\lambda}}\right)^2\Bigg[\Big[\lambda^D w_{\tau,\lambda}\mu_1 + w_{D,\lambda}\mu_2\Big]^2 + \Big[\lambda^{2D}w_{\tau,\lambda^2}\sigma_1^2 + w_{D,\lambda^2}\sigma_2^2\Big]\Bigg] \tag{A.47}$$

$$\begin{aligned}\mathrm{E}[-2\mu_2\bar{x}_{N,\lambda}] &= -2\mu_2\mathrm{E}[\bar{x}_{N,\lambda}]\\ &= -\frac{2\mu_2}{w_{N,\lambda}}\Big(\lambda^D(w_{\tau,\lambda})\mu_1 + (w_{D,\lambda})\mu_2\Big)\end{aligned}$$

$$\mathrm{E}[\mu_2^2] = \mu_2^2$$

Putting these together, and noticing the perfect square, we have

$$\begin{aligned}\mathrm{E}\big[(\bar{x}_{N,\lambda}-\mu_2)^2\big] = &\left[\left(\frac{1}{w_{N,\lambda}}\right)\left(\lambda^D w_{\tau,\lambda}\mu_1 + w_{D,\lambda}\mu_2\right) - \mu_2\right]^2 + \\ &+\left(\frac{1}{w_{N,\lambda}}\right)^2\left[\lambda^{2D} w_{\tau,\lambda^2}\sigma_1^2 + w_{D,\lambda^2}\sigma_2^2\right]\end{aligned}$$

Now, recalling our results from Equations (A.47) and (A.46):

$$\begin{aligned}\mathrm{E}\big[\bar{X}^2_{N,\lambda}\big] &= \left(\frac{1}{w_{N,\lambda}}\right)^2\left[\left[\lambda^D w_{\tau,\lambda}\mu_1 + w_{D,\lambda}\mu_2\right]^2 + \left[\lambda^{2D} w_{\tau,\lambda^2}\sigma_1^2 + w_{D,\lambda^2}\sigma_2^2\right]\right] \\ \mathrm{E}\big[\bar{X}_{N,\lambda}\big] &= \frac{1}{w_{N,\lambda}}\left(\lambda^D (w_{\tau,\lambda})\mu_1 + (w_{D,\lambda})\mu_2\right) \\ \Rightarrow \left(\mathrm{E}\big[\bar{X}_{N,\lambda}\big]\right)^2 &= \left(\frac{1}{w_{N,\lambda}}\right)^2\left(\lambda^D (w_{\tau,\lambda})\mu_1 + (w_{D,\lambda})\mu_2\right)^2\end{aligned}$$

we notice that $(\mathrm{E}[\bar{X}_{N,\lambda}])^2$ is the same as the first term of $\mathrm{E}[\bar{X}^2_{N,\lambda}]$. Therefore,

$$\begin{aligned}\mathrm{Var}\big[\bar{X}_{N,\lambda}\big] &= \mathrm{E}\big[\bar{X}^2_{N,\lambda}\big] - \left(\mathrm{E}\big[\bar{X}_{N,\lambda}\big]\right)^2 \\ \Rightarrow \mathrm{Var}\big[\bar{X}_{N,\lambda}\big] &= \left(\frac{1}{w_{N,\lambda}}\right)^2\left[\lambda^{2D} w_{\tau,\lambda^2}\sigma_1^2 + w_{D,\lambda^2}\sigma_2^2\right] \qquad \text{(A.48)}\end{aligned}$$

which is just the second term of $\mathrm{E}[\bar{X}^2_{N,\lambda}]$.

## A.5 Derivation of equations for Wood F method

In Section 7.3.4 and [159] it was not explicitly shown how Equations (7.5) were obtained. This section provides a sketch of that omitted computation. The G3F distribution [120, 79] $F(\alpha_1, \alpha_2, \beta)$ has density

$$f(z) = \frac{1}{\mathrm{B}(\alpha_1, \alpha_2)} \beta^{\alpha_2} z^{\alpha_1 - 1} (\beta + z)^{-\alpha_1 - \alpha_2}.$$

Therefore, its $r$th moment $m_r = E[z^r]$ can be computed to be

$$m_r = \frac{1}{\mathrm{B}(\alpha_1, \alpha_2)} \beta^{\alpha_2} \int_0^\infty z^{r+\alpha_1-1} (\beta + z)^{-\alpha_1-\alpha_2} \, \mathrm{d}z = \beta^r \frac{\prod_{i=1}^r (\alpha_1 + i - 1)}{\prod_{i=1}^r (\alpha_2 - i)}$$

This gives us three equations

$$m_1 = \beta \frac{\alpha_1}{\alpha_2 - 1} \tag{A.49}$$

$$m_2 = \beta^2 \frac{\alpha_1(\alpha_1 + 1)}{(\alpha_2 - 1)(\alpha_2 - 2)} \tag{A.50}$$

$$m_3 = \beta^2 \frac{\alpha_1(\alpha_1 + 1)(\alpha_1 + 2)}{(\alpha_2 - 1)(\alpha_2 - 2)(\alpha_2 - 3)} \tag{A.51}$$

Now, we want to solve for $(\alpha_1, \alpha_2, \beta)$ in terms of $\{m_1, m_2, m_3\}$. We start by substituting Equation (A.50) into Equation (A.51), and Equation (A.49) into Equation (A.50) to obtain:

$$\begin{aligned} m_1 &= \beta \frac{\alpha_1}{\alpha_2 - 1} \\ m_2 &= m_1 \beta \frac{\alpha_1 + 1}{\alpha_2 - 2} \\ m_3 &= m_2 \beta \frac{\alpha_1 + 2}{\alpha_2 - 3} \end{aligned}$$

and rearranging, to get

$$\begin{aligned}(\alpha_2 - 1)m_1 &= \beta\alpha_1 \\ (\alpha_2 - 2)m_2 &= m_1\beta(\alpha_1 + 1) \\ (\alpha_2 - 3)m_3 &= m_2\beta(\alpha_1 + 2)\end{aligned}$$

Now, it will be more convenient to let

$$\begin{aligned}(\alpha_1, \alpha_2, \beta) &= (x, y, z) \\ (m_1, m_2, m3) &= (a, b, c)\end{aligned}$$

and solve for $(x, y, z)$ in terms of $(a, b, c)$. And so we obtain (after a rearrangement), the system of nonlinear polynomial equations in $[x, y, z]$:

$$\begin{aligned}xz - ay + a &= 0 \\ axz - by + az + 2b &= 0 \\ bxz - cy + bz + 3c &= 0\end{aligned}$$

In order to solve this system of nonlinear equations, we use an algorithm due to Buchberger [21, 22], utilising Gröbner bases. A good introduction to Gröbner bases (and English translations of the original references) can be found in [23]; or see [39]. Maple solves such a nonlinear system using Gröbner bases [57]. An accessible introduction that provides all the details necessary to perform the computation is [2]. Defining $I =< f_1, f_2, f_3 >$ to be the ideal generated by

$$\begin{aligned}f_1 &= xz - ay + a \\ f_2 &= axz - by + az + 2b \\ f_3 &= bxz - cy + bz + 3c\end{aligned}$$

we can show that a Gröbner basis for $I$ with respect to the pure lexicographical order (the reason we use $x, y, z$) is $G =< g_1, g_2, g_3 >$, where

$$\begin{aligned}
g_1 &= xz - ay + a \\
g_2 &= (-a^2 + b)y - az + (a^2 - 2b) \\
g_3 &= z + t_1/t_2 \\
t_1 &= -2a^2c + ab^2 + bc \\
t_2 &= -a^2b + 2b^2 - ac
\end{aligned}$$

Using $g_3$ to reduce $g_2$ to $g_2'$, i.e. $g_2 \longrightarrow_{g_3} g_2'$, and $g_1 \longrightarrow_{\{g_3, g_2'\}} g_1'$,

$$\begin{aligned}
g_1' &= x - \left[2a(ac - b^2)\right]/t_1 \\
g_2' &= y + \left[a^2b + 3ac - 4b^2\right]/t_2 \\
g_3 &= z + t_1/t_2
\end{aligned}$$

These generators will provide $(\alpha_1, \alpha_2, \beta)$ in terms of the moments $(m_1, m_2, m_3)$. However, in order to obtain $(\alpha_1, \alpha_2, \beta)$ in terms of the cumulants $(\kappa_1, \kappa_2, \kappa_3)$, we can use the relations:

$$\begin{aligned}
a &= m_1 = \kappa_1 \\
b &= m_2 = \kappa_2 + \kappa_1^2 \\
c &= m_3 = \kappa_3 + 3\kappa_2\kappa_1 + \kappa_1^3
\end{aligned}$$

which can be obtained from Equation (7.3). Using these relations and simplifying, and replacing $(x, y, z)$ with $(\alpha_1, \alpha_2, \beta)$, we finally obtain

$$\begin{aligned}
\alpha_1 &= 2\kappa_1 \left[\kappa_1^2\kappa_2 + \kappa_1\kappa_3 - \kappa_2^2\right]/r_1 \\
\alpha_2 &= 3 + 2\kappa_2 \left[\kappa_1^2 + \kappa_2\right]/r_2 \\
\beta &= r_1/r_2
\end{aligned}$$

where

$$\begin{aligned} r_1 &= 4\kappa_1\kappa_2^2 + \kappa_3(\kappa_2 - \kappa_1^2) \\ r_2 &= \kappa_1\kappa_3 - 2\kappa_2^2 \end{aligned}$$

which are exactly the equations given in [159].

## A.6 Derivations for the forgetting factor variance

### A.6.1 Derivation of $v_{N,\lambda}$

Suppose we define $S_{N,\lambda}$ by

$$S_{N,\lambda} = \sum_{k=1}^{N} \lambda^{N-k}\big[x_k - \bar{x}_{N,\lambda}\big]^2. \tag{A.52}$$

Recall the definitions of $\bar{x}_{N,\lambda}$ and $w_{N,\lambda}$ from Equations (3.3) and (3.4),

$$\bar{x}_{N,\lambda} = \frac{1}{w_{N,\lambda}} \sum_{k=1}^{N} \lambda^{N-k} x_k, \qquad w_{N,\lambda} = \sum_{k=1}^{N} \lambda^{N-k}$$

the following relation holds:

$$\sum_{k=1}^{N} \lambda^{N-k}\big[x_k - \bar{x}_{N,\lambda}\big]^2 = \sum_{k=1}^{N} \lambda^{N-k} x_k^2 - w_{N,\lambda}\bar{x}_{N,\lambda}^2. \tag{A.53}$$

For the moment, let's assume this relation is true (it will be proved below). Then, assuming the variables $X_1, \ldots, X_N$ are i.i.d. (with $\mathrm{E}[X_k] = \mu$, $\mathrm{Var}[X_k] = \sigma^2$) we derive

$$\begin{aligned}
\mathrm{E}[S_{N,\lambda}] &= \mathrm{E}\bigg[\sum_{k=1}^{N} \lambda^{N-k}\big[X_k - \bar{X}_{N,\lambda}\big]^2\bigg] \\
&= \mathrm{E}\bigg[\sum_{k=1}^{N} \lambda^{N-k} X_k^2 - w_{N,\lambda}\bar{X}_{N,\lambda}^2\bigg] \\
&= \sum_{k=1}^{N} \mathrm{E}\big[\lambda^{N-k} X_k^2\big] - \mathrm{E}\big[w_{N,\lambda}\bar{X}_{N,\lambda}^2\big] \\
&= \sum_{k=1}^{N} \lambda^{N-k} \mathrm{E}\big[X_k^2\big] - w_{N,\lambda}\mathrm{E}\big[\bar{X}_{N,\lambda}^2\big] \\
&= \sum_{k=1}^{N} \lambda^{N-k}\big(\mathrm{Var}[X_k] + (\mathrm{E}[X_k])^2\big) - w_{N,\lambda}\mathrm{E}\big[\bar{X}_{N,\lambda}^2\big] \\
&= \sum_{k=1}^{N} \lambda^{N-k}\big(\sigma^2 + \mu^2\big) - w_{N,\lambda}\mathrm{E}\big[\bar{X}_{N,\lambda}^2\big] \\
&= w_{N,\lambda}\big(\sigma^2 + \mu^2\big) - w_{N,\lambda}\mathrm{E}\big[\bar{X}_{N,\lambda}^2\big] \\
&= w_{N,\lambda}\left(\sigma^2 + \mu^2 - \mathrm{E}\big[\bar{X}_{N,\lambda}^2\big]\right) \\
&= w_{N,\lambda}\left(\sigma^2 + \mu^2 - \mathrm{Var}\big[\bar{X}_{N,\lambda}\big] - \big(\mathrm{E}\big[\bar{X}_{N,\lambda}\big]\big)^2\right) \\
&= w_{N,\lambda}\left(\sigma^2 + \mu^2 - u_{N,\lambda}\sigma^2 - (\mu)^2\right) \\
&= w_{N,\lambda}(1 - u_{N,\lambda})\sigma^2 \\
&= v_{N,\lambda}\sigma^2
\end{aligned}$$

if we define

$$v_{N,\lambda} = w_{N,\lambda}(1 - u_{N,\lambda}). \tag{A.54}$$

Then,

$$\begin{aligned}
\mathrm{E}[s_{N,\lambda}^2] &= \mathrm{E}\bigg[\frac{1}{v_{N,\lambda}} S_{N,\lambda}\bigg] \\
&= \sigma^2
\end{aligned}$$

The relation in Equation (A.53) is shown by:

$$\begin{aligned}
\sum_{k=1}^{N} \lambda^{N-k} \big[x_k - \bar{x}_{N,\lambda}\big]^2 &= \sum_{k=1}^{N} \lambda^{N-k} \big[x_k^2 - 2x_k\bar{x}_{N,\lambda} + \bar{x}_{N,\lambda}^2\big] \\
&= \sum_{k=1}^{N} \lambda^{N-k} x_k^2 - 2\bar{x}_{N,\lambda} \sum_{k=1}^{N} \lambda^{N-k} x_k + \bar{x}_{N,\lambda}^2 \sum_{k=1}^{N} \lambda^{N-k} \\
&= \sum_{k=1}^{N} \lambda^{N-k} x_k^2 - 2\bar{x}_{N,\lambda} m_{N,\lambda} + \bar{x}_{N,\lambda}^2 w_{N,\lambda} \\
&= \sum_{k=1}^{N} \lambda^{N-k} x_k^2 - 2\bar{x}_{N,\lambda} \left(w_{N,\lambda}\bar{x}_{N,\lambda}\right) + \bar{x}_{N,\lambda}^2 w_{N,\lambda} \\
&= \sum_{k=1}^{N} \lambda^{N-k} x_k^2 - w_{N,\lambda}\bar{x}_{N,\lambda}^2
\end{aligned}$$

### A.6.2 Sequential update equation for $s_{N,\lambda}^2$

Recall from Appendix A.6.1 that we can define $s_{N,\lambda}^2$ by

$$s_{N,\lambda}^2 = \frac{1}{v_{N,\lambda}} S_{N,\lambda},$$

where

$$\begin{aligned}
S_{N,\lambda} &= \sum_{k=1}^{N} \lambda^{N-k} \big[x_k - \bar{x}_{N,\lambda}\big]^2 \\
v_{N,\lambda} &= w_{N,\lambda}(1 - u_{N,\lambda}). \qquad \text{(A.55)}
\end{aligned}$$

We already have sequential update equations for $w_{N,\lambda}$ and $u_{N,\lambda}$, so there is no need to find a sequential update equation for $v_{N,\lambda}$ (can simply use Equation (A.55)). A sequential update equation for $S_{N,\lambda}$ can be derived as follows:

$$
\begin{aligned}
S_{N+1,\lambda} &= \sum_{k=1}^{N+1} \lambda^{N+1-k}\big[x_k - \bar{x}_{N+1,\lambda}\big]^2 \\
&= \sum_{k=1}^{N+1} \lambda^{N+1-k} x_k^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \left(\sum_{k=1}^{N+1} \lambda^{N+1-k} x_k^2\right) - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \left(\sum_{k=1}^{N} \lambda^{N+1-k} x_k^2 + x_{N+1}^2\right) - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda\sum_{k=1}^{N} \lambda^{N-k} x_k^2 + x_{N+1}^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda\left(\sum_{k=1}^{N} \lambda^{N-k} x_k^2\right) + x_{N+1}^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda\left(\sum_{k=1}^{N} \lambda^{N-k} x_k^2 - w_{N,\lambda}\bar{x}_{N,\lambda}^2 + w_{N,\lambda}\bar{x}_{N,\lambda}^2\right) + x_{N+1}^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda\left(\sum_{k=1}^{N} \lambda^{N-k} x_k^2 - w_{N,\lambda}\bar{x}_{N,\lambda}^2\right) + \lambda w_{N,\lambda}\bar{x}_{N,\lambda}^2 + x_{N+1}^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda S_{N,\lambda} + (w_{N+1,\lambda} - 1)\bar{x}_{N,\lambda}^2 + x_{N+1}^2 - w_{N+1,\lambda}\bar{x}_{N+1,\lambda}^2 \\
&= \lambda S_{N,\lambda} + (w_{N+1,\lambda} - 1)\bar{x}_{N,\lambda}^2 + x_{N+1}^2 \\
&\quad - w_{N+1,\lambda}\left(\left(\frac{w_{N+1,\lambda} - 1}{w_{N+1,\lambda}}\right)\bar{x}_{N,\lambda} - \frac{1}{w_{N+1}}x_{N+1}\right)^2 \\
&= \lambda S_{N,\lambda} + (w_{N+1,\lambda} - 1)\bar{x}_{N,\lambda}^2 + x_{N+1}^2 \\
&\quad - \frac{1}{w_{N+1,\lambda}}\Big(\,(w_{N+1,\lambda} - 1)\,\bar{x}_{N,\lambda} - x_{N+1}\Big)^2 \\
&= \lambda S_{N,\lambda} + (w_{N+1,\lambda} - 1)\bar{x}_{N,\lambda}^2 + x_{N+1}^2 \\
&\quad - \frac{1}{w_{N+1,\lambda}}\Big(\,(w_{N+1,\lambda} - 1)^2\,\bar{x}_{N,\lambda}^2 - 2\,(w_{N+1,\lambda} - 1)\,\bar{x}_{N,\lambda}x_{N+1} + x_{N+1}^2\Big)
\end{aligned}
$$

Now, the coefficient of the $\bar{x}^2_{N,\lambda}$ term is:

$$\begin{aligned}
(w_{N+1,\lambda}-1)-\frac{(w_{N+1,\lambda}-1)^2}{w_{N+1,\lambda}} &= \frac{1}{w_{N+1,\lambda}}\big(w_{N+1,\lambda}(w_{N+1,\lambda}-1)-(w_{N+1,\lambda}-1)^2\big)\\
&= \frac{1}{w_{N+1,\lambda}}\Big(w^2_{N+1,\lambda}-w_{N+1,\lambda}\\
&\quad -(w^2_{N+1,\lambda}-2w_{N+1,\lambda}+1)\Big)\\
&= \frac{1}{w_{N+1,\lambda}}\Big(-w_{N+1,\lambda}+2w_{N+1,\lambda}-1\Big)\\
&= \frac{w_{N+1,\lambda}-1}{w_{N+1,\lambda}}.
\end{aligned}$$

The coefficient of the $x^2_{N+1}$ term is:

$$1-\frac{1}{w_{N+1,\lambda}}=\frac{w_{N+1,\lambda}-1}{w_{N+1,\lambda}}$$

The coefficient of the $\bar{x}_{N,\lambda}x_{N+1}$ is:

$$-2\left(\frac{w_{N+1,\lambda}-1}{w_{N+1,\lambda}}\right).$$

And so:

$$S_{N+1,\lambda}=\lambda S_{N,\lambda}+\left(\frac{w_{N+1,\lambda}-1}{w_{N+1,\lambda}}\right)\left(\bar{x}^2_{N,\lambda}-2\bar{x}_{N,\lambda}x_{N+1}+x^2_{N+1}\right),$$

which can be factorised to give the update equation

$$S_{N+1,\lambda}=\lambda S_{N,\lambda}+\left(\frac{w_{N+1,\lambda}-1}{w_{N+1,\lambda}}\right)\left(\bar{x}_{N,\lambda}-x_{N+1}\right)^2.$$

### A.6.3 The covariance of the FFF variance terms

Assuming $x_1, x_2, \ldots, x_N$ are sampled from i.i.d. $X_1, X_2, \ldots, X_N \sim \mathrm{N}(\mu, \sigma^2)$. Suppose that for $i = 1, 2, \ldots, N$, $y_i$ is defined as

$$\begin{aligned} y_i &= x_i - \bar{x}_{N,\lambda} \\ &= x_i - \frac{1}{w_{N,\lambda}} \sum_{j=1}^{N} \lambda^{N-j} x_j \end{aligned} \tag{A.56}$$

To compute $\mathrm{Cov}(y_i, y_j)$, first assume $i \neq j$, and define

$$\begin{aligned} y_i &= \sum_{p=1}^{N} a_p x_p \\ y_j &= \sum_{q=1}^{N} b_q x_q \end{aligned}$$

where

$$a_p = \begin{cases} -\frac{1}{w_{N,\lambda}} \lambda^{N-p} & \text{for } p \neq i \\ 1 - \frac{1}{w_{N,\lambda}} \lambda^{N-i} & \text{for } p = i \end{cases}$$

$$b_q = \begin{cases} -\frac{1}{w_{N,\lambda}} \lambda^{N-q} & \text{for } q \neq j \\ 1 - \frac{1}{w_{N,\lambda}} \lambda^{N-j} & \text{for } q = j \end{cases}$$

Now,

$$\begin{aligned}
\mathrm{Cov}(y_i, y_j) &= \mathrm{Cov}\Bigg(\sum_{p=1}^{N} a_p x_p, \sum_{q=1}^{N} b_q x_q\Bigg) \\
&= \sum_{p=1}^{N} a_p \sum_{q=1}^{N} b_q \mathrm{Cov}\Bigg(x_p, x_q\Bigg) \\
&= \sum_{p=1}^{N} \sum_{q=p} a_p b_q \mathrm{Cov}\Bigg(x_p, x_q\Bigg) + \sum_{p=1}^{N} \sum_{q \neq p} a_p b_q \mathrm{Cov}\Bigg(x_p, x_q\Bigg) \\
&= \sum_{p=1}^{N} \sum_{q=p} a_p b_q \mathrm{Cov}\Bigg(x_p, x_q\Bigg) + 0 \\
&= \sum_{p=1}^{N} a_p b_p \mathrm{Cov}\Bigg(x_p, x_p\Bigg) \\
&= \sum_{p=1}^{N} a_p b_p \mathrm{Var}[x_p] \\
&= \Bigg(\sum_{p=1}^{N} a_p b_p\Bigg) \mathrm{Var}[x_p] \\
&= \Bigg(\sum_{p=1}^{N} a_p b_p\Bigg) \sigma^2
\end{aligned}$$

We start with

$$
\begin{aligned}
\sum_{p=1}^{N} a_p b_p &= \sum_{\substack{p=1\\p\neq i\\p\neq j}}^{N} a_p b_p + a_i b_i + a_j b_j \\
&= \sum_{\substack{p=1\\p\neq i\\p\neq j}}^{N} \Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big)\Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big) + \\
&\quad + \Big(1-\frac{1}{w_{N,\lambda}}\lambda^{N-i}\Big)\Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-i}\Big) + \\
&\quad + \Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-j}\Big)\Big(1-\frac{1}{w_{N,\lambda}}\lambda^{N-j}\Big) \\
&= \sum_{\substack{p=1\\p\neq i\\p\neq j}}^{N} \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big)^2 + \\
&\quad - \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-i}\Big) + \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-i}\Big)^2 + \\
&\quad - \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-j}\Big) + \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-j}\Big)^2 \\
&= \sum_{p=1}^{N} \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big)^2 - \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-i}\Big) - \Big(\frac{1}{w_{N,\lambda}}\lambda^{N-j}\Big) \\
&= \Big(\frac{1}{w_{N,\lambda}}\Big)^2 \sum_{p=1}^{N} (\lambda^2)^{N-p} - \Big(\frac{1}{w_{N,\lambda}}\Big)\Big(\lambda^{N-i}+\lambda^{N-j}\Big) \\
&= u_{N,\lambda} - \frac{1}{w_{N,\lambda}}\Big(\lambda^{N-i}+\lambda^{N-j}\Big)
\end{aligned}
$$

Therefore,

$$
\begin{aligned}
\text{Cov}(y_i, y_j) &= \Big(\sum_{p=1}^{N} a_p b_p\Big)\sigma^2 \\
&= \Big(u_{N,\lambda} - \frac{1}{w_{N,\lambda}}\Big(\lambda^{N-i}+\lambda^{N-j}\Big)\Big)\sigma^2
\end{aligned}
$$

The variance of the $y_i$ are computed in the next section.

### A.6.4 The variance of $y_i$

We start by rewriting $y_k$:

$$\begin{aligned}
y_k &= x_k - \bar{x}_{N,\lambda} \\
y_k &= x_k - \frac{1}{w_{N,\lambda}} \sum_{p=1}^{N} \lambda^{N-p} x_p \\
&= \sum_{p=1}^{N} a_p x_p
\end{aligned}$$

where

$$a_p = \begin{cases} -\frac{1}{w_{N,\lambda}} \lambda^{N-p} & \text{for } p \neq k \\ 1 - \frac{1}{w_{N,\lambda}} \lambda^{N-k} & \text{for } p = k \end{cases} \tag{A.57}$$

First, we compute

$$\begin{aligned}
\sum_{p=1}^{N} a_p &= \left(1 - \frac{1}{w_{N,\lambda}} \lambda^{N-k}\right) - \frac{1}{w_{N,\lambda}} \sum_{\substack{p=1 \\ p\neq k}}^{N} \lambda^{N-p} \\
&= 1 - \frac{1}{w_{N,\lambda}} \sum_{p=1}^{N} \lambda^{N-p} \\
&= 1 - \frac{1}{w_{N,\lambda}} (w_{N,\lambda}) \\
\Rightarrow \sum_{p=1}^{N} a_p &= 0
\end{aligned}$$

Next, we compute

$$\begin{aligned}
\sum_{p=1}^{N}(a_p)^2 &= \Big(1-\frac{1}{w_{N,\lambda}}\lambda^{N-k}\Big)^2+\sum_{\substack{p=1\\p\neq k}}^{N}\Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big)^2\\
&= 1-\frac{2}{w_{N,\lambda}}\lambda^{N-k}+\Big(\frac{1}{w_{N,\lambda}}\lambda^{N-k}\Big)^2+\sum_{\substack{p=1\\p\neq k}}^{N}\Big(-\frac{1}{w_{N,\lambda}}\lambda^{N-p}\Big)^2\\
&= 1-\frac{2}{w_{N,\lambda}}\lambda^{N-k}+\Big(\frac{1}{w_{N,\lambda}}\Big)^2\sum_{p=1}^{N}(\lambda^2)^{N-p}\\
&= 1-\frac{2}{w_{N,\lambda}}\lambda^{N-k}+u_{N,\lambda}
\end{aligned}$$

Now,

$$\begin{aligned}
\text{Var}[y_k] &= \text{Var}\Bigg[\sum_{p=1}^{N}a_px_p\Bigg]\\
&= \sum_{p=1}^{N}(a_p)^2\text{Var}\big[x_p\big]\\
&= \Bigg(\sum_{p=1}^{N}(a_p)^2\Bigg)\sigma^2\\
\Rightarrow \text{Var}[y_k] &= \Bigg(1+u_{N,\lambda}-\frac{2}{w_{N,\lambda}}\lambda^{N-k}\Bigg)\sigma^2
\end{aligned}$$

### A.6.5 Computing the cumulants for the FFF variance

Supposing that the observations $x_1, x_2, \ldots, x_N$ are sampled from the random variables $X_1, X_2, \ldots, X_N$, where

$$X_i \sim \text{N}(\mu, \sigma^2) \tag{A.58}$$

and the fixed forgetting factor variance $s^2_{N,\lambda}$

$$s^2_{N,\lambda} = \frac{1}{v_{N,\lambda}}\sum_{i=1}^{N}\lambda^{N-i}\left[x_i-\bar{x}_{N,\lambda}\right]^2 \tag{A.59}$$

and that

$$y_i = x_i - \bar{x}_{N,\lambda}. \tag{A.60}$$

Defining $Y_i = X_i - \bar{X}_{N,\lambda}$, for $i \neq j$, the covariance (from Appendix A.6.3) is

$$\text{Cov}(Y_i, Y_j) = \left( u_{N,\lambda} - \frac{1}{w_{N,\lambda}} \left( \lambda^{N-i} + \lambda^{N-j} \right) \right) \sigma^2 \tag{A.61}$$

and the variance of a single $y_i$ (from Appendix A.6.4) is

$$\text{Var}\left[Y_i\right] = \left( 1 + u_{N,\lambda} - \frac{2}{w_{N,\lambda}} \lambda^{N-i} \right) \sigma^2. \tag{A.62}$$

Note the slight difference between the covariance of diagonal terms (variance) and the off-diagonal terms — an extra term with value $\sigma^2$. The FFF variance could be written in terms of the $y_i$ as

$$s^2_{N,\lambda} = \sum_{i=1}^{N} \frac{\lambda^{N-i}}{v_{N,\lambda}} y_i^2$$

and defining

$$z_i = \sqrt{\frac{\lambda^{N-i}}{v_{N,\lambda}}} y_i$$

the FFF variance can be expressed as

$$s^2_{N,\lambda} = \sum_{i=1}^{N} z_i^2. \tag{A.63}$$

Note that

$$\text{E}[Y_i] = \text{E}\left[X_i - \bar{X}_{N,\lambda}\right] = \text{E}\left[X_i\right] - \text{E}\left[\bar{X}_{N,\lambda}\right] = \mu - \mu = 0, \tag{A.64}$$

and so $\text{E}[z_i] = 0$. Now, the covariance of the $z_i$ variables can be expressed (for $i \neq j$) as

$$\text{Cov}(z_i, z_j) = \frac{1}{v_{N,\lambda}} \sqrt{\lambda^{N-i}} \sqrt{\lambda^{N-j}} \left( u_{N,\lambda} - \frac{1}{w_{N,\lambda}} \left( \lambda^{N-i} + \lambda^{N-j} \right) \right) \sigma^2 \tag{A.65}$$

and for $i = j$,

$$\begin{aligned}\text{Cov}(z_i, z_i) = \text{Var}[z_i] &= \frac{1}{v_{N,\lambda}}\lambda^{N-i}\left(1 + u_{N,\lambda} - \frac{2}{w_{N,\lambda}}\lambda^{N-i}\right)\sigma^2 \\ &= \frac{1}{v_{N,\lambda}}\lambda^{N-i}\left(u_{N,\lambda} - \frac{2}{w_{N,\lambda}}\lambda^{N-i}\right)\sigma^2 + \frac{1}{v_{N,\lambda}}\lambda^{N-i}\sigma^2 \qquad \text{(A.66)}\end{aligned}$$

Unfortunately these $z_i$ are dependent, and it would be preferable to rather deal with independent random variables. The Box-Cochran theorem says that there is a transformation

$$\sum_{i=1}^{N} z_i^2 = \sum_{j=1}^{R} \nu_j \xi_j^2 \qquad \text{(A.67)}$$

where the $\xi_j$ are independent (note that $\text{E}[z_i] = 0$, a requirement of the theorem). The $\nu_j$ are eigenvalues of the covariance matrix $Z$ of the $z_i$ variables, and so for $m = 1, 2, 3, \ldots,$

$$\text{tr}(Z^m) = \sum_{j=1}^{R} \nu_j^m. \qquad \text{(A.68)}$$

This formula is very useful, because in order to compute the $m$th cumulant of the $s^2_{N,\lambda}$, only the trace of the $m$th power of the $Z$ covariance matrix needs to be computed — there is no need to compute the eigenvalues of $Z$! In order to use Wood's $F$ method, only the first three cumulants are needed, and hence only $\text{tr}(Z^m)$ for $m = 1, 2, 3$ need to be computed. In fact, the cumulants are a scalar multiple of these terms, where the scalar equals $(m-1)!2^{m-1}$. Before we begin, note that the matrix covariance $Z$ can be expressed as:

$$Z_{ik} = \begin{cases} x_{ik} & \text{if } i \neq k \\ x_{ii} + \alpha_i & \text{if } i = k \end{cases} \qquad \text{(A.69)}$$

where

$$x_{ik} = \frac{1}{v_{N,\lambda}}\sqrt{\lambda^{N-i}}\sqrt{\lambda^{N-k}}\left(u_{N,\lambda} - \frac{1}{w_{N,\lambda}}(\lambda^{N-i} + \lambda^{N-k})\right)\sigma^2 \qquad \text{(A.70)}$$

$$\alpha_i = \frac{1}{v_{N,\lambda}}\lambda^{N-i}\sigma^2 \qquad \text{(A.71)}$$

and so

$$x_{ii} = \frac{1}{v_{N,\lambda}} \lambda^{N-i} \left( u_{N,\lambda} - \frac{2}{w_{N,\lambda}} \lambda^{N-i} \right) \sigma^2. \tag{A.72}$$

Note that $Z_{ik} = Z_{ki}$ and $x_{ik} = x_{ki}$ (symmetry). This will be very useful during the calculations below. Furthermore, for the remainder of this section, we shall simplify notation somewhat by dropping certain subscripts, i.e.

$$\begin{aligned} u &= u_{N,\lambda} \\ v &= v_{N,\lambda} \\ w &= w_{N,\lambda} \end{aligned}$$

However, if the subscripts are different, of course this will be written in full, e.g. $w_{N,\lambda^2}$. With these preliminaries dispensed with, the computation of the first three cumulants can now begin.

## First cumulant

$$
\begin{aligned}
\mathrm{tr}(Z) &= \sum_{i=1}^{N} Z_{ii} \\
&= \sum_{i=1}^{N} (x_{ii} + \alpha_i) \\
&= \sum_{i=1}^{N} \left( \frac{1}{v}\lambda^{N-i} \left( u - \frac{2}{w}\lambda^{N-i} \right) \sigma^2 + \frac{1}{v}\lambda^{N-i}\sigma^2 \right) \\
&= \sigma^2 \sum_{i=1}^{N} \frac{\lambda^{N-i}}{v} \left( 1 + u - \frac{2}{w}\lambda^{N-i} \right) \\
&= \frac{\sigma^2}{v} \left( (1+u) \sum_{i=1}^{N} \lambda^{N-i} - \frac{2}{w} \sum_{i=1}^{N} \left(\lambda^{N-i}\right)^2 \right) \\
&= \frac{\sigma^2}{v} \left( (1+u) \sum_{i=1}^{N} \lambda^{N-i} - 2w\frac{1}{w^2} \sum_{i=1}^{N} \left(\lambda^{N-i}\right)^2 \right) \\
&= \frac{\sigma^2}{v} \left( (1+u)\, w - 2wu \right) \\
&= \frac{\sigma^2}{v} \left(1-u\right) w \\
&= \sigma^2
\end{aligned}
$$

since $v = w(1-u)$.

## Second cumulant

$$
\begin{aligned}
(Z^2)_{ij} &= (ZZ)_{ij} \\
&= \sum_{k=1}^{N} Z_{ik} Z_{kj} \\
\Rightarrow \text{tr}(Z^2) &= \sum_{i=1}^{N} \sum_{k=1}^{N} Z_{ik} Z_{ki} \\
&= \sum_{k=1}^{N} \sum_{i=1}^{N} Z_{ik}^2 \\
&= \sum_{\substack{k=1\\k\neq i}}^{N} \sum_{i=1}^{N} Z_{ik}^2 + \sum_{k=i} \sum_{i=1}^{N} Z_{ik}^2 \\
&= \sum_{\substack{k=1\\k\neq i}}^{N} \sum_{i=1}^{N} Z_{ik}^2 + \sum_{i=1}^{N} Z_{ii}^2 \\
&= \sum_{\substack{k=1\\k\neq i}}^{N} \sum_{i=1}^{N} x_{ik}^2 + \sum_{i=1}^{N} (x_{ii} + \alpha_i)^2 \\
&= \sum_{\substack{k=1\\k\neq i}}^{N} \sum_{i=1}^{N} x_{ik}^2 + \sum_{i=1}^{N} (x_{ii}^2 + 2x_{ii}\alpha_i + \alpha_i^2) \\
&= \sum_{k=1}^{N} \sum_{i=1}^{N} x_{ik}^2 + \sum_{i=1}^{N} (2x_{ii}\alpha_i + \alpha_i^2) \\
&= \sum_{i=1}^{N} \sum_{k=1}^{N} x_{ik}^2 + 2\sum_{i=1}^{N} x_{ii}\alpha_i + \sum_{i=1}^{N} \alpha_i^2 \\
&= A + B + C,
\end{aligned}
$$

where

$$A = \sum_{i=1}^{N}\sum_{k=1}^{N} x_{ik}^2$$
$$B = 2\sum_{i=1}^{N} x_{ii}\alpha_i$$
$$C = \sum_{i=1}^{N} \alpha_i^2$$

Each of these terms will be tackled separately (indeed, breaking down the computation will be even more necessary for the third cumulant).

$$\begin{aligned}
A &= \sum_{i=1}^{N}\sum_{k=1}^{N}(x_{ik})^2 \\
&= \sum_{i=1}^{N}\sum_{k=1}^{N}\left(\frac{1}{v}\sqrt{\lambda^{N-i}}\sqrt{\lambda^{N-k}}\left(u-\frac{1}{w}(\lambda^{N-i}+\lambda^{N-k})\right)\sigma^2\right)^2 \\
&= \frac{\sigma^4}{v^2}\sum_{i=1}^{N}\sum_{k=1}^{N}\lambda^{N-i}\lambda^{N-k}\left(u^2-\frac{2u}{w}(\lambda^{N-i}+\lambda^{N-k})+\frac{1}{w^2}\left(\lambda^{N-i}+\lambda^{N-k}\right)^2\right) \\
&= \frac{\sigma^4}{v^2}\sum_{i=1}^{N}\sum_{k=1}^{N}\left[\lambda^{N-i}\lambda^{N-k}u^2-\frac{2u}{w}\lambda^{N-k}(\lambda^2)^{N-i}-\frac{2u}{w}(\lambda^2)^{N-k}\lambda^{N-i}\right. \\
&\qquad \left.+\frac{1}{w^2}(\lambda^3)^{N-i}\lambda^{N-k}+\frac{2}{w^2}(\lambda^2)^{N-k}(\lambda^2)^{N-i}+\frac{1}{w^2}\lambda^{N-i}(\lambda^3)^{N-k}\right] \\
&= \frac{\sigma^4}{v^2}\left[A_1+A_2+A_3+A_4+A_5+A_6\right]
\end{aligned}$$

where

$$\begin{aligned}
A_1 &= \sum_{i=1}^{N}\sum_{k=1}^{N} \lambda^{N-i}\lambda^{N-k}u^2 \\
A_2 &= -\sum_{i=1}^{N}\sum_{k=1}^{N} \frac{2u}{w}\lambda^{N-k}(\lambda^2)^{N-i} \\
A_3 &= -\sum_{i=1}^{N}\sum_{k=1}^{N} \frac{2u}{w}(\lambda^2)^{N-k}\lambda^{N-i} \\
A_4 &= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{1}{w^2}(\lambda^3)^{N-i}\lambda^{N-k} \\
A_5 &= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{2}{w^2}(\lambda^2)^{N-k}(\lambda^2)^{N-i} \\
A_6 &= \sum_{i=1}^{N}\sum_{k=1}^{N} \frac{1}{w^2}\lambda^{N-i}(\lambda^3)^{N-k}
\end{aligned}$$

Each of these terms can be easily computed:

$$\begin{aligned}
A_1 &= u^2w^2 \\
A_2 &= -\frac{2u}{w}\cdot w\cdot(w_{N,\lambda^2}) = -2u\cdot(w_{N,\lambda^2}) = -2u^2w^2 \\
A_3 &= A_2 \text{ (symmetry) } = -2u^2w^2 \\
A_4 &= \frac{1}{w^2}\cdot(w_{N,\lambda^3})\cdot w = \frac{1}{w}(w_{N,\lambda^3}) \\
A_5 &= \frac{2}{w^2}\cdot(w_{N,\lambda^2})\cdot(w_{N,\lambda^2}) = 2u(w_{N,\lambda^2}) = 2u^2w^2 \\
A_6 &= A_4 \text{ (symmetry) } = \frac{1}{w}(w_{N,\lambda^3})
\end{aligned}$$

where the relation

$$u = \frac{1}{w^2}(w_{N,\lambda^2}) \Leftrightarrow uw^2 = (w_{N,\lambda^2}) \tag{A.73}$$

has been used repeatedly. Therefore,

$$
\begin{aligned}
A &= \frac{\sigma^4}{v^2}\left[u^2w^2 - 2u^2w^2 - 2u^2w^2 + \frac{1}{w}(w_{N,\lambda^3}) + 2u^2w^2 + \frac{1}{w}(w_{N,\lambda^3})\right] \\
&= \frac{\sigma^4}{v^2}\left[-u^2w^2 + \frac{2}{w}(w_{N,\lambda^3})\right]
\end{aligned}
$$

Term $B$ is easier to compute:

$$
\begin{aligned}
B &= 2\sum_{i=1}^{N} x_{ii}\alpha_i \\
&= 2\sum_{i=1}^{N} \frac{1}{v}\lambda^{N-i}\left(u - \frac{2}{w}\lambda^{N-i}\right)\sigma^2\frac{1}{v}\lambda^{N-i}\sigma^2 \\
&= \frac{2\sigma^4}{v^2}\sum_{i=1}^{N}(\lambda^2)^{N-i}\left(u - \frac{2}{w}\lambda^{N-i}\right) \\
&= \frac{2\sigma^4}{v^2}\left[\sum_{i=1}^{N}(\lambda^2)^{N-i}u - \frac{2}{w}\sum_{i=1}^{N}(\lambda^3)^{N-i}\right] \\
&= \frac{2\sigma^4}{v^2}\left[u(w_{N,\lambda^2}) - \frac{2}{w}(w_{N,\lambda^3})\right] \\
&= \frac{\sigma^4}{v^2}\left[2u^2w^2 - \frac{4}{w}(w_{N,\lambda^3})\right]
\end{aligned}
$$

And term $C$ equals,

$$
\begin{aligned}
C &= \sum_{i=1}^{N}\alpha_i^2 \\
&= \sum_{i=1}^{N}\frac{1}{v^2}(\lambda^2)^{N-i}\sigma^4 \\
&= \frac{\sigma^4}{v^2}(w_{N,\lambda^2}) \\
&= \frac{\sigma^4}{v^2}uw^2
\end{aligned}
$$

Putting these three terms together,

$$\begin{aligned}\mathrm{tr}(U^2) &= A + B + C \\ &= \frac{\sigma^4}{v^2}\left[\left(-u^2w^2 + \frac{2}{w}(w_{N,\lambda^3})\right) + \left(2u^2w^2 - \frac{4}{w}(w_{N,\lambda^3})\right) + (uw^2)\right] \\ &= \frac{\sigma^4}{v^2}\left[u^2w^2 - \frac{2}{w}(w_{N,\lambda^3}) + uw^2\right]\end{aligned}$$

which completes the computation for the second cumulant.

## A short note on double summation notation

If we consider the term $X$,

$$X = \sum_{i=1}^{N}\sum_{\substack{k=1\\k\neq i}}^{N} x_{ik}, \tag{A.74}$$

in order to compute this term, the summation over $i$ would need to be done first, and then the summation over $k$ (when $k \neq i$) could then be done. Technically, therefore, this term should be written with a different order of summation

$$X = \sum_{\substack{k=1\\k\neq i}}^{N}\sum_{i=1}^{N} x_{ik}.$$

However, in the next section it will be more convenient to write terms such as $X$ using the form in Equation (A.74), even though it is understood that to compute the term the summation over $i$ will need to be done first.

## Third cumulant

$$\begin{aligned}
(Z^3)_{pq} &= ((ZZ)Z)_{pq} \\
&= \sum_{t=1}^{N} (ZZ)_{pt} Z_{tq} \\
&= \sum_{t=1}^{N} \left( \sum_{k=1}^{N} Z_{pk} Z_{kt} \right) Z_{tq} \\
&= \sum_{t=1}^{N} \sum_{t=1}^{N} Z_{pk} Z_{kt} Z_{tq} \\
\Rightarrow \text{tr}(Z^3) &= \sum_{q=1}^{N} \sum_{t=1}^{N} \sum_{k=1}^{N} Z_{qk} Z_{kt} Z_{tq}
\end{aligned}$$

First, this sum will need to be rewritten in terms of $x_{qk}$ and $\alpha_q$ terms.

$$\begin{aligned}
\text{tr}(Z^3) &= \sum_{q=1}^{N} \sum_{t=1}^{N} \sum_{k=1}^{N} Z_{qk} Z_{kt} Z_{tq} \\
&= \sum_{q=1}^{N} \sum_{t=1}^{N} \left[ \sum_{\substack{k=1 \\ k \neq t \\ k \neq q}}^{N} Z_{qk} Z_{kt} Z_{tq} + \sum_{k=t} Z_{qk} Z_{kt} Z_{tq} + \sum_{k=q} Z_{qk} Z_{kt} Z_{tq} \right] \\
&= \sum_{q=1}^{N} \sum_{t=1}^{N} \left[ \sum_{\substack{k=1 \\ k \neq t \\ k \neq q}}^{N} Z_{qk} Z_{kt} Z_{tq} + Z_{qt} Z_{tt} Z_{tq} + Z_{qq} Z_{qt} Z_{tq} \right] \\
&= A + B + C
\end{aligned}$$

where

$$A = \sum_{q=1}^{N} \sum_{t=1}^{N} \sum_{\substack{k=1 \\ k \neq t \\ k \neq q}}^{N} Z_{qk} Z_{kt} Z_{tq}$$

$$B = \sum_{q=1}^{N} \sum_{t=1}^{N} Z_{qt} Z_{tt} Z_{tq}$$

$$C = \sum_{q=1}^{N} \sum_{t=1}^{N} Z_{qq} Z_{qt} Z_{tq}$$

Because $Z$ is symmetric, by swapping $q$ and $t$ we can see that $B = C$. Before we compute $A$, let us first expand the term $T$:

$$\begin{aligned}
T &= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N} x_{qk}x_{kt}x_{tq} \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\left[\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{k=t} x_{qk}x_{kt}x_{tq} + \sum_{k=q} x_{qk}x_{kt}x_{tq}\right] \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\left[\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + x_{qt}x_{tt}x_{tq} + x_{qq}x_{qt}x_{tq}\right] \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{q=1}^{N}\sum_{t=1}^{N} x_{qt}x_{tt}x_{tq} + \sum_{q=1}^{N}\sum_{t=1}^{N} x_{qq}x_{qt}x_{tq} \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + 2\sum_{q=1}^{N}\sum_{t=1}^{N} x_{tt}x_{qt}x_{tq} \quad \text{(by symmetry)} \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + 2\sum_{q=1}^{N}\sum_{t=1}^{N} x_{tt}(x_{qt})^2 \\
&= \left(\sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{q=1}^{N}\sum_{t=q}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq}\right) + 2\sum_{q=1}^{N}\sum_{t=1}^{N} x_{tt}(x_{qt})^2 \\
\Rightarrow T &= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} x_{qq}(x_{qk})^2 + 2\sum_{q=1}^{N}\sum_{t=1}^{N} x_{tt}(x_{qt})^2 \qquad \text{(A.75)}
\end{aligned}$$

Now, returning to term $A$,

$$\begin{aligned}
A &= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} Z_{qk}Z_{kt}Z_{tq} \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} Z_{qk}Z_{kt}Z_{tq} + \sum_{q=1}^{N}\sum_{t=q}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} Z_{qk}Z_{kt}Z_{tq} \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} Z_{qk}Z_{kt}Z_{tq} + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} Z_{qk}Z_{kq}Z_{qq} \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} Z_{qk}Z_{kt}Z_{tq} + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} Z_{qq}(Z_{qk})^2 \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} (x_{qq}+\alpha_q)(x_{qk})^2 \\
\Rightarrow A &= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N}\sum_{\substack{k=1\\k\neq t\\k\neq q}}^{N} x_{qk}x_{kt}x_{tq} + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} x_{qq}(x_{qk})^2 + \sum_{q=1}^{N}\sum_{\substack{k=1\\k\neq q}}^{N} \alpha_q(x_{qk})^2
\end{aligned}$$

Note that the first two terms of $A$ coincide with the first two terms of $T$. Now, term $C$ equals

$$
\begin{aligned}
C &= \sum_{q=1}^{N}\sum_{t=1}^{N} Z_{qq}Z_{qt}Z_{tq} \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N} Z_{qq}(Z_{tq})^2 \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} Z_{qq}(Z_{tq})^2 + \sum_{q=1}^{N}\sum_{t=q} Z_{qq}(Z_{tq})^2 \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} Z_{qq}(Z_{tq})^2 + \sum_{q=1}^{N} Z_{qq}(Z_{qq})^2 \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} (x_{qq}+\alpha_q)(x_{tq})^2 + \sum_{q=1}^{N}(x_{qq}+\alpha_q)^3 \\
&= \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} x_{qq}(x_{tq})^2 + \sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} \alpha_q(x_{tq})^2 \\
&+ \sum_{q=1}^{N}(x_{qq})^3 + 3\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 3\sum_{q=1}^{N} x_{qq}(\alpha_q)^2 + \sum_{q=1}^{N}(\alpha_q)^3 \\
&= \left(\sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} x_{qq}(x_{tq})^2 + \sum_{q=1}^{N}(x_{qq})^3\right) + \left(\sum_{q=1}^{N}\sum_{\substack{t=1\\t\neq q}}^{N} \alpha_q(x_{tq})^2 + \sum_{q=1}^{N}(x_{qq})^2\alpha_q\right) \\
&\quad + 2\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 3\sum_{q=1}^{N} x_{qq}(\alpha_q)^2 + \sum_{q=1}^{N}(\alpha_q)^3 \\
&= \left(\sum_{q=1}^{N}\sum_{t=1}^{N} x_{qq}(x_{tq})^2\right) + \left(\sum_{q=1}^{N}\sum_{t=1}^{N} \alpha_q(x_{tq})^2\right) \\
&\quad + 2\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 3\sum_{q=1}^{N} x_{qq}(\alpha_q)^2 + \sum_{q=1}^{N}(\alpha_q)^3
\end{aligned}
$$

and so since $B = C$, and

$$\begin{aligned} B + C &= 2C \\ &= 2\sum_{q=1}^{N}\sum_{t=1}^{N} x_{qq}(x_{tq})^2 + 2\sum_{q=1}^{N}\sum_{t=1}^{N} \alpha_q (x_{tq})^2 \\ &\quad + 4\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 6\sum_{q=1}^{N} x_{qq}(\alpha_q)^2 + 2\sum_{q=1}^{N}(\alpha_q)^3 \end{aligned}$$

Note that the first term of $B + C$ is the third term of $T$ (after swapping $t$ andf $q$), and so

$$A + B + C = \left( \sum_{q=1}^{N} \sum_{\substack{t=1 \\ t \neq q}}^{N} \sum_{\substack{k=1 \\ k \neq t \\ k \neq q}}^{N} x_{qk} x_{kt} x_{tq} + \sum_{q=1}^{N} \sum_{\substack{k=1 \\ k \neq q}}^{N} x_{qq} (x_{qk})^2 \right) + \sum_{q=1}^{N} \sum_{\substack{k=1 \\ k \neq q}}^{N} \alpha_q (x_{qk})^2$$
$$+ \left( 2 \sum_{q=1}^{N} \sum_{t=1}^{N} x_{qq} (x_{tq})^2 \right) + 2 \sum_{q=1}^{N} \sum_{t=1}^{N} \alpha_q (x_{tq})^2$$
$$+ 4 \sum_{q=1}^{N} (x_{qq})^2 \alpha_q + 6 \sum_{q=1}^{N} x_{qq} (\alpha_q)^2 + 2 \sum_{q=1}^{N} (\alpha_q)^3$$

$$= (T) + \sum_{q=1}^{N} \sum_{\substack{k=1 \\ k \neq q}}^{N} \alpha_q (x_{qk})^2 + 2 \sum_{q=1}^{N} \sum_{t=1}^{N} \alpha_q (x_{tq})^2$$
$$+ 4 \sum_{q=1}^{N} (x_{qq})^2 \alpha_q + 6 \sum_{q=1}^{N} x_{qq} (\alpha_q)^2 + 2 \sum_{q=1}^{N} (\alpha_q)^3$$

$$= T + \left( \sum_{q=1}^{N} \sum_{\substack{k=1 \\ k \neq q}}^{N} \alpha_q (x_{qk})^2 + \sum_{q=1}^{N} (x_{qq})^2 \alpha_q \right) + 2 \sum_{q=1}^{N} \sum_{t=1}^{N} \alpha_q (x_{tq})^2$$
$$+ 3 \sum_{q=1}^{N} (x_{qq})^2 \alpha_q + 6 \sum_{q=1}^{N} x_{qq} (\alpha_q)^2 + 2 \sum_{q=1}^{N} (\alpha_q)^3$$

$$= T + \left( \sum_{q=1}^{N} \sum_{k=1}^{N} \alpha_q (x_{qk})^2 \right) + 2 \sum_{q=1}^{N} \sum_{t=1}^{N} \alpha_q (x_{tq})^2$$
$$+ 3 \sum_{q=1}^{N} (x_{qq})^2 \alpha_q + 6 \sum_{q=1}^{N} x_{qq} (\alpha_q)^2 + 2 \sum_{q=1}^{N} (\alpha_q)^3$$

which gives us

$$\begin{aligned}\text{tr}(U^3) &= A + B + C \\ &= T + \sum_{q=1}^{N}\sum_{k=1}^{N}\alpha_q(x_{qk})^2 + 2\sum_{q=1}^{N}\sum_{t=1}^{N}\alpha_q(x_{tq})^2 \\ &+ 3\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 6\sum_{q=1}^{N}x_{qq}(\alpha_q)^2 + 2\sum_{q=1}^{N}(\alpha_q)^3\end{aligned}$$

$$= T + 3\Big(\sum_{q=1}^{N}\sum_{k=1}^{N}\alpha_q(x_{qk})^2\Big) + 3\sum_{q=1}^{N}(x_{qq})^2\alpha_q + 6\sum_{q=1}^{N}x_{qq}(\alpha_q)^2 + 2\sum_{q=1}^{N}(\alpha_q)^3$$

by relabelling the index $t$ as $k$ in the third term, and so finally

$$\text{tr}(U^3) = E_1 + E_2 + E_3 + E_4 + E_5 \tag{A.76}$$

where

$$E_1 = T = \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}x_{qk}x_{kt}x_{tq} \tag{A.77}$$

$$E_2 = 3\sum_{q=1}^{N}\sum_{k=1}^{N}\alpha_q(x_{qk})^2 \tag{A.78}$$

$$E_3 = 3\sum_{q=1}^{N}\alpha_q(x_{qq})^2 \tag{A.79}$$

$$E_4 = 6\sum_{q=1}^{N}x_{qq}(\alpha_q)^2 \tag{A.80}$$

$$E_5 = 2\sum_{q=1}^{N}(\alpha_q)^3 \tag{A.81}$$

This whole section was spent performing necessary manipulation in order to obtain computable summations (in terms of $x_{ij}$ and $\alpha_i$). The computations of $E_1$ to $E_5$ will be done in the next sections.

## Computation of $E_1$

From Equation (A.77),

$$
\begin{aligned}
E_1 = T &= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N} x_{qk}x_{kt}x_{tq} \\
&= \sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N} \frac{\sigma^6}{v^3}\sqrt{\lambda^{N-q}}\sqrt{\lambda^{N-k}}\sqrt{\lambda^{N-k}}\sqrt{\lambda^{N-t}}\sqrt{\lambda^{N-t}}\sqrt{\lambda^{N-q}}. \\
&\quad \cdot \left(u - \frac{1}{w}(\lambda^{N-q}+\lambda^{N-k})\right)\left(u - \frac{1}{w}(\lambda^{N-k}+\lambda^{N-t})\right)\left(u - \frac{1}{w}(\lambda^{N-t}+\lambda^{N-q})\right) \\
&= \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N} \lambda^{N-q}\lambda^{N-k}\lambda^{N-t}\cdot\left(u - \frac{1}{w}\left(\lambda^{N-q}+\lambda^{N-k}\right)\right)\cdot \\
&\quad \cdot \left(u - \frac{1}{w}\left(\lambda^{N-k}+\lambda^{N-t}\right)\right)\left(u - \frac{1}{w}\left(\lambda^{N-t}+\lambda^{N-q}\right)\right)
\end{aligned}
$$

Now, we first work out that:

$$\begin{aligned}
X &= \left(u - \frac{1}{w}(\alpha+\beta)\right)\left(u - \frac{1}{w}(\beta+\gamma)\right)\left(u - \frac{1}{w}(\gamma+\alpha)\right) \\
&= \left(u^2 - \frac{u}{w}(\alpha+2\beta+\gamma) + \frac{1}{w^2}(\alpha\beta+\beta^2+\alpha\gamma+\beta\gamma)\right)\left(u - \frac{1}{w}(\gamma+\alpha)\right) \\
&= u^3 - \frac{u^2}{w}(\alpha+2\beta+\gamma) + \frac{u}{w^2}(\alpha\beta+\beta^2+\alpha\gamma+\beta\gamma) - \frac{u^2}{w}(\gamma+\alpha) \\
&+ \frac{u}{w^2}(\alpha+2\beta+\gamma)(\gamma+\alpha) - \frac{1}{w^3}(\alpha\beta+\beta^2+\alpha\gamma+\beta\gamma)(\gamma+\alpha) \\
\\
&= u^3 - \frac{u^2}{w}(2\alpha+2\beta+2\gamma) + \frac{u}{w^2}(\alpha\beta+\beta^2+\alpha\gamma+\beta\gamma) \\
&+ \frac{u}{w^2}(\alpha\gamma+2\beta\gamma+\gamma^2+\alpha^2+2\alpha\beta+\alpha\gamma) \\
&- \frac{1}{w^3}(\alpha\beta\gamma+\beta^2\gamma+\alpha\gamma^2+\beta\gamma^2+\alpha^2\beta+\alpha\beta^2+\alpha^2\gamma+\alpha\beta\gamma) \\
\\
&= u^3 - \frac{2u^2}{w}(\alpha+\beta+\gamma) + \frac{u}{w^2}(3\alpha\beta+3\alpha\gamma+3\beta\gamma+\alpha^2+\beta^2+\gamma^2) \\
&- \frac{1}{w^3}(2\alpha\beta\gamma+\alpha\beta^2+\alpha\gamma^2+\beta\alpha^2+\beta\gamma^2+\gamma\alpha^2+\gamma\beta^2)
\end{aligned} \tag{A.82}$$

and if we label

$$\alpha = \lambda^{N-q}, \qquad \beta = \lambda^{N-k}, \qquad \gamma = \lambda^{N-t}$$

we can use the algebra for $X$ above to rewrite $E_1$ as

$$\begin{aligned}
E_1 &= \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma\Big[u^3 - \frac{2u^2}{w}(\alpha+\beta+\gamma) \\
&+ \frac{u}{w^2}(3\alpha\beta+3\alpha\gamma+3\beta\gamma+\alpha^2+\beta^2+\gamma^2) \\
&- \frac{1}{w^3}(2\alpha\beta\gamma+\alpha\beta^2+\alpha\gamma^2+\beta\alpha^2+\beta\gamma^2+\gamma\alpha^2+\gamma\beta^2)\Big] \\
&= E_{11} + E_{12} + E_{13} + E_{14}
\end{aligned}$$

where

$$\begin{aligned}
E_{11} &= \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma u^3 \\
&= \frac{\sigma^6 u^3}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\lambda^{N-k}\lambda^{N-q}\lambda^{N-t} \\
&= \frac{\sigma^6 u^3}{v^3} w\cdot w\cdot w \\
&= \frac{\sigma^6 u^3}{v^3} w^3
\end{aligned}$$

and

$$\begin{aligned}
E_{12} &= \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma\Big[-\frac{2u^2}{w}(\alpha+\beta+\gamma)\Big] \\
&= -\frac{2\sigma^6 u^2}{wv^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\Big[\alpha^2\beta\gamma+\alpha\beta^2\gamma+\alpha\beta\gamma^2\Big] \\
&= -\frac{2\sigma^6 u^2}{wv^3}\big[(w_{N,\lambda^2})w^2 + w(w_{N,\lambda^2})w + w^2(w_{N,\lambda^2})\big] \\
&= -\frac{6\sigma^6 u^2}{wv^3}(w_{N,\lambda^2})w^2 \\
&= -\frac{6\sigma^6 u^2 w^3}{v^3}\frac{1}{w^2}(w_{N,\lambda^2}) \\
&= -\frac{6\sigma^6 u^3 w^3}{v^3}
\end{aligned}$$

Now, since

$$\begin{aligned}
\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}3\alpha^2\beta^2\gamma &= 3(w_{N,\lambda^2})(w_{N,\lambda^2})w \\
\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha^3\beta\gamma &= (w_{N,\lambda^3})w^2
\end{aligned}$$

using symmetry we can compute

$$\begin{aligned}
E_{13} &= \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma\left[\frac{u}{w^2}(3\alpha\beta+3\alpha\gamma+3\beta\gamma+\alpha^2+\beta^2+\gamma^2)\right] \\
&= \frac{\sigma^6 u}{w^2 v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\left[3\alpha^2\beta^2\gamma+3\alpha^2\beta\gamma^2+3\alpha\beta^2\gamma^2+\alpha^3\beta\gamma+\alpha\beta^3\gamma+\alpha\beta\gamma^3\right] \\
&= \frac{\sigma^6 u}{w^2 v^3}\left[3\cdot 3(w_{N,\lambda^2})(w_{N,\lambda^2})w+3\cdot(w_{N,\lambda^3})w^2\right] \\
&= \sigma^6\left(\frac{9u}{v^3 w}(w_{N,\lambda^2})^2+\frac{3u}{v^3}(w_{N,\lambda^3})\right) \\
&= \sigma^6\left(\frac{9uw^3}{v^3}\frac{1}{w^4}(w_{N,\lambda^2})^2+\frac{3u}{v^3}(w_{N,\lambda^3})\right) \\
&= \sigma^6\left(\frac{9u^3w^3}{v^3}+\frac{3u}{v^3}(w_{N,\lambda^3})\right)
\end{aligned}$$

and finally

$$E_{14} = \frac{\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma\left[-\frac{1}{w^3}(2\alpha\beta\gamma+\alpha\beta^2+\alpha\gamma^2+\beta\alpha^2+\beta\gamma^2+\gamma\alpha^2+\gamma\beta^2)\right] \tag{A.83}$$

we break this up further

$$\begin{aligned}
E_{141} &= -\frac{\sigma^6}{v^3}\frac{1}{w^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma(2\alpha\beta\gamma) \\
&= -\frac{2\sigma^6}{w^3 v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha^2\beta^2\gamma^2 \\
&= -\frac{2\sigma^6}{w^3 v^3}(w_{N,\lambda^2})^3 \\
&= -\frac{2\sigma^6 w^3}{v^3}\frac{1}{w^6}(w_{N,\lambda^2})^3 \\
&= -\frac{2\sigma^6 u^3 w^3}{v^3}
\end{aligned}$$

and

$$\begin{aligned}
E_{142} &= -\frac{\sigma^6}{w^3 v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha\beta\gamma(\alpha\beta^2) \\
&= -\frac{\sigma^6}{w^3 v^3}\sum_{q=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{N}\alpha^2\beta^3\gamma \\
&= -\frac{\sigma^6}{w^3 v^3}(w_{N,\lambda^2})(w_{N,\lambda^3})w \\
&= -\frac{\sigma^6}{v^3}\frac{1}{w^2}(w_{N,\lambda^2})(w_{N,\lambda^3}) \\
&= -\frac{\sigma^6 u}{v^3}(w_{N,\lambda^3})
\end{aligned} \tag{A.84}$$

so using symmetry

$$\begin{aligned}
E_{14} &= E_{141} + 6E_{142} \\
&= \sigma^6\left(-\frac{2u^3w^3}{v^3} - \frac{6u}{v^3}(w_{N,\lambda^3})\right)
\end{aligned}$$

Putting these four terms together,

$$\begin{aligned}
\frac{1}{\sigma^6}E_1 &= E_{11} + E_{12} + E_{13} + E_{14} \\
&= \left(\frac{u^3w^3}{v^3}\right) + \left(-\frac{6u^3w^3}{v^3}\right) + \left(\frac{9u^3w^3}{v^3} + \frac{3u}{v^3}(w_{N,\lambda^3})\right) + \left(-\frac{2u^3w^3}{v^3} - \frac{6u}{v^3}(w_{N,\lambda^3})\right) \\
&= \frac{u^3w^3}{v^3}(1-6+9-2) + \left(\frac{u}{v^3}(w_{N,\lambda^3})\right)(3-6) \\
&= \frac{2u^3w^3}{v^3} - \frac{3u}{v^3}(w_{N,\lambda^3}) \\
\Rightarrow E_1 &= \sigma^6\left(\frac{2u^3w^3}{v^3} - \frac{3u}{v^3}(w_{N,\lambda^3})\right)
\end{aligned} \tag{A.85}$$

## Computation of $E_2$

Now for $E_2$, from Equation (A.78):

$$
\begin{aligned}
E_2 &= 3\sum_{q=1}^{N}\sum_{k=1}^{N}\alpha_q(x_{qk})^2 \\
&= 3\sum_{q=1}^{N}\sum_{k=1}^{N}\frac{\lambda^{N-q}}{v}\sigma^2\left(\frac{1}{v}\sqrt{\lambda^{N-q}}\sqrt{\lambda^{N-k}}\left(u-\frac{1}{w}(\lambda^{N-q}+\lambda^{N-k})\right)\sigma^2\right)^2 \\
&= \frac{3\sigma^6}{v^3}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}\Bigg(u^2-\frac{2u}{w}\left(\lambda^{N-q}+\lambda^{N-k}\right) \\
&\qquad +\frac{1}{w^2}\left((\lambda^2)^{N-q}+(\lambda^2)^{N-k}+2\lambda^{N-q}\lambda^{N-k}\right)\Bigg) \\
&= \frac{3\sigma^6}{v^3}\left(E_{21}+E_{22}+E_{23}+E_{24}+E_{25}+E_{26}\right)
\end{aligned}
$$

where

$$
\begin{aligned}
E_{21} &= \sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}u^2 \\
&= (w_{N,\lambda^2})\,wu^2 \\
&= (w_{N,\lambda^2})\,\frac{1}{w^2}w^3u^2 \\
&= u^3w^3
\end{aligned}
$$

and

$$
\begin{aligned}
E_{22} &= -\frac{2u}{w}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}\left(\lambda^{N-q}\right) \\
&= -\frac{2u}{w}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^3)^{N-q}\lambda^{N-k} \\
&= -\frac{2u}{w}(w_{N,\lambda^3})w \\
&= -2u(w_{N,\lambda^3})
\end{aligned}
$$

and

$$\begin{aligned}
E_{23} &= -\frac{2u}{w}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}\left(\lambda^{N-k}\right) \\
&= -\frac{2u}{w}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}(\lambda^2)^{N-k} \\
&= -\frac{2u}{w}(w_{N,\lambda^2})(w_{N,\lambda^2}) \\
&= -\frac{2uw^3}{w^4}(w_{N,\lambda^2})(w_{N,\lambda^2}) \\
&= -2u^3w^3
\end{aligned}$$

and

$$\begin{aligned}
E_{24} &= \sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}\frac{1}{w^2}(\lambda^2)^{N-q} \\
&= \frac{1}{w^2}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^4)^{N-q}\lambda^{N-k} \\
&= \frac{1}{w^2}(w_{N,\lambda^4})w \\
&= \frac{1}{w}(w_{N,\lambda^4})
\end{aligned}$$

and

$$\begin{aligned}
E_{25} &= \sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}\lambda^{N-k}\frac{1}{w^2}(\lambda^2)^{N-k} \\
&= \frac{1}{w^2}\sum_{q=1}^{N}\sum_{k=1}^{N}(\lambda^2)^{N-q}(\lambda^3)^{N-k} \\
&= \frac{1}{w^2}(w_{N,\lambda^2})(w_{N,\lambda^3}) \\
&= u(w_{N,\lambda^3})
\end{aligned}$$

and

$$\begin{aligned}
E_{26} &= \sum_{q=1}^{N} \sum_{k=1}^{N} (\lambda^2)^{N-q} \lambda^{N-k} \frac{2}{w^2} \lambda^{N-q} \lambda^{N-k} \\
&= \frac{2}{w^2} \sum_{q=1}^{N} \sum_{k=1}^{N} (\lambda^3)^{N-q} (\lambda^2)^{N-k} \\
&= \frac{2}{w^2} (w_{N,\lambda^3})(w_{N,\lambda^2}) \\
&= 2u(w_{N,\lambda^3})
\end{aligned}$$

and putting these six terms together

$$\begin{aligned}
E_2 &= E_{21} + E_{22} + E_{23} + E_{24} + E_{25} + E_{26} \\
&= \frac{3\sigma^6}{v^3} \left( u^3 w^3 - 2u(w_{N,\lambda^3}) - 2u^3 w^3 + \frac{1}{w}(w_{N,\lambda^4}) + u(w_{N,\lambda^3}) + 2u(w_{N,\lambda^3}) \right) \\
&= \frac{3\sigma^6}{v^3} \left( u^3 w^3 (1-2) + u(w_{N,\lambda^3})(-2+1+2) + \frac{1}{w}(w_{N,\lambda^4}) \right) \\
&= \frac{3\sigma^6}{v^3} \left( -u^3 w^3 + u(w_{N,\lambda^3}) + \frac{1}{w}(w_{N,\lambda^4}) \right) \\
&= \sigma^6 \left( \frac{-3u^3 w^3}{v^3} + \frac{3u}{v^3}(w_{N,\lambda^3}) + \frac{3}{wv^3}(w_{N,\lambda^4}) \right)
\end{aligned}$$

## Computation of $E_3$, $E_4$ and $E_5$

From Equation (A.79),

$$\begin{aligned}
E_3 &= 3\sum_{q=1}^{N}\alpha_q(x_{qq})^2 \\
&= 3\sum_{q=1}^{N}\frac{\lambda^{N-q}}{v}\sigma^2\left(\frac{\lambda^{N-q}}{v}\left(u-\frac{2}{w}\lambda^{N-q}\right)\sigma^2\right)^2 \\
&= \frac{3\sigma^6}{v^3}\sum_{q=1}^{N}(\lambda^3)^{N-q}\left(u^2-\frac{4u}{w}\lambda^{N-q}+\frac{4}{w^2}(\lambda^2)^{N-q}\right) \\
&= \frac{3\sigma^6}{v^3}\sum_{q=1}^{N}\left((\lambda^3)^{N-q}u^2-\frac{4u}{w}(\lambda^4)^{N-q}+\frac{4}{w^2}(\lambda^5)^{N-q}\right) \\
&= \frac{3\sigma^6}{v^3}\left(u^2(w_{N,\lambda^3})-\frac{4u}{w}(w_{N,\lambda^4})+\frac{4}{w^2}(w_{N,\lambda^5})\right) \\
&= \sigma^6\left(3\frac{u^2}{v^3}(w_{N,\lambda^3})-\frac{12u}{wv^3}(w_{N,\lambda^4})+\frac{12}{w^2v^3}(w_{N,\lambda^5})\right)
\end{aligned}$$

and from Equation (A.80),

$$\begin{aligned}
E_4 &= 6\sum_{q=1}^{N}\left(\frac{\lambda^{N-q}}{v}\sigma^2\right)^2\left(\frac{\lambda^{N-q}}{v}\left(u-\frac{2}{w}\lambda^{N-q}\right)\sigma^2\right) \\
&= \frac{6\sigma^6}{v^3}\sum_{q=1}^{N}(\lambda^3)^{N-q}\left(u-\frac{2}{w}\lambda^{N-q}\right) \\
&= \frac{6\sigma^6}{v^3}\sum_{q=1}^{N}\left((\lambda^3)^{N-q}u-\frac{2}{w}(\lambda^4)^{N-q}\right) \\
&= \frac{6\sigma^6}{v^3}\left((w_{N,\lambda^3})u-\frac{2}{w}(w_{N,\lambda^4})\right) \\
&= \sigma^6\left(\frac{6u}{v^3}(w_{N,\lambda^3})-\frac{12}{wv^3}(w_{N,\lambda^4})\right)
\end{aligned}$$

and from Equation (A.81),

$$\begin{aligned}
E_5 &= 2\sum_{q=1}^{N}(\alpha_q)^3 \\
&= 2\sum_{q=1}^{N}\left(\frac{\lambda^{N-q}}{v}\sigma^2\right)^3 \\
&= \frac{2\sigma^6}{v^3}\sum_{q=1}^{N}\left(\lambda^{N-q}\right)^3 \\
&= \frac{2\sigma^6}{v^3}(w_{N,\lambda^3})
\end{aligned}$$

## Putting terms together

After all these computations, we have

$$\begin{aligned}
E_1 &= \sigma^6\left(\frac{2u^3w^3}{v^3} - \frac{3u}{v^3}(w_{N,\lambda^3})\right) \\
E_2 &= \sigma^6\left(\frac{-3u^3w^3}{v^3} + \frac{3u}{v^3}(w_{N,\lambda^3}) + \frac{3}{wv^3}(w_{N,\lambda^4})\right) \\
E_3 &= \sigma^6\left(3\frac{u^2}{v^3}(w_{N,\lambda^3}) - \frac{12u}{wv^3}(w_{N,\lambda^4}) + \frac{12}{w^2v^3}(w_{N,\lambda^5})\right) \\
E_4 &= \sigma^6\left(\frac{6u}{v^3}(w_{N,\lambda^3}) - \frac{12}{wv^3}(w_{N,\lambda^4})\right) \\
E_5 &= \sigma^6\left(\frac{2}{v^3}(w_{N,\lambda^3})\right)
\end{aligned}$$

and putting these all together, we finally get

$$\begin{aligned}
\text{tr}(Z^3) = \frac{\sigma^6}{v^3}\Big[&\left(u^3w^3\right)(2-3) + (w_{N,\lambda^3})(-3u+3u+3u^2+6u+2) \\
&+ \frac{1}{w}(w_{N,\lambda^4})(3-12u-12) + \frac{12}{w^2}(w_{N,\lambda^5})\Big] \\
= \frac{\sigma^6}{v^3}\Big[&-\left(u^3w^3\right) + (w_{N,\lambda^3})(3u^2+6u+2) + \frac{1}{w}(w_{N,\lambda^4})(-9-12u) + \frac{12}{w^2}(w_{N,\lambda^5})\Big]
\end{aligned}$$

## Summary

Since the $r$the cumulant of a chi-squared random variable is $\kappa_r(W) = 2^{r-1}(r-1)!$ [18], the $r$th cumulant of $s^2_{N,\lambda}$ is

$$\kappa_r(s^2_{N,\lambda}) = 2^{r-1}(r-1)!\text{tr}(U^r),$$

(see [24]) and so

$$\begin{aligned}
\kappa_1(s^2_{N,\lambda}) &= \sigma^2 \\
\kappa_2(s^2_{N,\lambda}) &= \frac{2\sigma^4}{v^2}\left[u^2w^2 - \frac{2}{w}(w_{N,\lambda^3}) + uw^2\right] \\
\kappa_3(s^2_{N,\lambda}) &= \frac{8\sigma^6}{v^3}\left[-\left(u^3w^3\right) + (w_{N,\lambda^3})(3u^2+6u+2) + \frac{1}{w}(w_{N,\lambda^4})(-9-12u) + \frac{12}{w^2}(w_{N,\lambda^5})\right].
\end{aligned}$$

With these three cumulants, Wood's $F$ method can be used to compute the cdf of $s^2_{N,\lambda}$ to a good accuracy.

The cumulants can be computed by sequentially updating $u, v, w, w_{N,\lambda^3}, w_{N,\lambda^4}, w_{N,\lambda^5}$.

### A.6.6 Adaptive forgetting factor variance

Suppose we have the adaptive forgetting factor

$$\overrightarrow{B} = (B_1, B_2, \ldots, B_3), \tag{A.86}$$

for the mean

$$\bar{x}_{N,\overrightarrow{B}} = \frac{1}{w_{N,\overrightarrow{B}}}\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1} B_p\right)x_k, \tag{A.87}$$

And we define

$$s^2_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{1}{V_{N,\overrightarrow{\lambda},\overrightarrow{B}}} S_{N,\overrightarrow{\lambda},\overrightarrow{B}},$$

where

$$S_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\left[x_k - \bar{x}_{N,\overrightarrow{B}}\right]^2.$$

First, we determine $V_{N,\vec{\lambda},\vec{B}}$ so that

$$\mathrm{E}[s^2_{N,\vec{\lambda},\vec{B}}] = \sigma^2. \tag{A.88}$$

## Derivation of $V_{N,\vec{\lambda},\vec{B}}$

In order to compute $E[S_{N,\vec{\lambda},\vec{B}}]$, we first expand $S_{N,\vec{\lambda},\vec{B}}$:

$$\begin{aligned}
S_{N,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\left[x_k - \bar{x}_{N,\vec{B}}\right]^2 \\
&= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\left[x_k^2 - 2x_k\bar{x}_{N,\vec{B}} + \bar{x}^2_{N,\vec{B}}\right] \\
&= \sum_{k=1}^{N}\left[\left(\prod_{p=k}^{N-1}\lambda_p\right)x_k^2 - 2\bar{x}_{N,\vec{B}}\left(\prod_{p=k}^{N-1}\lambda_p\right)x_k + \bar{x}^2_{N,\vec{B}}\left(\prod_{p=k}^{N-1}\lambda_p\right)\right] \\
\Rightarrow S &= \sum_{k=1}^{N}\left[\left(\prod_{p=k}^{N-1}\lambda_p\right)x_k^2 - 2\bar{x}_{N,\vec{B}}m_{N,\vec{\lambda}} + \bar{x}^2_{N,\vec{B}}w_{N,\vec{\lambda}}\right].
\end{aligned} \tag{A.89}$$

Computing the expectation of the middle term takes some care. Recalling

$$\text{Var}[x] = \mathrm{E}[x^2] - (\mathrm{E}[x])^2,$$

and writing

$$\beta_i = \prod_{p=i}^{N-1} B_p, \qquad \alpha_j = \prod_{q=j}^{N-1}\lambda_q,$$

and

$$P_{N,\vec{\lambda},\vec{B}} = \sum_{i=1}^{N}\beta_i\alpha_i = \sum_{i=1}^{N}\left(\prod_{p=i}^{N-1}B_p\right)\left(\prod_{q=i}^{N-1}\lambda_q\right). \tag{A.90}$$

We then assume

$$\mathrm{E}[x_i] = \mu, \qquad \text{Var}[x_i] = \sigma^2,\ \ i = 1, 2, \ldots, N,$$

and start by computing:

$$
\begin{aligned}
\mathrm{E}[\bar{x}_{N,\vec{B}} m_{N,\vec{\lambda}}] &= \frac{1}{w_{N,\vec{B}}}\mathrm{E}[w_{N,\vec{B}}\bar{x}_{N,\vec{B}} m_{N,\vec{\lambda}}] \\
&= \frac{1}{w_{N,\vec{B}}}\mathrm{E}\Big[(m_{N,\vec{B}})(m_{N,\vec{\lambda}})\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\mathrm{E}\Big[\Big(\sum_{i=1}^{N}(\prod_{p=i}^{N-1}B_p)x_i\Big)\Big(\sum_{j=1}^{N}(\prod_{q=j}^{N-1}\lambda)x_j\Big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\mathrm{E}\Big[\Big(\sum_{i=1}^{N}\beta_i x_i\Big)\Big(\sum_{j=1}^{N}\alpha_j x_j\Big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\mathrm{E}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j x_i x_j\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j \mathrm{E}\Big[x_i x_j\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\beta_i\alpha_j \mathrm{E}\Big[x_i x_j\Big] + \sum_{i=1}^{N}\sum_{j=i}^{N}\beta_i\alpha_j \mathrm{E}\Big[x_i x_j\Big]\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\beta_i\alpha_j \mathrm{E}\Big[x_i x_j\Big] + \sum_{i=1}^{N}\beta_i\alpha_i \mathrm{E}\Big[x_i^2\Big]\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1,j\neq i}^{N}\beta_i\alpha_j \mathrm{E}\big[x_i\big]\mathrm{E}\big[x_j\big] + \sum_{i=1}^{N}\beta_i\alpha_i\big(\mathrm{Var}[x_i] + \mathrm{E}[x_i]\mathrm{E}[x_i]\big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j \mathrm{E}\big[x_i\big]\mathrm{E}\big[x_j\big] + \sum_{i=1}^{N}\beta_i\alpha_i\big(\mathrm{Var}[x_i]\big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j \mathrm{E}\big[x_i\big]\mathrm{E}\big[x_j\big] + \sum_{i=1}^{N}\beta_i\alpha_i\big(\sigma^2\big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j(\mu\cdot\mu) + \sum_{i=1}^{N}\beta_i\alpha_i\big(\sigma^2\big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\alpha_j(\mu\cdot\mu)\Big] + \frac{1}{w_{N,\vec{B}}}\Big[\sum_{i=1}^{N}\beta_i\alpha_i\big(\sigma^2\big)\Big] \\
&= \frac{1}{w_{N,\vec{B}}}\Big[\Big(\sum_{i=1}^{N}\beta_i\Big)\Big(\sum_{j=1}^{N}\alpha_j\Big)(\mu^2)\Big] + \frac{1}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\big(\sigma^2\big) \\
&= (\mu^2)\frac{1}{w_{N,\vec{B}}}\Big[\Big(w_{N,\vec{B}}\Big)\Big(w_{N,\vec{\lambda}}\Big)\Big] + \big(\sigma^2\big)\frac{1}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}} \\
&= \mu^2\cdot w_{N,\vec{\lambda}} + \sigma^2\cdot\frac{1}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}} \qquad \text{(A.91)}
\end{aligned}
$$

Computing $E[S_{N,\vec{\lambda},\vec{B}}]$ is now straightforward:

$$\begin{aligned}
\mathrm{E}[S_{N,\vec{\lambda},\vec{B}}] &= \mathrm{E}\left[\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)x_k^2 - 2\bar{x}_{N,\vec{B}}m_{N,\vec{\lambda}} + \bar{x}_{N,\vec{B}}^2 w_{N,\vec{\lambda}}\right] \\
&= \mathrm{E}\left[\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)x_k^2\right] - 2\mathrm{E}\left[\bar{x}_{N,\vec{B}}m_{N,\vec{\lambda}}\right] + \mathrm{E}\left[\bar{x}_{N,\vec{B}}^2 w_{N,\vec{\lambda}}\right] \\
&= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\mathrm{E}[x_k^2] - 2\left(\mu^2\cdot w_{N,\vec{\lambda}} + \sigma^2\cdot\frac{1}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\right) \\
&\quad + w_{N,\vec{\lambda}}\mathrm{E}[\bar{x}_{N,\vec{B}}^2] \\
&= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\left(\mathrm{Var}[x_k] + \left(\mathrm{E}[x_k]\right)^2\right) - 2\left(\mu^2\cdot w_{N,\vec{\lambda}} + \frac{\sigma^2}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\right) \\
&\quad + w_{N,\vec{\lambda}}\left(\mathrm{Var}[\bar{x}_{N,\vec{B}}] + \left(\mathrm{E}[\bar{x}_{N,\vec{B}}]\right)^2\right) \\
&= w_{N,\vec{\lambda}}\left(\sigma^2+\mu^2\right) - 2\left(\mu^2\cdot w_{N,\vec{\lambda}} + \frac{\sigma^2}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\right) + w_{N,\vec{\lambda}}\left(u_{N,\vec{B}}\sigma^2+\mu^2\right) \\
&= w_{N,\vec{\lambda}}\left(\sigma^2\right) - 2\left(\frac{\sigma^2}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\right) + w_{N,\vec{\lambda}}\left(u_{N,\vec{B}}\sigma^2\right) \\
&= \left[(1+u_{N,\vec{B}})w_{N,\vec{\lambda}} - \frac{2}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}\right]\sigma^2 \qquad \text{(A.92)}
\end{aligned}$$

Therefore, setting

$$V_{N,\vec{\lambda},\vec{B}} = (1+u_{N,\vec{B}})w_{N,\vec{\lambda}} - \frac{2}{w_{N,\vec{B}}}P_{N,\vec{\lambda},\vec{B}}, \qquad \text{(A.93)}$$

results in

$$s_{N,\vec{\lambda},\vec{B}}^2 = \mathrm{E}\left[\frac{1}{V_{N,\vec{\lambda},\vec{B}}}S_{N,\vec{\lambda},\vec{B}}\right] = \sigma^2,$$

as desired. However, the definition of $V_{N,\vec{\lambda},\vec{B}}$ is in terms of $P_{N,\vec{\lambda},\vec{B}}$. In order to find a sequential update equation for $V_{N,\vec{\lambda},\vec{B}}$, we need a sequential update equation for $P_{N,\vec{\lambda},\vec{B}}$.

Recalling that the empty product equals 1,

$$\begin{aligned}
P_{N+1,\vec{\lambda},\vec{B}} &= \sum_{i=1}^{N+1}\left(\prod_{p=i}^{N} B_p\right)\left(\prod_{q=i}^{N}\lambda_q\right)\\
&= \sum_{i=1}^{N}\left(\prod_{p=i}^{N} B_p\right)\left(\prod_{q=i}^{N}\lambda_q\right) + \left(\prod_{p=N+1}^{N} B_p\right)\left(\prod_{q=N+1}^{N}\lambda_q\right)\\
&= B_N\lambda_N\sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left(\prod_{q=i}^{N-1}\lambda_q\right) + (1)\,(1)\\
&= B_N\lambda_N P_{N,\vec{\lambda},\vec{B}} + 1
\end{aligned}$$

## Summary of update equations for $V_{N,\vec{\lambda},\vec{B}}$

In order to update $V_{N,\vec{\lambda},\vec{B}}$ and $S_{N,\vec{\lambda},\vec{B}}$ the following quantities are needed:

$$x_{N+1}, \lambda_N, m_{N,\vec{\lambda}}, w_{N,\vec{\lambda}}, w_{N+1,\vec{\lambda}}, B_N, w_{N+1,\vec{B}}, u_{N+1,\vec{B}}, \bar{x}_{N,\vec{B}}, \bar{x}_{N+1,\vec{B}}.$$

The sequential update equations for $V_{N,\vec{\lambda},\vec{B}}$ are:

$$\begin{aligned}
w_{N+1,\vec{\lambda}} &= \lambda_N w_{N,\vec{\lambda}} + 1\\
w_{N+1,\vec{B}} &= B_N w_{N,\vec{B}} + 1\\
u_{N+1,\vec{B}} &= \left(1 - \frac{1}{w_{N+1,\vec{B}}}\right)^2 u_{N,\vec{B}} + \left(\frac{1}{w_{N+1,\vec{B}}}\right)^2\\
P_{N+1,\vec{\lambda},\vec{B}} &= B_N\lambda_N\left(P_{N,\vec{\lambda},\vec{B}}\right) + 1\\
V_{N+1,\vec{\lambda},\vec{B}} &= (1 + u_{N+1,\vec{B}})w_{N+1,\vec{\lambda}} - \frac{2}{w_{N+1,\vec{B}}}P_{N+1,\vec{\lambda},\vec{B}},
\end{aligned}$$

## Sequential update equations for $S_{N,\vec{\lambda},\vec{B}}$

In order to derive sequential update equations for $S_{N,\vec{\lambda},\vec{B}}$, it will be convenient to write

$$X_{k,N,\vec{B}} = \left[x_k - \bar{x}_{N,\vec{B}}\right]^2,$$

so that me may rewrite

$$S_{N,\vec{\lambda},\vec{B}} = \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)X_{k,N,\vec{B}}.$$

Now, recall the standard update equation for the mean in Equation (3.11), which gives

$$\bar{x}_{N+1,\vec{B}} = \bar{x}_{N,\vec{B}} - \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right).$$

This allows us to derive:

$$\begin{aligned}
X_{k,N+1,\vec{B}} &= \left[x_k - \bar{x}_{N+1,\vec{B}}\right]^2 \\
&= \left[x_k - \bar{x}_{N,\vec{B}} + \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)\right]^2 \\
&= \left[\left(x_k - \bar{x}_{N,\vec{B}}\right) + \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)\right]^2 \\
&= \left(x_k - \bar{x}_{N,\vec{B}}\right)^2 + 2\left[x_k - \bar{x}_{N,\vec{B}}\right]\frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right) + \\
&\quad + \left(\frac{1}{w_{N+1,\vec{B}}}\right)^2\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)^2
\end{aligned}$$

Then, defining the intermediate variables

$$\xi_{N+1,\vec{B}} = \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right) \tag{A.94}$$

$$\gamma_{k,N,\vec{B}} = 2\left[x_k - \bar{x}_{N,\vec{B}}\right]\frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right) \tag{A.95}$$

this then gives us

$$X_{k,N+1,\vec{B}} = X_{k,N,\vec{B}} + \gamma_{k,N,\vec{B}} + (\xi_{N+1,\vec{B}})^2 \tag{A.96}$$

Now, the sequential update equation for $S_{N,\vec{\lambda},\vec{B}}$:

$$\begin{aligned}
S_{N+1,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N+1} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} \\
&= \sum_{k=1}^{N} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} + \sum_{k=N+1}^{N+1} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} \\
&= \sum_{k=1}^{N} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} + \left( \prod_{p=N+1}^{N} \lambda_p \right) X_{N+1,N+1,\vec{B}} \\
&= \sum_{k=1}^{N} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} + (1) \left[ x_{N+1} - \bar{x}_{N+1,\vec{B}} \right]^2 \\
&= \sum_{k=1}^{N} \left( \prod_{p=k}^{N} \lambda_p \right) X_{k,N+1,\vec{B}} + \left[ x_{N+1} - \bar{x}_{N+1,\vec{B}} \right]^2 \\
&= \lambda_N \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) X_{k,N+1,\vec{B}} + \left[ x_{N+1} - \bar{x}_{N+1,\vec{B}} \right]^2 \\
&= \lambda_N \left[ \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \left( X_{k,N,\vec{B}} + \gamma_{k,N,\vec{B}} + (\xi_{N+1,\vec{B}})^2 \right) \right] + \left[ x_{N+1} - \bar{x}_{N+1,\vec{B}} \right]^2 \\
&= \lambda_N \left[ \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) X_{k,N,\vec{B}} + \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) \gamma_{k,N,\vec{B}} \right. \\
&\qquad \left. + \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) (\xi_{N+1,\vec{B}})^2 \right) \right] + \left[ x_{N+1} - \bar{x}_{N+1,\vec{B}} \right]^2
\end{aligned}$$

Breaking this equation up:

$$\sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p \right) X_{k,N,\vec{B}} = S_{N,\vec{\lambda},\vec{B}} \tag{A.97}$$

$$\begin{aligned}\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\gamma_{k,N,\vec{B}} &= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\left[x_k-\bar{x}_{N,\vec{B}}\right]\frac{2}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}}-x_{N+1}\right)\\ &= \left[m_{N,\vec{\lambda}}-w_{N,\vec{\lambda}}\bar{x}_{N,\vec{B}}\right]\frac{2}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}}-x_{N+1}\right)\\ &= 2\left[m_{N,\vec{\lambda}}-w_{N,\vec{\lambda}}\bar{x}_{N,\vec{B}}\right]\left(\xi_{N+1,\vec{B}}\right)\end{aligned} \tag{A.98}$$

$$\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)(\xi_{N+1,\vec{B}})^2 = w_{N,\vec{\lambda}}(\xi_{N+1,\vec{B}})^2 \tag{A.99}$$

And putting Equations (A.97), (A.98), (A.99) back together,

$$\begin{aligned}S_{N+1,\vec{\lambda},\vec{B}} &= \lambda_N\Bigg[\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)X_{k,N,\vec{B}}+\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\gamma_{k,N,\vec{B}}+\\ &\quad+\sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)(\xi_{N+1,\vec{B}})^2\Big)\Bigg]+\left[x_{N+1}-\bar{x}_{N+1,\vec{B}}\right]^2\\ &= \lambda_N\left[S_{N,\vec{\lambda},\vec{B}}+2\left[m_{N,\vec{\lambda}}-w_{N,\vec{\lambda}}\bar{x}_{N,\vec{B}}\right]\left(\xi_{N+1,\vec{B}}\right)+w_{N,\vec{\lambda}}(\xi_{N+1,\vec{B}})^2\right]+\\ &\quad+\left[x_{N+1}-\bar{x}_{N+1,\vec{B}}\right]^2\\ &= \lambda_N\left[S_{N,\vec{\lambda},\vec{B}}+2m_{N,\vec{\lambda}}\xi_{N+1,\vec{B}}-w_{N,\vec{\lambda}}\xi_{N+1,\vec{B}}\left(2\bar{x}_{N,\vec{B}}-\xi_{N+1,\vec{B}}\right)\right]+\\ &\quad+\left[x_{N+1}-\bar{x}_{N+1,\vec{B}}\right]^2\\ &= \lambda_N\left[S_{N,\vec{\lambda},\vec{B}}+\xi_{N+1,\vec{B}}\left[2m_{N,\vec{\lambda}}-w_{N,\vec{\lambda}}\left(2\bar{x}_{N,\vec{B}}-\xi_{N+1,\vec{B}}\right)\right]\right]+\\ &\quad+\left[x_{N+1}-\bar{x}_{N+1,\vec{B}}\right]^2\\ &= \lambda_N\left[S_{N,\vec{\lambda},\vec{B}}+H_{N+1,\vec{\lambda},\vec{B}}\right]+\left[x_{N+1}-\bar{x}_{N+1,\vec{B}}\right]^2\end{aligned}$$

where

$$H_{N+1,\vec{\lambda},\vec{B}} = \xi_{N+1,\vec{B}}\big[2m_{N,\vec{\lambda}} - w_{N,\vec{\lambda}}\big(2\bar{x}_{N,\vec{B}} - \xi_{N+1,\vec{B}}\big)\big]. \tag{A.100}$$

But, all the terms in $H_{N+1,\vec{\lambda},\vec{B}}$ already have update equations, or are defined in terms of simpler terms, there the sequential update equation for $S_{N+1,\vec{\lambda},\vec{B}}$ is indeed

$$S_{N+1,\vec{\lambda},\vec{B}} = \lambda_N\left[S_{N,\vec{\lambda},\vec{B}} + H_{N+1,\vec{\lambda},\vec{B}}\right] + \big[x_{N+1} - \bar{x}_{N+1,\vec{B}}\big]^2. \tag{A.101}$$

## Summary of update equations for $S_{N,\vec{\lambda},\vec{B}}$

In order to update $S_{N+1,\vec{\lambda},\vec{B}}$,

$$\begin{aligned}
\bar{x}_{N,\vec{B}} &= \bar{x}_{N-1,\vec{B}} - \frac{1}{w_{N,\vec{B}}}\left(\bar{x}_{N-1,\vec{B}} - x_N\right)\\
\xi_{N+1,\vec{B}} &= \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)\\
H_{N+1,\vec{\lambda},\vec{B}} &= \xi_{N+1,\vec{B}}\big[2m_{N,\vec{\lambda}} - w_{N,\vec{\lambda}}\big(2\bar{x}_{N,\vec{B}} - \xi_{N+1,\vec{B}}\big)\big]\\
S_{N,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)\big[x_k - \bar{x}_{N,\vec{B}}\big]^2\\
S_{1,\vec{\lambda},\vec{B}} &= 0\\
S_{N+1,\vec{\lambda},\vec{B}} &= \lambda_N\left[S_{N,\vec{\lambda},\vec{B}} + H_{N+1,\vec{\lambda},\vec{B}}\right] + \big[x_{N+1} - \bar{x}_{N+1,\vec{B}}\big]^2\\
m_{N+1,\vec{\lambda}} &= \lambda_N m_{N,\vec{\lambda}} + x_{N+1}\\
w_{N+1,\vec{\lambda}} &= \lambda_N w_{N,\vec{\lambda}} + 1.
\end{aligned}$$

Finally,

$$s^2_{N,\vec{\lambda},\vec{B}} = \frac{1}{V_{N,\vec{\lambda},\vec{B}}} S_{N,\vec{\lambda},\vec{B}}.$$

### A.6.7 Derivative of the adaptive forgetting factor variance

We start by defining the derivative of $S_{N,\overrightarrow{\lambda},\overrightarrow{B}}$, again using Lemma 1 proved in Appendix A.3.3.

$$\begin{aligned}
S_{N,\overrightarrow{\lambda},\overrightarrow{B}} &= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)X_{k,N,\overrightarrow{B}} \\
\Rightarrow S_{N,\overrightarrow{\lambda},\overrightarrow{B}+\epsilon} &= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}(\lambda_p+\epsilon)\right)X_{k,N,\overrightarrow{B}} \\
&= \sum_{k=1}^{N}\left[\prod_{p=k}^{N-1}\lambda_p + \epsilon\left(\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\right) + O(\epsilon^2)\right]X_{k,N,\overrightarrow{B}} \\
&= \sum_{k=1}^{N}\left(\prod_{p=k}^{N-1}\lambda_p\right)X_{k,N,\overrightarrow{B}} + \epsilon\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\right)X_{k,N,\overrightarrow{B}} + O(\epsilon^2) \\
&= S_{N,\overrightarrow{\lambda},\overrightarrow{B}} + \epsilon\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\right)X_{k,N,\overrightarrow{B}} + O(\epsilon^2) \qquad \text{(A.102)}
\end{aligned}$$

Therefore, we can use Equation (A.102) to define the derivative to be

$$\begin{aligned}
Z_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \frac{\partial}{\partial\overrightarrow{\lambda}}S_{N,\overrightarrow{\lambda},\overrightarrow{B}} &= \lim_{\epsilon\to 0}\frac{1}{\epsilon}\left[S_{N,\overrightarrow{\lambda},\overrightarrow{B}+\epsilon} - S_{N,\overrightarrow{\lambda},\overrightarrow{B}}\right] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\right)X_{k,N,\overrightarrow{B}}.
\end{aligned}$$

Therefore,

$$Z_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\right)\right)X_{k,N,\overrightarrow{B}} \qquad \text{(A.103)}$$

Using the expression for $X_{k,N+1,\overrightarrow{B}}$ in Equation (A.96), we have

$$\begin{aligned}
Z_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N+1,\overrightarrow{B}} \\
&= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\left[X_{k,N,\overrightarrow{B}} + \gamma_{k,N,\overrightarrow{B}} + (\xi_{N+1,\overrightarrow{B}})^2\right] \\
&= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\left[\gamma_{k,N,\overrightarrow{B}} + (\xi_{N+1,\overrightarrow{B}})^2\right] \\
&= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} \qquad \text{(A.104)}
\end{aligned}$$

where

$$F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\left[\gamma_{k,N,\overrightarrow{B}} + (\xi_{N+1,\overrightarrow{B}})^2\right], \qquad \text{(A.105)}$$

where are $\gamma_{k,N}$ and $\xi_{N+1,\overrightarrow{B}}$ are defined in Equations (A.94) and (A.95). Now, some manipulation of the expression in Equation (A.104) yields:

$$\begin{aligned}
Z_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + \sum_{k=N+1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + \left(\sum_{t=N+1}^{N}\Big(\prod_{\substack{p=N+1\\p\neq t}}^{N}\lambda_p\Big)\right)X_{N+1,N,\overrightarrow{B}} + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\overrightarrow{B}} + (0) + F_{N+1,\overrightarrow{\lambda},\overrightarrow{B}}
\end{aligned}$$

Now,

$$\begin{aligned}
Z_{N+1,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \sum_{t=N}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \Big(\prod_{\substack{p=k\\p\neq N}}^{N}\lambda_p\Big)\right)X_{k,N,\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)X_{k,N,\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\vec{B}} + \sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)X_{k,N,\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)X_{k,N,\vec{B}} + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}}
\end{aligned}$$

Now, this is equivalent to:

$$\begin{aligned}
Z_{N+1,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left[\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\Big(\prod_{\substack{p=N\\p\neq t}}^{N}\lambda_p\Big)\right]\right)X_{k,N,\vec{B}} + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left[\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\Big(\prod_{p=N}^{N}\lambda_p\Big)\right]\right)X_{k,N,\vec{B}} + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\left[\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\Big(\lambda_N\Big)\right]\right)X_{k,N,\vec{B}} + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \lambda_N\left[\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right)X_{k,N,\vec{B}}\right] + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}} \\
&= \lambda_N Z_{N,\vec{\lambda},\vec{B}} + S_{N,\vec{\lambda},\vec{B}} + F_{N+1,\vec{\lambda},\vec{B}}
\end{aligned}$$

using the expression for $Z_{N,\vec{\lambda},\vec{B}}$ in Equation A.103. This provides a sequential update equation for $Z_{N,\vec{\lambda},\vec{B}}$. However, we need a sequential update equation for $F_{N+1,\vec{\lambda},\vec{B}}$. Starting with the non-sequential form in Equation (A.105),

$$\begin{aligned}
F_{N+1,\vec{\lambda},\vec{B}} &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}} + (\xi_{N+1,\vec{B}})^2\Big]\\
&= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] + \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[(\xi_{N+1,\vec{B}})^2\Big]
\end{aligned}$$

Now set,

$$\begin{aligned}
A_1 &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big]\\
A_2 &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[(\xi_{N+1,\vec{B}})^2\Big]
\end{aligned}$$

and recall the definitions of $\gamma_{k,N}$ and $\xi_{N+1,\vec{B}}$ from Equations (A.94) and (A.95).

$$\begin{aligned}
\xi_{N+1,\vec{B}} &= \frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)\\
\gamma_{N,N,\vec{B}} &= 2\Big[x_N - \bar{x}_{N,\vec{B}}\Big]\frac{1}{w_{N+1,\vec{B}}}\left(\bar{x}_{N,\vec{B}} - x_{N+1}\right)
\end{aligned}$$

$$\begin{aligned}
A_1 &= \sum_{k=1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] + \sum_{k=N+1}^{N+1}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] + \left(\sum_{t=N+1}^{N}\Big(\prod_{\substack{p=N+1\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{N+1,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] + \Big(0\Big)\Big[\gamma_{N,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \sum_{t=N}^{N}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] + 0 \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \Big(\prod_{\substack{p=k\\p\neq N}}^{N}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big) + \Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)\Big[\gamma_{k,N,\vec{B}}\Big] \\
&= \sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}} + \sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}} \\
&= \lambda_N\left[\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}}\right] + \sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}}
\end{aligned}$$

$$\Rightarrow A_1 = \lambda_N C_1 + C_2$$

where

$$C_1 = \sum_{k=1}^{N} \left( \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) \right) \gamma_{k,N,\overrightarrow{B}}$$

$$C_2 = \sum_{k=1}^{N} \left( \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \right) \gamma_{k,N,\overrightarrow{B}}$$

Recall the non-sequential equations for $\Delta_{N,\overrightarrow{\lambda}}$ and $\Omega_{N,\overrightarrow{\lambda}}$ from Equations (A.19) and (A.26),

$$\Delta_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \left( \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) x_k \right)$$

$$\Omega_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \left( \sum_{t=k}^{N-1} \Big( \prod_{\substack{p=k \\ p \neq t}}^{N-1} \lambda_p \Big) \right)$$

and note the expansion of $\gamma_{k,N}$ into

$$\begin{aligned} \gamma_{k,N,\overrightarrow{B}} &= 2\left[x_k - \bar{x}_{N,\overrightarrow{B}}\right] \frac{1}{w_{N+1,\overrightarrow{B}}} \left(\bar{x}_{N,\overrightarrow{B}} - x_{N+1}\right) \\ &= x_k \left[ 2 \frac{1}{w_{N+1,\overrightarrow{B}}} \left(\bar{x}_{N,\overrightarrow{B}} - x_{N+1}\right) \right] - 2 \frac{1}{w_{N+1,\overrightarrow{B}}} \left[\bar{x}_{N,\overrightarrow{B}}\right] \left(\bar{x}_{N,\overrightarrow{B}} - x_{N+1}\right) \end{aligned}$$

Then, we can compute $C_1$:

$$\begin{aligned}
C_1 &= \left[\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}}\right]\\
&= \left[\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right)x_k\right]\left[2\frac{1}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\right]-\\
&\quad-\left[\sum_{k=1}^{N}\left(\sum_{t=k}^{N-1}\Big(\prod_{\substack{p=k\\p\neq t}}^{N-1}\lambda_p\Big)\right)\right]\left[2\frac{1}{w_{N+1,\vec{B}}}\left[\bar{x}_{N,\vec{B}}\right]\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\right]\\
&= \Delta_{N,\vec{\lambda}}\left[2\frac{1}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\right]-\\
&\quad-\Omega_{N,\vec{\lambda}}\left[2\frac{1}{w_{N+1,\vec{B}}}\left[\bar{x}_{N,\vec{B}}\right]\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\right]\\
&= \left[\Delta_{N,\vec{\lambda}}-\Omega_{N,\vec{\lambda}}\bar{x}_{N,\vec{B}}\right]\left[2\frac{1}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\right]
\end{aligned}$$

Now for $C_2$:

$$\begin{aligned}
\gamma_{k,N,\vec{B}} &= \left[x_k-\bar{x}_{N,\vec{B}}\right]\frac{2}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\\
C_2 &= \sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)\gamma_{k,N,\vec{B}}\\
&= \left[\sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)x_k-\bar{x}_{N,\vec{B}}\sum_{k=1}^{N}\left(\Big(\prod_{p=k}^{N-1}\lambda_p\Big)\right)\right]\frac{2}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\\
&= \left[w_{N,\vec{\lambda}}\bar{x}_{N,\vec{\lambda}}-\bar{x}_{N,\vec{B}}w_{N,\vec{\lambda}}\right]\frac{2}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)\\
&= \left[\bar{x}_{N,\vec{\lambda}}-\bar{x}_{N,\vec{B}}\right]\frac{2w_{N,\vec{\lambda}}}{w_{N+1,\vec{B}}}\Big(\bar{x}_{N,\vec{B}}-x_{N+1}\Big)
\end{aligned}$$

## Derivative of $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$

Finally, we derive update equations for the derivative of $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$. Recall Equations (A.90) and (A.93),

$$\begin{aligned}
P_{N,\overrightarrow{\lambda},\overrightarrow{B}} &= \sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left(\prod_{q=i}^{N-1}\lambda_q\right),\\
V_{N,\overrightarrow{\lambda},\overrightarrow{B}} &= (1+u_{N,\overrightarrow{B}})w_{N,\overrightarrow{\lambda}} - \frac{2}{w_{N,\overrightarrow{B}}}P_{N,\overrightarrow{\lambda},\overrightarrow{B}},
\end{aligned}$$

and so

$$P_{N,\overrightarrow{\lambda}+\epsilon,\overrightarrow{B}} = \sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left(\prod_{q=i}^{N-1}(\lambda_q+\epsilon)\right).$$

Then, we can define the derivative by,

$$\begin{aligned}
\Upsilon_{N,\overrightarrow{\lambda},\overrightarrow{B}} &= \frac{\partial}{\partial\overrightarrow{\lambda}}P_{N,\overrightarrow{\lambda},\overrightarrow{B}}\\
&= \lim_{\epsilon\to\infty}\frac{1}{\epsilon}\left[P_{N,\overrightarrow{\lambda}+\epsilon,\overrightarrow{B}} - P_{N,\overrightarrow{\lambda},\overrightarrow{B}}\right]\\
&= \lim_{\epsilon\to\infty}\frac{1}{\epsilon}\left[\sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left(\prod_{q=i}^{N-1}(\lambda_q+\epsilon)\right) - \sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left(\prod_{q=i}^{N-1}\lambda_q\right),\right]\\
&= \lim_{\epsilon\to\infty}\frac{1}{\epsilon}\left[\sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left\{\left(\prod_{q=i}^{N-1}(\lambda_q+\epsilon)\right) - \left(\prod_{q=i}^{N-1}\lambda_q\right)\right\}\right]\\
&= \lim_{\epsilon\to\infty}\frac{1}{\epsilon}\left[\sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left\{\epsilon\sum_{t=i}^{N-1}\Big(\prod_{\substack{q=i\\q\neq t}}^{N-1}\lambda_p\Big) + O(\epsilon^2)\right\}\right]\\
&= \lim_{\epsilon\to\infty}\left[\sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\left\{\sum_{t=i}^{N-1}\Big(\prod_{\substack{q=i\\q\neq t}}^{N-1}\lambda_p\Big) + O(\epsilon)\right\}\right]\\
&= \sum_{i=1}^{N}\left(\prod_{p=i}^{N-1} B_p\right)\sum_{t=i}^{N-1}\Big(\prod_{\substack{q=i\\q\neq t}}^{N-1}\lambda_p\Big) \qquad \text{(A.106)}
\end{aligned}$$

It can be shown that Equation (A.106) is equivalent to

$$\Upsilon_{N,\overrightarrow{\lambda},\overrightarrow{B}} = \sum_{i=1}^{N-1} \left( \prod_{p=i}^{N-1} B_p \right) \sum_{t=i}^{N-1} \Big( \prod_{\substack{q=i \\ q\neq t}}^{N-1} \lambda_p \Big) \tag{A.107}$$

(the first summation sums to $N-1$ rather than $N$). Similar to the derivation for the update equation for $\Delta_{N,\overrightarrow{\lambda}}$, we obtain:

$$\begin{aligned}
\Upsilon_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} &= \sum_{i=1}^{N} \left( \prod_{p=i}^{N} B_p \right) \sum_{t=i}^{N} \Big( \prod_{\substack{q=i \\ q\neq t}}^{N} \lambda_p \Big) \\
&= \sum_{i=1}^{N} \left( \prod_{p=i}^{N} B_p \right) \left( \sum_{t=i}^{N-1} \Big( \prod_{\substack{q=i \\ q\neq t}}^{N} \lambda_p \Big) + \Big( \prod_{\substack{q=i \\ q\neq N}}^{N} \lambda_p \Big) \right) \\
&= \sum_{i=1}^{N} B_N \left( \prod_{p=i}^{N-1} B_p \right) \left( \sum_{t=i}^{N-1} \lambda_N \Big( \prod_{\substack{q=i \\ q\neq t}}^{N-1} \lambda_p \Big) + \Big( \prod_{q=i}^{N-1} \lambda_p \Big) \right) \\
&= B_N \lambda_N \sum_{i=1}^{N} \left( \prod_{p=i}^{N-1} B_p \right) \sum_{t=i}^{N-1} \Big( \prod_{\substack{q=i \\ q\neq t}}^{N-1} \lambda_p \Big) + \sum_{i=1}^{N} \left( \prod_{p=i}^{N-1} B_p \right) \Big( \prod_{q=i}^{N-1} \lambda_p \Big)
\end{aligned}$$

$$\Rightarrow \Upsilon_{N+1,\overrightarrow{\lambda},\overrightarrow{B}} = B_N \lambda_N \Upsilon_{N,\overrightarrow{\lambda},\overrightarrow{B}} + P_{N,\overrightarrow{\lambda},\overrightarrow{B}}. \tag{A.108}$$

### A.6.8 Summary of AFF variance with a single forgetting factor

The adaptive forgetting factor variance, for a single forgetting factor $\overrightarrow{\lambda}$, can be defined as in Equation (8.25) by

$$s^2_{N,\overrightarrow{\lambda}+\epsilon} = \frac{1}{v_{N,\overrightarrow{\lambda}}} \sum_{k=1}^{N} \left( \prod_{p=k}^{N-1} \lambda_p + \epsilon \right) \left[ x_k - \bar{x}_{N,\overrightarrow{\lambda}+\epsilon} \right]^2 .$$

If we defined the sum of the residuals at time $N$ to be

$$\zeta_{N,\overrightarrow{\lambda}} = \sum_{k=1}^{N} \Big( \prod_{p=k}^{N-1} \lambda_p \Big) \Big[ x_k - \bar{x}_{N-1,\overrightarrow{\lambda}} \Big]^2, \tag{A.109}$$

it is possible to obtain the update equation

$$\zeta_{N,\overrightarrow{\lambda}} = \lambda_{N-1} \varepsilon_{N-1,\overrightarrow{\lambda}} + \alpha^2_{N,N-2} + Q_N,$$

where

$$\alpha_{k,N} = x_k - \bar{x}_{N,\overrightarrow{\lambda}} \tag{A.110}$$

$$\xi_{N-2} = \left( \frac{1}{w_{N-1,\overrightarrow{\lambda}}} \right) \left( \bar{x}_{N-2,\overrightarrow{\lambda}} - x_{N-1} \right) \tag{A.111}$$

$$Q_N = 2\xi_{N-2} \left[ m_{N,\overrightarrow{\lambda}} - \bar{x}_{N-2,\overrightarrow{\lambda}} w_{N,\overrightarrow{\lambda}} \right] + \big(\xi_{N-2}\big)^2 w_{N,\overrightarrow{\lambda}}. \tag{A.112}$$

A summary of equations for $\frac{\partial}{\partial \overrightarrow{\lambda}} \zeta_{N,\overrightarrow{\lambda}}$ is:

$$\frac{\partial}{\partial \overrightarrow{\lambda}} \zeta_{N,\overrightarrow{\lambda}} = \Phi_{N,\overrightarrow{\lambda}} + \Psi_{N,\overrightarrow{\lambda}} \tag{A.113}$$

$$\Phi_{N,\overrightarrow{\lambda}} = -2 \left( \frac{\Delta_{N-1,\overrightarrow{\lambda}}}{w_{N-1,\overrightarrow{\lambda}}} \right) \left[ m_{N,\overrightarrow{\lambda}} - w_{N,\overrightarrow{\lambda}} \bar{x}_{N-1,\overrightarrow{\lambda}} \right] \tag{A.114}$$

$$\Psi_{N,\overrightarrow{\lambda}} = \lambda_{N-1} \Psi_{N-1,\overrightarrow{\lambda}} + \varepsilon_{N-1,\overrightarrow{\lambda}} + 2\xi_{N-2} \left[ \Delta_N - \Omega_N \bar{x}_{N-2,\overrightarrow{\lambda}} \right] + \Omega_N \big(\xi_{N-2}\big)^2. \tag{A.115}$$

The derivative of $v_{N,\overrightarrow{\lambda}}$ is omitted, but can be computed by following the derivation for $V_{N,\overrightarrow{\lambda},\overrightarrow{B}}$. The initial values of the above quantities are given by:

$$\begin{aligned}
\zeta_{0,\overrightarrow{\lambda}} &= 0 \\
\zeta_{1,\overrightarrow{\lambda}} &= x_1^2 \\
\Psi_{1,\overrightarrow{\lambda}} &= 0 \\
\Psi_{2,\overrightarrow{\lambda}} &= 0 \\
\Phi_{1,\overrightarrow{\lambda}} & \text{ is undefined} \\
\Phi_{2,\overrightarrow{\lambda}} &= 0
\end{aligned}$$

Note that $\alpha_{N,N-2}$, $\xi_{N-2}$ and $Q_N$ are only defined for $N = 3, 4, \ldots$. Therefore,

$$\begin{aligned}
&\frac{\partial}{\partial \overrightarrow{\lambda}} \zeta_{1,\overrightarrow{\lambda}} \text{ is undefined} \\
&\frac{\partial}{\partial \overrightarrow{\lambda}} \zeta_{2,\overrightarrow{\lambda}} = 0 \\
&\frac{\partial}{\partial \overrightarrow{\lambda}} \zeta_{N,\overrightarrow{\lambda}} = \Phi_{N,\overrightarrow{\lambda}} + \Psi_{N,\overrightarrow{\lambda}}, \qquad N = 3, 4, \ldots.
\end{aligned}$$

# References

[1] Cisco IOS NetFlow. `http://www.cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html`. Accessed: 2014-08-29.

[2] I. A. Ajwa, Z. Liu, and P. S. Wang. Gröbner bases algorithm. Technical report, Kent State University, 2003.

[3] C. Anagnostopoulos. *A statistical framework for streaming data analysis*. PhD thesis, Imperial College London, 2010.

[4] C. Anagnostopoulos, D. K. Tasoulis, N. M. Adams, N. G. Pavlidis, and D. J. Hand. Online linear and quadratic discriminant analysis with adaptive forgetting for streaming classification. *Statistical Analysis and Data Mining*, 5(2):139–166, 2012.

[5] D. W. Apley and C. H. Chin. An optimal filter design approach to statistical process control. *Journal of Quality Technology*, 39(2):93–117, 2007.

[6] K. J. Åström, U. Borisson, L. Ljung, and B. Wittenmark. Theory and applications of self-tuning regulators. *Automatica*, 13(5):457–476, 1977.

[7] K. J. Åström and B. Wittenmark. On self tuning regulators. *Automatica*, 9(2):185–199, 1973.

[8] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 1–16. ACM, 2002.

[9] M. Basseville and I. V. Nikiforov. *Detection of Abrupt Changes: Theory and Application*. Prentice Hall Englewood Cliffs, 1993.

[10] P. M. Bentler and J. Xie. Corrections to test statistics in principal Hessian directions. *Statistics & Probability Letters*, 47(4):381–389, 2000.

[11] K. Bhaduri, K. Das, K. Borne, C. Giannella, T. Mahule, and H. Kargupta. Scalable, asynchronous, distributed eigen monitoring of astronomy data streams. *Statistical Analysis and Data Mining*, 4(3):336–352, 2011.

[12] R. B. Blazek, H. Kim, B. L. Rozovskii, and A. G. Tartakovsky. A novel approach to detection of denial-of-service attacks via adaptive sequential and batch-sequential change-point detection methods. In *Proceedings of the IEEE systems, man and cybernetics information assurance workshop*, pages 220–226, 2001.

[13] D. A. Bodenham and N. M. Adams. Continuous monitoring of a computer network using multivariate adaptive estimation. In *Proceedings of the IEEE 13th International Conference on Data Mining Workshops (ICDMW)*, pages 311–388, 2013.

[14] D. A. Bodenham and N. M. Adams. A comparison of efficient approximations for a weighted sum of chi-squared random variables. *Compuational Statistics and Data Analysis (submitted)*, 2014.

[15] D. A. Bodenham and N. M. Adams. Continuous changepoint monitoring of data streams using adaptive estimation. *Pattern Analysis and Machine Intelligence (submitted)*, 2014.

[16] D. A. Bodenham and N. M. Adams. Adaptive change detection for relay-like behaviour. In *Proceedings of the IEEE Joint Information and Security Informatics Conference, 2014*, 2014, accepted.

[17] L. Bottou. Stochastic learning. In *Advanced Lectures on Machine Learning*, pages 146–168. Springer, 2004.

[18] G. E. P. Box. Some theorems on quadratic forms applied in the study of analysis of variance problems, I. Effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics*, 25(2):290–302, 1954.

[19] S. P. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[20] M. B. Brown. 400: A method for combining non-independent, one-sided tests of significance. *Biometrics*, 31:987–992, 1975.

[21] B. Buchberger. *An algorithm for finding the bases elements of the residue class ring modulo a zero-dimensional polynomial ideal (German)*. PhD thesis, University of Innsbruck, Austria, 1965.

[22] B. Buchberger. An algorithmical criterion for the solvability of algebraic systems of equations (German). *Aequationes Mathematicae*, 4(3):374–383, 1970.

[23] B. Buchberger and F. Winkler. *Gröbner bases and applications*, volume 251. Cambridge University Press, 1998.

[24] M. J. Buckley and G. K. Eagleson. An approximation to the distribution of quadratic forms in normal random variables. *Australian Journal of Statistics*, 30(1):150–159, 1988.

[25] H. S. Burkom, S. Murphy, J. Coberly, and K. Hurt-Mullen. Analytic methods: Public health monitoring tools for multiple data streams. *Morbidity and Mortality Weekly Report*, 54:55–62, 2005.

[26] G. Capizzi and G. Masarotto. An adaptive exponentially weighted moving average control chart. *Technometrics*, 45(3):199–207, 2003.

[27] G. Capizzi and G. Masarotto. Self-starting CUSCORE control charts for individual multivariate observations. *Journal of Quality Technology*, 42(2):136–152, 2010.

[28] G. Capizzi and G. Masrotto. Adaptive generalized likelihood ratio control charts for detecting unknown patterned mean shifts. *Journal of Quality Technology*, 44(4):281–303, 2012.

[29] P. Casas, S. Vaton, L. Fillatre, and I. Nikiforov. Optimal volume anomaly detection and isolation in large-scale IP networks using coarse-grained measurements. *Computer Networks*, 54(11):1750–1766, 2010.

[30] P. Castagliola and P. Maravelakis. A CUSUM control chart for monitoring the variance when parameters are estimated. *Journal of Statistical Planning and Inference*, 141(4):1463–1478, 2011.

[31] A. Castaño-Martínez and F. López-Blázquez. Distribution of a sum of weighted central chi-square variables. *Communications in Statistics–Theory and Methods*, 34(3):515–524, 2005.

[32] C. W. Champ and W. H. Woodall. Exact results for Shewhart control charts with supplementary runs rules. *Technometrics*, 29(4):393–399, 1987.

[33] J. Chen and A. K. Gupta. Testing and locating variance changepoints with application to stock prices. *Journal of the American Statistical Association*, 92(438):739–747, 1997.

[34] J.-C. Chen. Testing goodness of fit of polynomial models via spline smoothing techniques. *Statistics & Probability Letters*, 19(1):65–76, 1994.

[35] Z. Chen. Is the weighted Z-test the best method for combining probabilities from independent tests? *Journal of Evolutionary Biology*, 24(4):926–930, 2011.

[36] S. W. Choi, E. B. Martin, A. J. Morris, and I.-B. Lee. Adaptive multivariate statistical process control for monitoring time-varying processes. *Industrial & Engineering Chemistry Research*, 45(9):3108–3118, 2006.

[37] F. Choobineh and J. L. Ballard. Control-limits of QC charts for skewed distributions using weighted-variance. *Reliability, IEEE Transactions on*, 36(4):473–477, 1987.

[38] G. Covarrubias, B. T. Ewing, S. E. Hein, and M. A. Thompson. Modeling volatility changes in the 10-year treasury. *Physica A: Statistical Mechanics and its Applications*, 369(2):737–744, 2006.

[39] D. A. Cox, J. Little, and D. O'Shea. *Ideals, Varieties, and Algorithms: An Introduction to Computational Algebraic Geometry and Commutative Algebra*. Springer, 2007.

[40] R. B. Crosier. Multivariate generalizations of cumulative sum quality-control schemes. *Technometrics*, 30(3):291–303, 1988.

[41] R. B. Davies. Algorithm as 155: The distribution of a linear combination of $\chi^2$ random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29(3):323–333, 1980.

[42] A. W. Davis. A differential equation approach to linear combinations of independent chi-squares. *Journal of the American Statistical Association*, 72(357):212–214, 1977.

[43] P. Duchesne and P. Lafaye De Micheaux. Computing the distribution of quadratic forms: Further comparisons between the Liu–Tang–Zhang approximation and exact methods. *Computational Statistics & Data Analysis*, 54(4):858–862, 2010.

[44] C. Estan and G. Varghese. Data streaming in computer networking. In *Proceedings of the Workshop on Management and processing of Data Streams*, pages 1–3, 2003.

[45] J. M. Estevez-Tapiador, P. Garcia-Teodoro, and J. E. Diaz-Verdejo. Anomaly detection methods in wired networks: a survey and taxonomy. *Computer Communications*, 27(16):1569–1584, 2004.

[46] H. Fairfield-Smith. The problem of comparing the results of two experiments with unequal errors. *Journal Council for Scientific and Industrial Research Australia*, 9:211–212, 1936.

[47] R. W. Farebrother. Algorithm AS 204: The distribution of a positive linear combination of $\chi^2$ random variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 33(3):332–339, 1984.

[48] W. Feller. *An Introduction to Probability Theory and its Applications*, volume 1. John Wiley & Sons, 1966.

[49] R. A. Fisher. *Statistical Methods for Research Workers*. Oliver and Boyd, Edinburgh, 1925.

[50] T. R. Fortescue, L. S. Kershenbaum, and B. E. Ydstie. Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, 17(6):831–835, 1981.

[51] S. E. Fraker, W. H. Woodall, and S. Mousavi. Performance metrics for surveillance schemes. *Quality Engineering*, 20(4):451–464, 2008.

[52] M. Frisén. Statistical surveillance. Optimality and methods. *International Statistical Review*, 71(2):403–434, 2003.

[53] J. Gama, P. P. Rodrigues, E. J. Spinosa, and A. C. P. L. Ferreira de Carvalho. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC Boca Raton, 2010.

[54] F. F. Gan. Joint monitoring of process mean and variance using exponentially weighted moving average control charts. *Technometrics*, 37(4):446–453, 1995.

[55] R. R. German, L. M. Lee, J. M. Horan, R. L. Milstein, C. A. Pertowski, and M. N. Waller. Updated guidelines for evaluating public health surveillance systems. *Morbidity and Mortality Weekly Report*, 50:1–35, 2001.

[56] T. M. Gil. *MULTOPS: A data structure for denial-of-service attack detection*. PhD thesis, Massachusetts Institute of Technology, 2000.

[57] G. H. Gonnet and M. B. Monagan. Solving systems of algebraic equations, or the interface between software and mathematics, computer science research report cs-89-13. Technical report, University of Waterloo, 1989.

[58] F. Gustafsson. *Adaptive Filtering and Change Detection*. Wiley Online Library, 2000.

[59] P. Hall. Chi squared approximations to the distribution of a sum of independent random variables. *The Annals of Probability*, 11(4):1028–1036, 1983.

[60] D. Han and F. Tsung. A generalized EWMA control chart and its comparison with the optimal EWMA, CUSUM and GLR schemes. *The Annals of Statistics*, 32(1):316–339, 2004.

[61] D. J. Hand and D. J. Weston. Statistical techniques for fraud detection, prevention, and assessment. *Mining Massive Data Sets for Security*, pages 257–270, 2008.

[62] D. M. Hawkins. Self-starting CUSUM charts for location and scale. *The Statistician*, pages 299–316, 1987.

[63] D. M. Hawkins. Cumulative sum control charting: an underutilized SPC tool. *Quality Engineering*, 5(3):463–477, 1993.

[64] D. M. Hawkins, S. Choi, and S. Lee. A general multivariate exponentially weighted moving-average control chart. *Journal of Quality Technology*, 39(2):118–125, 2007.

[65] D. M. Hawkins and E. M. Maboudou-Tchao. Self-starting multivariate exponentially weighted moving average control charting. *Technometrics*, 49(2):199–209, 2007.

[66] D. M. Hawkins, P. Qiu, and C. W. Kang. The changepoint model for statistical process control. *Journal of Quality Technology*, 35(4):355–366, 2003.

[67] D. M Hawkins and Q. Wu. The CUSUM and the EWMA head-to-head. *Quality Engineering*, 26(2):215–222, 2014.

[68] D. M. Hawkins and K. D. Zamba. A change-point model for a shift in variance. *Journal of Quality Technology*, 37(1):21–31, 2005.

[69] S. Haykin. *Adaptive Filter Theory, 2002*. Prentice-Hall, 2002.

[70] J. D. Healy. A note on multivariate CUSUM procedures. *Technometrics*, 29(4):409–412, 1987.

[71] M. R. Henzinger, P. Raghavan, and S. Rajagopalan. Computing on data streams. In *External Memory Algorithms: DIMACS Workshop External Memory Algorithms and Visualization, May 20-22, 1998*, volume 50, page 107. AMS Bookstore, 1999.

[72] H. Hotelling. Multivariate quality control, illustrated by the air testing of sample bombsights. In C. Eisenhart, editor, *Techniques of statistical analysis*, pages 113–184, 1947.

[73] G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106. ACM, 2001.

[74] J. P. Imhof. Computing the distribution of quadratic forms in normal variables. *Biometrika*, 48(3/4):419–426, 1961.

[75] B. R. Jayasuriya. Testing for polynomial regression using nonparametric regression techniques. *Journal of the American Statistical Association*, 91(436):1626–1631, 1996.

[76] D. R. Jensen and H. Solomon. A Gaussian approximation to the distribution of a definite quadratic form. *Journal of the American Statistical Association*, 67(340):898–902, 1972.

[77] W. A. Jensen, L. Jones-Farmer, C. W. Champ, and W. H. Woodall. Effects of parameter estimation on control chart properties: a literature review. *Journal of Quality Technology*, 38(4):349–364, 2006.

[78] W. Jiang, W. Shu, and D. W. Apley. Adaptive CUSUM procedures with EWMA-based shift estimators. *IIE Transactions*, 40(10):992–1003, 2008.

[79] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Univariate Distributions*, volume 2. John Wiley & Sons, 1995.

[80] N. L. Johnson, S. Kotz, and N. Balakrishnan. *Continuous Multivariate Distributions*, volume 1. New York: John Wiley & Sons, 2002.

[81] L. A. Jones, C. W. Champ, and S. E. Rigdon. The performance of exponentially weighted moving average charts with estimated parameters. *Technometrics*, 43(2):156–167, 2001.

[82] J. Jung, V. Paxson, A. W. Berger, and H. Balakrishnan. Fast portscan detection using sequential hypothesis testing. In *Proceedings of 2004 IEEE Symposium on Security and Privacy*, pages 211–225, 2004.

[83] R. E. Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1):35–45, 1960.

[84] S. Kent. On the trail of intrusions into information systems. *IEEE Spectrum*, 37(12):52–56, 2000.

[85] D. Kifer, S. Ben-David, and J. Gehrke. Detecting change in data streams. In *Proceedings of the Thirtieth International Conference on Very Large Data Bases-Volume 30*, pages 180–191. VLDB Endowment, 2004.

[86] R. Killick and I. A. Eckley. Changepoint: an R package for changepoint analysis. *Lancaster University*, 2011.

[87] R. Killick, P. Fearnhead, and I. A. Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[88] J. T. Kost and M. P. McDermott. Combining dependent p-values. *Statistics & Probability Letters*, 60(2):183–190, 2002.

[89] P. Lafaye de Micheaux. CompQuadForm. `http://cran.r-project.org/web/packages/CompQuadForm/index.html`, March 2011.

[90] T. L. Lai. Sequential analysis: some classical problems and new challenges. *Statistica Sinica*, 11(2):303–350, 2001.

[91] A. Lakhina, M. Crovella, and C. Diot. Mining anomalies using traffic feature distributions. *ACM SIGCOMM Computer Communication Review*, 35(4):217–228, 2005.

[92] M. Lavielle and G. Teyssiere. Detection of multiple change-points in multivariate time series. *Lithuanian Mathematical Journal*, 46(3):287–306, 2006.

[93] Z. Li, J. Zhang, and Z. Wang. Self-starting control chart for simultaneously monitoring process mean and variance. *International Journal of Production Research*, 48(15):4537–4553, 2010.

[94] B. G. Lindsay, R. S. Pilla, and P. Basak. Moment-based approximations of distributions using mixtures: Theory and applications. *Annals of the Institute of Statistical Mathematics*, 52(2):215–230, 2000.

[95] R. C. Littell and J. L. Folks. Asymptotic optimality of Fisher's method of combining independent tests. *Journal of the American Statistical Association*, 66(336):802–806, 1971.

[96] R. C. Littell and J. L. Folks. Asymptotic optimality of Fisher's method of combining independent tests ii. *Journal of the American Statistical Association*, 68(341):193–194, 1973.

[97] G. Lorden. Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, pages 1897–1908, 1971.

[98] C. A. Lowry, W. H. Woodall, C. W. Champ, and S. E. Rigdon. A multivariate exponentially weighted moving average control chart. *Technometrics*, 34(1):46–53, 1992.

[99] J. M. Lucas. The design and use of V-mask control schemes. *Journal of Quality Technology*, 8:1–11, 1976.

[100] J. M. Lucas and M. S. Saccucci. Exponentially weighted moving average control schemes: properties and enhancements. *Technometrics*, pages 1–12, 1990.

[101] E. M. Maboudou-Tchao and D. M. Hawkins. Self-starting multivariate control charts for location and scale. *Journal of Quality Technology*, 43(2):113–126, 2011.

[102] E. M. Maboudou-Tchao and D. M. Hawkins. Detection of multiple change-points in multivariate data. *Journal of Applied Statistics*, 40(9):1979–1995, 2013.

[103] J. F. MacGregor and T. J. Harris. Exponentially weighted moving average control schemes: Properties and enhancements: Discussion. *Technometrics*, 32(1):23–26, 1990.

[104] J. F. MacGregor and T. J. Harris. The exponentially weighted moving variance. *Journal of Quality Technology*, 25(2):106–118, 1993.

[105] J. F. MacGregor and T. H. Kourti. Statistical process control of multivariate processes. *Control Engineering Practice*, 3(3):403–414, 1995.

[106] R. Mahajan, S. M. Bellovin, S. Floyd, J. Ioannidis, V. Paxson, and S. Shenker. Controlling high bandwidth aggregates in the network. *ACM SIGCOMM Computer Communication Review*, 32(3):62–73, 2002.

[107] Y. Mei. Efficient scalable schemes for monitoring a large number of data streams. *Biometrika*, 97(2):419–433, 2010.

[108] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher. *Internet Denial of Service: Attack and Defense Mechanisms*. Prentice Hall, 2005.

[109] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. In *Proceedings of 10th IEEE International Conference on Network Protocols*, pages 312–321, 2002.

[110] D. C. Montgomery. *Introduction to statistical quality control*. John Wiley & Sons, 5th edition, 2005.

[111] D. S. Moore and M. C. Spruill. Unified large-sample theory of general chi-squared statistics for tests of fit. *The Annals of Statistics*, pages 599–616, 1975.

[112] G. V. Moustakides. Optimal stopping times for detecting changes in distributions. *The Annals of Statistics*, 14(4):1379–1387, 1986.

[113] L. S. Nelson. Technical aids. *Journal of Quality Technology*, 16(4):238–239, 1984.

[114] C. Nemeth, P. Fearnhead, and L. Mihaylova. Sequential Monte Carlo methods for state and parameter estimation in abruptly changing environments. *IEEE Transactions on Signal Processing*, 62(5):1245–1255, 2014.

[115] L. O'Callaghan, N. Mishra, A. Meyerson, S. Guha, and R. Motwani. Streaming-data algorithms for high-quality clustering. In *Proceedings of the 18th International Conference on Data Engineering*, pages 685–694. IEEE, 2002.

[116] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[117] M. Paolella. *Intermediate Probability: A Computational Approach*. John Wiley & Sons, 2007.

[118] P. B. Patnaik. The non-central $\chi^2$ and F-distribution and their applications. *Biometrika*, 36(1/2):202–232, 1949.

[119] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer networks*, 31(23):2435–2463, 1999.

[120] T. Pham-Gia and Q. P. Duong. The generalized Beta and F distributions in statistical modelling. *Mathematical and Computer Modelling*, 12(12):1613–1625, 1989.

[121] J. J. Pignatiello and G. C. Runger. Comparisons of multivariate CUSUM charts. *Journal of Quality Technology*, 22(3):173–186, 1990.

[122] A. S. Polunchenko and A. G. Tartakovsky. On optimality of the shiryaev-roberts procedure for detecting a change in distribution. *The Annals of Statistics*, pages 3445–3457, 2010.

[123] M. R. Reynolds, Jr and G.-Y. Cho. Multivariate control charts for monitoring the mean vector and covariance matrix. *Journal of Quality Technology*, 38(3):230–253, 2006.

[124] M. R. Reynolds, Jr and G.-Y. Cho. Multivariate control charts for monitoring the mean vector and covariance matrix with variable sampling intervals. *Sequential Analysis*, 30(1):1–40, 2011.

[125] M. R. Reynolds, Jr and K. Kim. Multivariate control charts for monitoring the process mean and variability using sequential sampling. *Sequential Analysis*, 26(3):283–315, 2007.

[126] M. R. Reynolds, Jr, J. Lou, J. Lee, and S. Wang. The design of GLR control charts for monitoring the process mean and variance. *Journal of Quality Technology*, 45(1):34–60, 2013.

[127] H. Robbins and E. J. G. Pitman. Application of the method of mixtures to quadratic forms in normal variates. *The Annals of Mathematical Statistics*, 20(4):552–560, 1949.

[128] S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1(3):239–250, 1959.

[129] S. W. Roberts. A comparison of some control chart procedures. *Technometrics*, pages 411–430, 1966.

[130] M. Roesch. Snort-lightweight intrusion detection for networks. In *Proceedings of the 13th USENIX conference on System administration*, pages 229–238, 1999.

[131] H. Rolka, H. Burkom, G. F. Cooper, M. Kulldorff, D. Madigan, and W.-K. Wong. Issues in applied statistics for public health bioterrorism surveillance using multiple data streams: research needs. *Statistics in Medicine*, 26(8):1834–1856, 2007.

[132] G. J. Ross and N. M. Adams. Two nonparametric control charts for detecting arbitrary distribution changes. *Journal of Quality Technology*, 44(2):102, 2012.

[133] G. J. Ross, D. K. Tasoulis, and N. M. Adams. Nonparametric monitoring of data streams for changes in location and scale. *Technometrics*, 53(4):379–389, 2011.

[134] S. M. Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. Academic press, 2004.

[135] F. E. Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114, 1946.

[136] J. G Saw, M. C. K. Yang, and T. C. Mo. Chebyshev inequality with estimated mean and variance. *The American Statistician*, 38(2):130–132, 1984.

[137] J. Sheil and I. O'Muircheartaigh. Algorithm AS 106: The distribution of non-negative quadratic forms in normal variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 26(1):92–98, 1977.

[138] W. A. Shewhart. *Economic Control of Quality of Manufactured Product*. D. Van Nostrand Company, Inc., 1931.

[139] A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Probability & Its Applications*, 8(1):22–46, 1963.

[140] H. Solomon and M. A. Stephens. Distribution of a sum of weighted chi-square variables. *Journal of the American Statistical Association*, 72(360):881–885, 1977.

[141] A. Sperotto, M. Mandjes, R. Sadre, P.-T. de Boer, and A. Pras. Autonomic parameter tuning of anomaly-based IDSs: an SSH case study. *IEEE Transactions on Network and Service Management*, 9(2):128–141, 2012.

[142] M. C. Spruill. A comparison of chi-square goodness-of-fit tests based on approximate Bahadur slope. *The Annals of Statistics*, 4(2):409–412, 1976.

[143] M. Staudacher, S. Telser, A. Amann, H. Hinterhuber, and M. Ritsch-Marte. A new method for change-point detection developed for on-line analysis of the heart beat variability during sleep. *Physica A: Statistical Mechanics and its Applications*, 349(3):582–596, 2005.

[144] S. A. Stouffer, E. A. Suchman, L. C. DeVinney, S. A. Star, and R. M. Williams, Jr. *The American Soldier: Adjustment During Army Life, Vol. I. Studies in Social Psychology in World War II.* Princeton University Press, Princeton, 1949.

[145] Z. G. Stoumbos, M. R. Reynolds, Jr, T. P. Ryan, and W. H. Woodall. The state of statistical process control as we proceed into the 21st century. *Journal of the American Statistical Association*, 95(451):992–998, 2000.

[146] J. H. Sullivan. Detection of multiple change points from clustering individual observations. *Journal of Quality Control*, 34(4):371–383, 2002.

[147] J. H. Sullivan and L. A. Jones. A self-starting control chart for multivariate individual observations. *Technometrics*, 44(1):24–33.

[148] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blažek, and H. Kim. Detection of intrusions in information systems by sequential change-point methods. *Statistical Methodology*, 3(3):252–293, 2006.

[149] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing*, 54(9):3372–3382, 2006.

[150] V. G. Tercero-Gómez, A. Cordero-Franco, A. Pérez-Blanco, and A. Hernández-Luna. A self-starting CUSUM chart combined with a maximum likelihood estimator for the time of a detected shift in the process mean. *Quality and Reliability Engineering International*, 30(4):591–599, 2014.

[151] F. Tsung and T. Wang. Adaptive charting techniques: literature review and extensions. In H. Lenz et al., editors, *Frontiers in Statistical Quality Control, 9*, pages 19–35. Springer-Verlag, 2010.

[152] J. V. Uspensky. *Introduction to Mathematical Probability*. McGraw-Hill New York, 1937.

[153] M. De Vivo, E. Carrasco, G. Isern, and G. O. de Vivo. A review of port scanning techniques. *ACM SIGCOMM Computer Communication Review*, 29(2):41–48, 1999.

[154] S. Wang and M. R. Reynolds, Jr. A GLR control chart for monitoring the mean vector of a multivariate normal process. *Journal of Quality Technology*, 45(1):18–33, 2013.

[155] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham. A taxonomy of computer worms. In *Proceedings of the 2003 ACM Workshop on Rapid Malcode*, pages 11–18, 2003.

[156] B. L. Welch. The significance of the difference between two means when the population variances are unequal. *Biometrika*, 29(3/4):350–362, 1938.

[157] M. C. Whitlock. Combining probability from independent tests: the weighted Z-method is superior to Fisher's approach. *Journal of Evolutionary Biology*, 18(5):1368–1373, 2005.

[158] H. Wickham. *ggplot2: elegant graphics for data analysis*. Springer New York, 2009.

[159] A. T. A. Wood. An F approximation to the distribution of a linear combination of chi-squared variables. *Communications in Statistics-Simulation and Computation*, 18(4):1439–1456, 1989.

[160] W. H. Woodall and M. M. Ncube. Multivariate CUSUM quality-control procedures. *Technometrics*, 27(3):285–292, 1985.

[161] H. Wu, B. Salzberg, and D. Zhang. Online event-driven subsequence matching over financial data streams. In *Proceedings of the 2004 ACM SIGMOD international conference on Management of data*, pages 23–34. ACM, 2004.

[162] Y. Xie and D. Siegmund. Sequential multi-sensor change-point detection. *The Annals of Statistics*, 41(2):670–692, 2013.

[163] Y. Xie and D. Sigmund. Sequential multi-sensor change-point detection. *Annals of Statistics*, 31(2):670–692, 2013.

[164] K. D. Zamba and D. M. Hawkins. A multivariate change-point model for statistical process control. *Technometrics*, 48(4):539–549, 2006.

[165] K. D. Zamba and D. M. Hawkins. A multivariate change-point model for change in mean vector and/or covariance structure. *Journal of Quality Technology*, 41(3):285–303, 2009.

[166] J. Zhang, Z. Li, and Z. Wang. A multivariate control chart for simultaneously monitoring process mean and variability. *Computational Statistics & Data Analysis*, 54(10):2244–2252, 2010.

[167] Jin-Ting Zhang and Jianwei Chen. Statistical inferences for functional data. *The Annals of Statistics*, 35(3):1052–1079, 2007.

[168] C. Zou and P. Qiu. Multivariate statistical process control using lasso. *Journal of the American Statistical Association*, 104(488), 2009.