UNIVERSITY OF CALIFORNIA
RIVERSIDE

Sequential Procedures for Nonparametric Statistical Process Control and Longitudinal Data Classification

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Xin Zhang

March 2014

Dissertation Committee:

Professor Jun Li, Chairperson
Professor Daniel R. Jeske
Professor Xing Pan

The Dissertation of Xin Zhang is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to thank Professor Jun Li and Professor Daniel Jeske for their guidance and support, and for the research opportunities they have provided over the last few years. I feel so lucky to have them as my advisor and co-advisor. I learned not only statistical knowledge, but also how to become a better statistician and researcher from both of them.

I am grateful to Dr. Vance Wong and Dr. Brian Noland from Alere Inc., San Diego and Dr. Mazda Marvasti from VMWare Inc., Irvine for granting me access to the Sepsis and network surveillance data used in this dissertation.

The text appearing in Chapter 2 of this dissertation, in part or in full, is a reprint of the material as it appears in *Journal of Nonparametric Statistics* (2013), 25:1-20. The co-author Jun Li listed in that publication directed and supervised the research which forms the basis for this dissertation. The co-author Daniel Jeske listed in that publication contributed innovative idea, general research collaboration, and editorial assistance. This joint work was supported in part by Jun Li's National Science Foundation Grant DMS-0907655.

The text appearing in Chapter 4 of this dissertation, in part or in full, has been submitted as a working paper for publication to *Statistics in Medicine*. The co-author Daniel Jeske listed in that working paper directed and supervised the research which forms the basis for this dissertation. The co-author Jun Li listed in that working paper contributed detailed revision, general research collaboration, and editorial assistance. The co-author Vance Wong listed in that working paper contribute the real data and general research collaboration.

To my parents, Miao Xin and Shaojun Zhang.

ABSTRACT OF THE DISSERTATION

Sequential Procedures for Nonparametric Statistical Process Control and Longitudinal Data Classification

by

Xin Zhang

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, March 2014
Professor Jun Li, Chairperson

Sequential analysis could potentially reduce financial and human cost due to its capability of reaching an earlier conclusion. Since its introduction, sequential analysis has been widely applied to many areas such as statistical process control (SPC) and clinical design. However, nonparametric SPC cumulative sum (CUSUM) procedures for multivariate data and correlated observations are still rare in literatures, and there is little discussion in sequential classification for longitudinal data. In this dissertation we try to develop new sequential procedures for nonparametric statistical process control applicable to multivariate and serially correlated data, and sequential classifier for longitudinal data.

First, we develop two nonparametric multivariate CUSUM control charts based on spatial sign and data depth. These two procedures can be considered as the nonparametric counterparts of the two parametric multivariate CUSUM procedures developed in Crosier (1988). We show that the two proposed CUSUM procedures are affine-invariant and asymptotically distribution-free over a broad family of distributions. In our simulation

studies, the proposed CUSUM procedures perform well across a broad range of settings, and compare favorably with existing CUSUM procedures for detecting location and scale changes.

Second, on the foundation of the above nonparameric multivariate CUSUM control charts, a nonparametric SPC procedure for correlated data is proposed. We incorporate wavelet decomposition with Box and Jenkins time series models and the above multivariate CUSUM control chart to obtain a procedure that is robust under correlated processes without distributional assumption. The procedure is also shown to be powerful in detecting location shift through extensive simulation studies.

Last, we develop a first of its kind sequential classification procedure for longitudinal data. The procedure adapts a neutral zone classifier framework, and attempts to reduce overall cost when the cost of time is considered. The sequential classifier evaluates each subject at each longitudinal time point for evidence of classification. A classification decision is not made until sufficient confidence is present or the last time point where the data can be collected is reached. The early decision property of the proposed classifier may aid the early diagnosis of severe disease diagnosis as in our real data example.

Contents

List of Figures **x**

List of Tables **xii**

1 Introduction **1**

2 Nonparametric Multivariate CUSUM Control Charts **7**
2.1 Introduction 7
2.2 Spatial Sign, Data Depth and Transformation 10
2.2.1 Spatial Sign 10
2.2.2 Data Depth 10
2.2.3 Transformation 11
2.3 Spatial Sign CUSUM Control Chart (SS-CUSUM) 13
2.4 Data Depth CUSUM Control Chart (DD-CUSUM) 15
2.5 Simulation Studies 19
2.5.1 Robustness of SS-CUSUM 20
2.5.2 Location Shift Detection 23
2.5.3 Scale Increase Detection 30

3 Nonparametric CUSUM Control Chart for Autocorrelated Processes **37**
3.1 Introduction 37
3.2 Wavelets 41
3.3 Wavelet-Based SS-CUSUM Control Chart 45
3.4 Simulation Studies 49
3.4.1 ARL_0 Performance Evaluation 52
3.4.2 ARL_1 Performance Evaluation 59
3.4.3 Implementation Issues 62
3.5 Real Data Example 67

4 Sequential Classifier for Longitudinal Data **76**
4.1 Introduction 76
4.2 Sequential Classifier for Longitudinal Data 79

4.2.1 Neutral Zone Classifier 79
4.2.2 Mixed Effects Model Based Logistic Regression 82
4.2.3 Sequential Classification Procedure 88
4.3 Performance Evaluation 89
4.4 Application to Sepsis Study 96

5 Concluding Remarks **99**

References **102**

Appendix A Proofs **108**

Appendix B Additional Power Comparison Results for SS-CUSUM **114**

Appendix C Derivations of eBLUPs for New Subjects **119**

Appendix D Mixed Effects Model Based Logistic Regression Classifier Performance Evaluation **121**
D.1 Motivation 122
D.2 MER Comparison 124
D.2.1 LME Model Based Classification 124
D.2.2 NME Model Based Classification 128

List of Figures

2.1 Power comparision of SS-CUSUM, MCUSUM, AR-CUSUMF (AR-CUSUM with the first anti-rank), and AR-CUSUML (AR-CUSUM with the last anti-rank) with a shift of b in the first component under: (a) $t_{5,3}$; (b) Cauchy$_5$; (c) $\chi^2_{5,1}$; (d) Gamma$_{5,1}$. 27

2.2 Power comparision of SS-CUSUM, MCUSUM, AR-CUSUMF (AR-CUSUM with the first anti-rank), and AR-CUSUML (AR-CUSUM with the last anti-rank) with a downward shift of b in all the components under: (a) Norm$_5$; (b) $t_{5,3}$; (c) Cauchy$_5$; (d) $\chi^2_{5,1}$; (e) Gamma$_{5,1}$. 31

3.1 Real CPU Usage Data: (a) illustrates the CPU usage data, (b) demonstrates the histogram of the data in (a), and (c) represents the autocorrelation function (ACF) of the data in (a) 39

3.2 ACF of Wavelet Coefficients: (a) - (c) represents the ACFs of wavelet coefficients from data in Figure 3.1 (a) at scale 1, 2, and 3, respectively. 44

3.3 ARL$_0$ Performance for Data Generated from Fractional ARIMA (p,0.2,q) Models: (a) Normal innovation, $p = 1, q = 1$, (b) Normal innovation, $p = 3, q = 2$, (c) Poission innovation, $p = 1, q = 1$, and (d) Poisson innovation, $p = 3, q = 2$. The six boxplots, from left to right, represent the proposed method with $l = 4$ and $k = 0.2$, with $l = 4$ and $k = 0.3$, with $l = 5$ and $k = 0.2$, with $l = 5$ and $k = 0.3$, Runger chart, and Rank CUSUM chart, respectively. 55

3.4 ARL$_0$ Performance for Data Generated from Fractional ARIMA (p,0.3,q) Models:(a) Normal innovation, $p = 1, q = 1$, (b) Normal innovation, $p = 3, q = 2$, (c) Poission innovation, $p = 1, q = 1$, and (d) Poisson innovation, $p = 3, q = 2$. The six boxplots, from left to right, represent the proposed method with l=5 and k=0.2, with l=5 and k=0.3, with l=6 and k=0.2, with l=6 and k=0.3, Runger chart, and Rank CUSUM chart, respectively. 56

3.5 Goodness-of-fit of the GLMM Model to the Real Data 59

3.6 ARL$_0$ Performance for Data Generated from GLMM Model: (a) Normal $\rho = 0.9$, (b) Normal $\rho = 0.5$, (c) Poisson $\rho = 0.9$, and (d) Possion $\rho = 0.5$. The six boxplots are in the same order as in Figure 3.3 60

3.7 Multivariate Run Test for Decomposition Scale Determination. 64
3.8 Sample Size Comparison for: (a) Wavelet SSCUSUM Chart with l=4, (b) Wavelet SSCUSUM Chart with l=5, (c) Runger Chart, and (d) Rank CUSUM Chart 66
3.9 Multivariate Run Test for Decomposition Scale Determination on Real Data. 69
3.10 Real CPU Usage Monitoring with (a) Wavelet-Based SS-CUSUM control chart, (b) Runger Control Chart, and (c) Rank CUSUM Control Chart. . . 71
3.11 Real CPU Usage Monitoring with location shift δ, with (a) Wavelet-Based SS-CUSUM control chart and (b) Runger Control Chart 72

4.1 Biomarker measurements of selected 40 patients with Severe Sepsis symptoms. The black lines represent non-severe sepsis group and the grey lines represent severe sepsis group 78
4.2 Typical Sample Path for LME Simulation: V-shape 91
4.3 Cost Comparisons 1: (a) - (c) represents cost comparisons for V-shape profiles; (d) - (f) represents cost comparisons for reverse V-shape profiles. . . . 94
4.4 Cost Comparisons 2: (a) - (c) represents cost comparisons for X-shape profiles; (d) - (f) represents cost comparisons for trigonometric model. 95

D.1 MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for V-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$ 125
D.2 MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for reverse V-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$ . . . 126
D.3 MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for X-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$ 127
D.4 MER Comparison for the NME based Classifier and the Original Observation based Classifier for $\sigma^2 = 4$ 130

List of Tables

2.1 Simulated ARL_0 of SS-CUSUM for different distributions 22
2.2 Power comparison for location shifts: $Norm_5$ 25
2.3 Detection power for DD-CUSUM: scale increases in all components 33
2.4 Power comparison for scale increases in one component 35
2.5 Power comparison for scale increases in one component 36

3.1 ARL_1 Comparison for Fractional ARIMA (d=0.2) 73
3.2 ARL_1 Comparison for Fractional ARIMA (d=0.3) 74
3.3 ARL_1 Comparison for GLMM 75

4.1 Cost Structure of Two Class Neutral Zone Classifier 80
4.2 Cost Structures for Simulations 91
4.3 Average Cost: Sepsis Data 97
4.4 Average Waiting Time: Sepsis Data 98

B.1 Power Comparison for One Direction Location Shift: $Cauchy_5$ 115
B.2 Power Comparison for One Direction Location Shift: $t_{5,3}$ 116
B.3 Power Comparison for One Direction Location Shift: $\chi^2_{5,1}$ 117
B.4 Power Comparison for One Direction Location Shift: $Gamma_{5,1}$ 118

Chapter 1

Introduction

Sequential analysis is a statistical analysis where no pre-determined sample size is assigned and data are evaluated as they are collected. A pre-defined stopping rule is applied to terminate further sampling which may sometimes result in a much earlier conclusion and consequently reduce the financial and human cost. The conception of sequential analysis could be dated back to quality control chart introduced by Shewhart (1931). The formal introduction of sequential analysis appears in Wald's sequential probability ratio test (SPRT) (Wald (1945)). Since its introduction, the sequential analysis and sequential procedures are widely applied to various areas such as Statistical Process Control (SPC) and clinical trials and design.

The online SPC procedures are direct use of the sequential analysis. This procedure is a real-time monitoring tool, and therefore more suitable for real applications compared to offline SPC procedures. Throughout this dissertation we use SPC to refer to online SPC procedure if no confusion is generated. The SPC problem involves evaluating each (batch)

of the observations as they are collected from the process based on a known underlying distribution of the process or an estimated distribution from the observed in-control sample (referred to as the reference sample). The procedure stops when the monitoring statistic goes beyond the control limit (stopping rule). The performance of a SPC procedure can be evaluated by the Average Run Length (ARL) defined as the expected time for the procedure to breach the control limit. We define ARL_0 as the in-control ARL, which is analogous to the type I error in a hypothesis testing, and ARL_1 as the out-of-control ARL, which can be considered as the power in a hypothesis testing. Our task is to seek a procedure which has a smaller ARL_1 given a fixed ARL_0. This procedure would minimize the waste and the problems passing on to the customers when the monitored process has abnormal variations.

As the processes are getting more and more complicated, in many practical situations, multiple measurements are collected to characterize the underlying processes. The correlation between the multiple measurements may cause problems when they are monitored separately. Therefore, a multivariate control chart which can monitor multiple measurements simultaneously is needed. Within different multivariate control charts, multivariate CUSUM charts are popular choice for detecting small and moderate changes in the process. However, most of the existing multivariate CUSUM control charts are relying on multivariate normality assumption, which, in practice, is difficult to justify. To overcome this difficulty, we propose two new nonparametric multivariate CUSUM control charts for location and scale change detection. The proposed control charts use spatial sign and data depth which are proven to be promising nonparametric tools. More precisely, the two control charts are formed based on procedures proposed by Crosier (1988) by replacing the

original data with their spatial sign or data depth related statistics. It could be shown that our proposed procedures are distribution-free (nonparametric) under the elliptical directions family, a broad family of distributions. Our simulation studies show that they are robust even outside the elliptical directions family. This distribution-free property makes our procedures widely applicable to many multivariate SPC problems, especially when the underlying distribution of the data is difficult to justify.

Another challenge for SPC problems is a result of the rapid development of sampling techniques in information technology. The processes can now sample a lot more frequently than before. The resulting larger sample size indeed provides more information, but on the other hand increases the autocorrelation within the observations. In some processes where SPC is applied, the correlation structures are even considered as Long Range Dependent (LRD). The variants of the conventional SPC procedures such as Shewhart, CUSUM, and exponentially weighted moving average (EWMA) could not successfully handle the autocorrelated process. Most of the existing SPC procedures for autocorrelated processes are based on some parametric model to account for the correlation structure within the data. In this dissertation, we develop a nonparametric CUSUM control chart for serially correlated processes. The procedure is based on the decorrelation property of wavelet decomposition. The data are first divided into batches according to the wavelet decomposition level. Then the coefficients from each level of wavelet decomposition are further modeled by Box and Jenkins time series model to extract the approximately uncorrelated residuals, which are grouped appropriately as multivariate vectors based on which batch the residual is from. The multivariate vectors are finally treated as inputs for the nonparametric

multivariate CUSUM control chart developed above. The procedure is shown to be robust under processes generated from different distributions in our simulation studies. The proposed procedure is especially useful when the monitored process possesses long memory. A network surveillance data is monitored using our proposed method to demonstrate its application. The procedure is shown to have well controlled ARL_0 and quick response to an artificially created process location change.

Another application of sequential analysis in this dissertation is to develop sequential classifiers for longitudinal data. Similarly as other sequential procedures, sequential classifiers can lead to earlier classification and could potentially save cost when the cost of time is taken into account. In practice, an earlier classification is highly desirable in many applications. One example would be severe disease diagnosis. However, sequential procedure for longitudinal data classification is still lacking in the literature. In this dissertation, we propose a sequential classifier for longitudinal data which utilize the neutral zone classifier framework. To overcome the commonly encountered difficulties of longitudinal data, such as missing values and irregular sampled data, we also incorporate mixed effects model in our procedure. The classifier utilizes the subject-specific effects estimated from the mixed effects model as input. The classification procedure evaluates each subject sequentially at each longitudinal time point. If there is not adequate confidence in making a classification at a given time point, the decision will wait until the next time point where another measurement is collected. This process continues until there is enough confidence of making a classification or until the last time point where data can be collected is reached. The procedure is shown to be able to reduce cost via extensive simulation studies. As a

demonstration, we apply the procedure to a severe sepsis diagnosis data set. The procedure shows reduction in overall cost and average diagnosis waiting time.

The rest of the dissertation is organized as follows. In Chapter 2, the work on nonparametric multivariate CUSUM control charts is collected. It begins with a background introduction and literature review. Then notations on spatial sign and data depth, and required transformation is introduced. It follows by the details of the two nonparametric multivariate CUSUM control charts. A simulation study is included to evaluate our proposed procedures. In Chapter 3, we present the nonparametric CUSUM control chart for serially correlated processes. After discussing the network surveillance data that motivate this work and the related literatures, we briefly review the preliminary knowledge on wavelets. Then we introduce our proposed procedure. A simulation study to compare our proposed method to the "residual-based" methods is shown next. Some implementation issues are also included. Finally we enclose a network surveillance example to illustrate the application of our proposed method. In Chapter 4, we develop the sequential classifier for longitudinal data. Motivation and literature review are first included, followed by review of neutral zone classifier framework. We next introduce the mixed effects model based logistic regression procedure, which leads to our sequential classification procedure. A performance evaluation demonstrating the overall cost reduction via simulation studies is also included. Moreover, our proposed sequential procedure is applied to the motivating severe sepsis diagnosis example to show overall cost and average waiting time reduction. Some concluding remarks are given in Chapter 5. Appendix A collects all the proofs in Chapter 2 and Appendix B contains some extra simulation results for Chapter 2. Appendix C details

derivation of eBLUPs used in Chapter 4. Appendix D compares the performance between our proposed mixed effects model based logistic regression classifier and the observation based logistic regression classifier based on misclassification error rate (MER).

Chapter 2

Nonparametric Multivariate CUSUM Control Charts

2.1 Introduction

A typical setup for multivariate control charts is the following. There are m independent and identically distributed historical (reference) data for the p monitored characteristics, denoted by $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_m \in \Re^p$, from the in-control process. Let F_0 be the underlying distribution of $\boldsymbol{Y}_i$, also referred to as the in-control distribution. Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots$ be future observations of the process, under the distribution F_1. The task of multivariate control charts is to determine if F_1 is the same as F_0 and if not, to signal when F_1 changes from F_0 as early as possible.

There are many existing multivariate control charts in the literature. We refer to Bersimis, Psarakis and Panretos (2007) for an overview on this topic. The multivari-

ate control charts we are particularly interested in is multivariate CUSUM control charts since they are the popular choice for detecting small and moderate changes in F_1 from F_0. Crosier (1988) proposed two multivariate CUSUM procedures for detecting location shifts in F_1 from F_0 by assuming that F_0 and F_1 are both multivariate normal distributions. One of the proposed procedures is the univariate CUSUM procedure for monitoring the T statistic, which is the square root of Hotelling's T^2 statistic, therefore it is referred to as COT (CUSUM of T) procedure. Another one Crosier proposed is called MCUSUM procedure which monitors the cumulative sum of $\boldsymbol{X}_i$ directly. Based on the simulations in his paper, the MCUSUM procedure performs better than the COT procedure for detecting location shifts. Both of the two CUSUM procedures in Crosier's paper were developed under the multivariate normality assumption. However, this assumption may not hold in many situations. Therefore, nonparametric multivariate CUSUM procedures are more desirable.

Qiu and Hawkins (2001, 2003) developed a nonparametric multivariate CUSUM procedure for detecting location shifts based on the anti-ranks of the p components in $\boldsymbol{X}_i$. However, the method is not distribution-free since it depends on the in-control distributions of the anti-ranks. The method also only uses subset of the anti-ranks of the p components, which may lead to loss of power for detecting location changes. Furthermore, how to choose the subset of anti-ranks is not clear especially when no information about the possible location shift is available. In this chapter, we propose a new simple nonparametric CUSUM procedure for detecting location changes, which is based on spatial signs of the $\boldsymbol{X}_i$ and can be considered as the nonparametric counterpart of Crosier's MCUSUM procedure. This procedure is shown to be asymptotically distribution-free for a broad family of distributions,

and our simulation studies show this procedure is very robust even for distributions outside of this family. More importantly, our CUSUM procedure is shown to be more powerful than the anti-rank CUSUM procedure for detecting location shifts in a variety of simulation settings.

So far, most of the existing nonparametric CUSUM procedures were developed for detecting location changes. In practice, scale changes of the measurements may also indicate abnormal variations of the process, and therefore need to be detected as well. A number of papers have taken on this problem under the normality assumption, for example, Chan and Zhang (2001), Reynolds and Cho (2006), Reynolds and Stoumbos (2008), Hawkins and Maboudou-Tchao (2008), Yen and Shiau (2010). However, little progress has been made on nonparametric CUSUM procedures for detecting scale changes. Therefore, the second objective of this chapter is to develop a nonparametric CUSUM procedure for detecting scale changes. In many applications, scale increases indicate increases in variability, and therefore are of more concern than scale decreases. Thus, we focus our nonparametric procedure on detecting scale increases. The CUSUM procedure we propose is based on the so-called data depth and can be considered as a nonparametric version of the aforementioned Crosier's COT procedure. Crosier's COT procedure was originally proposed for detecting location shifts, but was later found to be more powerful for detecting scale increases (Hawkins and Olwell, 1998).

The rest of the chapter is organized as follows. In Section 2.2, we review the background materials, including spatial sign, data depth and transformation needed in our proposed CUSUM procedures. In Section 2.3, we propose the nonparametric CUSUM pro-

cedure based on spatial sign for detecting location shifts. In Section 2.4, we introduce another nonparametric CUSUM scheme based on data depth for detecting scale increases. We present some simulation studies in Section 2.5 to evaluate the performance of our proposed CUSUM procedures.

2.2 Spatial Sign, Data Depth and Transformation

2.2.1 Spatial Sign

In the univariate case, the sign of a number x has three values, namely -1, 0, and 1, for $x < 0, x = 0,$ and $x > 0$. The sign of non-zero x can be also calculated by $U(x) = x/|x|$. We can extend this definition to the multivariate case, and obtain spatial sign of any non-zero multivariate observation by $U(\boldsymbol{x}) = \boldsymbol{x}/\|\boldsymbol{x}\|$, where $\boldsymbol{x}$ is any non-zero p-dimensional vector, and $\|\cdot\|$ denotes the Euclidean norm. Based on the definition, the spatial sign of a non-zero multivariate observation is a vector of unit length pointing to the same direction as the observation. Therefore, it can be considered as the direction vector of the multivariate observation.

2.2.2 Data Depth

Data depth is a measure of centrality of a given point with respect to a multivariate data cloud or its underlying distribution. There are many notions of data depth in the literature, for example, the half-space depth introduced by Tukey (1975), simplicial depth proposed by Liu (1990), and projection depth used in Stahel (1981), Donoho (1982), Donoho and Gasko (1992), and Zuo (2003). To see a more complete list of different notions of data

depth, see Liu et al. (1999) and Zuo and Serfling (2000). In this chapter, we use spatial depth in our data depth based CUSUM procedure due to the fact that the spatial depth is much easier to compute for higher dimensional data than other depths. The definition of spatial depth is given below.

Definition 2.1. The *spatial depth* (Chaudhuri (1996), Vardi and Zhang (2000), and Serfling (2002)) at $\boldsymbol{x}$ with respect to F is defined as

$$SPD_F(\boldsymbol{x}) = 1 - \left\| E_F \left\{ \frac{\boldsymbol{x} - \boldsymbol{Y}}{\|\boldsymbol{x} - \boldsymbol{Y}\|} \right\} \right\|, \quad \text{where } \boldsymbol{Y} \sim F.$$

Based on the above definition of spatial sign, the spatial depth can be written as

$$SPD_F(\boldsymbol{x}) = 1 - \left\| E_F \left\{ U(\boldsymbol{x} - \boldsymbol{Y}) \right\} \right\|, \quad \text{where } \boldsymbol{Y} \sim F.$$

When the observation $\boldsymbol{x}$ is near the center of the distribution F, $E_F \{U(\boldsymbol{x} - \boldsymbol{Y})\}$ would be very close to $\mathbf{0}$, and therefore $SPD_F(\boldsymbol{x})$ would attain its maximum value 1. On the other hand, if $\boldsymbol{x}$ is relatively near the outskirts, $SPD_F(\boldsymbol{x})$ would approach its minimum value 0. Therefore, the spatial depth provides a reasonable measure of "depth" of $\boldsymbol{x}$ with respect to the distribution F.

Given a sample $\{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_m\}$ from F, the sample spatial depth is defined as

$$SPD_{F_m}(\boldsymbol{x}) = 1 - \left\| \frac{1}{m} \sum_{i=1}^{m} U(\boldsymbol{x} - \boldsymbol{Y}_i) \right\|,$$

where F_m represents the empirical distribution of $\{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_m\}$.

2.2.3 Transformation

Our proposed CUSUM procedures in the next two sections, if directly based on the above spatial sign and spatial depth, are only invariant under rotation or when the

same scale transformation is done on all components. They are not invariant under general affine transformations of the data. The affine invariance is sometimes a desired property for a CUSUM procedure, and is also a key property in order to obtain a distribution-free procedure in Section 2.3.

In the literature, many procedures based on spatial sign and spatial depth can achieve affine invariance through transformation-retransformation procedure (Chakraborty, Chaudhuri, and Oja (1998)). The idea behind the transformation-retransformation procedure is making some appropriate transformation on the data first and then applying the procedure to the transformed data. We will adopt this transformation-retransformation approach. In particular, the transformation we will use on our data is motivated by Hettmansperger and Randles (2002). Recall that $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ is the reference sample, and $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots\}$ is the new observations from the process. The transformation we will use on $\boldsymbol{Y}_i$ and $\boldsymbol{X}_i$ is,

$$\boldsymbol{Y}_i^* = \hat{A}_m(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m), \quad \boldsymbol{X}_i^* = \hat{A}_m(\boldsymbol{X}_i - \hat{\boldsymbol{\theta}}_m), \tag{2.1}$$

where $(\hat{\boldsymbol{\theta}}_m, \hat{A}_m)$ is the solution to the following equations,

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{A}_m(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m)}{\|\hat{A}_m(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m)\|}\right) = \mathbf{0}, \tag{2.2}$$

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{A}_m(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m)(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m)'\hat{A}_m'}{\|\hat{A}_m(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_m)\|^2}\right) = \frac{1}{p}I_p. \tag{2.3}$$

and $\hat{A}_m$ is the upper triangular $p \times p$ matrix with positive diagonal elements and a 1 in the upper-left element. We follow the iterative algorithm developed in Hettmansperger and Randles (2002) to find the solution $(\hat{\boldsymbol{\theta}}_m, \hat{A}_m)$ to the above two equations. In both of our proposed CUSUM procedures in the following sections, we first transform $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ and

$\{\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots\}$ as in (2.1), and then work with the transformed data $\boldsymbol{Y}_i^*$ and $\boldsymbol{X}_i^*$.

2.3 Spatial Sign CUSUM Control Chart (SS-CUSUM)

Crosier (1988) introduced the MCUSUM procedure, which has been shown to work well for detecting small location shifts for multivariate normal distribution. The MCUSUM procedure is described as follows.

Define

$$C_n = [(\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_0)' \hat{\Sigma}_0^{-1} (\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_0)]^{1/2}$$

$$\boldsymbol{S}_n = \begin{cases} \boldsymbol{0} & \text{if } C_n \le k \\ (\boldsymbol{S}_{n-1} + \boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_0)(1 - k/C_n) & \text{if } C_n > k \end{cases}$$

where $k > 0$, $\boldsymbol{S}_0 = \boldsymbol{0}$ and $\hat{\boldsymbol{\mu}}_0$ and $\hat{\Sigma}_0$ are the sample mean and sample covariance matrix from the multinormal reference sample $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$. Let $L_n = [\boldsymbol{S}_n' \hat{\Sigma}_0^{-1} \boldsymbol{S}_n]^{1/2}$, and the system triggers an alarm when $L_n > h$, where h is the control limit.

To develop a nonparametric version of the above MCUSUM procedure, we first notice that the nonparametric sign test in the univariate case was obtained by replacing the original observations by their signs. Therefore, we can follow this idea and replace $\boldsymbol{X}_n$ in the above MCUSUM procedure by its spatial sign. Again to achieve affine invariance, we build our procedure on the transformed data $\{\boldsymbol{Y}_1^*, ..., \boldsymbol{Y}_m^*\}$ and $\{\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, \cdots\}$. Therefore, $\boldsymbol{X}_n$ in the MCUSUM is replaced by $\boldsymbol{U}_n = U(\boldsymbol{X}_n^*)$, the spatial sign of $\boldsymbol{X}_n^*$, $\hat{\boldsymbol{\mu}}_0$ and $\hat{\Sigma}_0$ are replaced by the sample mean and sample covariance matrix of the spatial signs of the transformed

reference sample $\{\boldsymbol{Y}_1^*, ..., \boldsymbol{Y}_m^*\}$. Based on (2.2) and (2.3), it is not difficult to see that the sample mean and sample covariance matrix of the spatial signs of $\{\boldsymbol{Y}_1^*, ..., \boldsymbol{Y}_m^*\}$ are $\mathbf{0}$ and I_p/p, respectively. Therefore, our proposed nonparametric CUSUM procedure is as follows. Define

$$C_n = [(\boldsymbol{S_{n-1}} + \boldsymbol{U_n})^{'}(\boldsymbol{S_{n-1}} + \boldsymbol{U_n})]^{1/2}$$

$$\boldsymbol{S}_n = \begin{cases} \mathbf{0} & \text{if } C_n \le k \\ \\ \boldsymbol{S}_n = (\boldsymbol{S_{n-1}} + \boldsymbol{U_n})(1 - k/C_n) & \text{if } C_n > k \end{cases}$$

where $k > 0$, and $\boldsymbol{S}_0 = \mathbf{0}$. Let $L_n = (\boldsymbol{S}_n^{'}\boldsymbol{S}_n)^{1/2}$, and the procedure triggers an alarm when $L_n > h$, where h is the control limit predetermined by k and the desired in-control average run length (denoted by ARL_0). We call this procedure Spatial Sign CUSUM (SS-CUSUM).

To illustrate the properties of our SS-CUSUM procedure, we first introduce different distributional assumptions for the underlying population in the nonparametric multivariate data analysis literature. Let $\{\boldsymbol{Z}_i\}$ be the independent and identically distributed random sample from F. The distribution F is said to belong to the family of elliptical symmetric distributions if $\boldsymbol{Z}_i = r_i D\boldsymbol{u}_i + \boldsymbol{\mu}$, where D is a fixed $p \times p$ nonsingular matrix, $\boldsymbol{\mu}$ is a fixed p-dimensional vector, the $\boldsymbol{u}_i$ are independent and identically uniformly distributed on the unit p sphere, and the r_i are independent and identically distributed positive scalars, independent of the $\boldsymbol{u}_i$. A weaker assumption than the elliptical symmetric family is the elliptical directions family, which was first introduced by Randles (1989). In the elliptical directions family, the above r_i are only assumed to be positive values, not necessarily

random or independent and identically distributed or independent of the $\boldsymbol{u}_i$. This family includes certain skewed distributions in addition to the elliptically symmetric family.

The following results show the properties of our proposed SS-CUSUM procedure.

Proposition 1 *The SS-CUSUM procedure is affine-invariant.*

Proposition 2 *The SS-CUSUM procedure is asymptotically distribution-free for the distributions in the elliptical directions family.*

Based on the above results, determining the control limit h in our SS-CUSUM procedure can be achieved by simulating data from standard multivariate normal distribution and finding h to obtain the desired ARL_0 for any given k. Although the SS-CUSUM is shown to be asymptotically distribution-free for the elliptical directions family, our simulation studies in Section 2.5 show that this SS-CUSUM procedure is also very robust for distributions outside of the elliptical directions family. By investigating the value of $\boldsymbol{S}_n$ at the alarm point, SS-CUSUM can also provide us with information of the direction of the location shift.

2.4 Data Depth CUSUM Control Chart (DD-CUSUM)

We first review the COT procedure proposed in Croiser (1988). The COT procedure is given by

$$S_n = \max(0, S_{n-1} + T_n - k),$$

where $S_0 \geq 0$, $k > 0$, T_n is the square root of Hotelling T^2 statistic

$$T_n^2 = (\boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_0)' \hat{\Sigma}_0^{-1} (\boldsymbol{X}_n - \hat{\boldsymbol{\mu}}_0),$$

and $\hat{\boldsymbol{\mu}}_0$ and $\hat{\Sigma}_0$ are the sample estimates of $\boldsymbol{\mu}_0$ and Σ_0 for the multinormal reference sample $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$. The procedure triggers an alarm when $S_n > h$, where h is the control limit. Because T_n is more sensitive to the changes in the covariance matrix from Σ_0 to $a\Sigma_0$ $(a > 1)$ than the changes in $\boldsymbol{\mu}_0$, the COT procedure is more powerful for detecting scale increases than it is for detecting location changes (Hawkins and Olwell, 1998).

To develop a nonparametric counterpart of the above COT procedure, we first review the depth-based R statistic introduced by Liu and Singh (1993). Let $D(\cdot)$ denote any valid notion of depth. For any given $\boldsymbol{x} \in \Re^p$, the R statistic is defined as

$$R_F(\boldsymbol{x}) = P\{D_F(\boldsymbol{Y}) \leq D_F(\boldsymbol{x})\},$$

where $\boldsymbol{Y}$ is a random vector drawn from F. If F is not known and a sample $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ is given, the sample version of R statistic is defined by:

$$R_{F_m}(\boldsymbol{x}) = \#\{\boldsymbol{Y}_j \mid D_{F_m}(\boldsymbol{Y}_j) \leq D_{F_m}(\boldsymbol{x}), j = 1, \cdots, m\}/m.$$

In Liu (1995), the statistic R_{F_m} above is used to construct several nonparametric multivariate control charts. In this chapter, we propose a CUSUM control chart based on the R statistic. For this purpose, we calculate $R_{F_m}(\boldsymbol{X}_n)$ of the new observation $\boldsymbol{X}_n$ with respect to the reference sample $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ by using spatial depth. According to the definition of spatial depth, if $\boldsymbol{X}_n$ is near the center of the reference sample, there are many $\boldsymbol{Y}_j$ having smaller depth than $\boldsymbol{X}_n$, and therefore $R_{F_m}(\boldsymbol{X}_n)$ will be large. On the other hand, if $\boldsymbol{X}_n$ is near the outskirts of the reference sample, there are a few $\boldsymbol{Y}_j$ having smaller depth than $\boldsymbol{X}_n$, and therefore $R_{F_m}(\boldsymbol{X}_n)$ will be small. Hence, $1 - R_{F_m}(\boldsymbol{X}_n)$ can be used to quantify the relative distance between $\boldsymbol{X}_n$ and the center of the reference sample. Recall

that, in the COT procedure, T_n can be considered as a standardized distance between $\boldsymbol{X}_n$ and the center of the reference sample. This motivates us to replace T_n by $1 - R_{F_m}(\boldsymbol{X}_n)$ in the COT procedure and obtain the following CUSUM procedure:

$$S_n = \max(0, S_{n-1} + (1 - R_{F_m}(\boldsymbol{X}_n)) - k),$$

where $S_0 = 0$ and $k > 0$. The procedure triggers an alarm when $S_n > h$, where h is the control limit depending on the choice of k and the desired ARL_0.

Liu and Singh (1993) show the following two important properties of the R statistic.

(i) If $\boldsymbol{X} \sim F$, and $D_F(\boldsymbol{X})$ has a continuous distribution, then $R_F(\boldsymbol{X}) \sim \text{Uniform}[0, 1]$, where Uniform$[0, 1]$ denotes a uniform distribution supported in [0,1].

(ii) If $\boldsymbol{X} \sim F$, as $m \to \infty$, $R_{F_m}(\boldsymbol{X}) \xrightarrow{\mathcal{L}} \text{Uniform}[0, 1]$ along almost all $\{\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_m\}$ sequences, provided that $D_{F_m}(\cdot)$ coverges to $D_F(\cdot)$ uniformly as $m \to \infty$. The notation $\xrightarrow{\mathcal{L}}$ stands for covergence in law.

If F is continuous, the spatial depth $SPD_F(\cdot)$ is a continuous function, and $SPD_{F_m}(\cdot)$ converges uniformly to $SPD_F(\cdot)$ (Kolchinskii (1997) and Serfling (2002)). Therefore, the above two properties apply to the spatial depth, which implies that $1 - R_{F_m}(\boldsymbol{X}_n)$ asymptotically follows Uniform$[0, 1]$ when the process is in control. Since the mean of Uniform$[0, 1]$ is 0.5, we further modify our CUSUM procedure as

$$S_n = \max(0, S_{n-1} + (0.5 - R_{F_m}(\boldsymbol{X}_n)) - k).$$

Here $0.5 - R_{F_m}(\boldsymbol{X}_n) = (1 - R_{F_m}(\boldsymbol{X}_n)) - 0.5$. To attain affine invariance, we apply the above CUSUM procedure to the transformed data $\{\boldsymbol{Y}_1^*, ..., \boldsymbol{Y}_m^*\}$ and $\{\boldsymbol{X}_1^*, \boldsymbol{X}_2^*, ...\}$ in (2.1),

and obtain

$$S_n = \max(0, S_{n-1} + (0.5 - R_{F_m^*}(\boldsymbol{X}_n^*)) - k),$$

where $R_{F_m^*}(\boldsymbol{X}_n^*)$ is the sample R statistic of $\boldsymbol{X}_n^*$ with respect to the transformed reference sample $\{\boldsymbol{Y}_1^*, ..., \boldsymbol{Y}_m^*\}$ based on spatial depth. Again, the procedure triggers an alarm when $S_n > h$, where h is the control limit depending on the choice of k and the desired ARL_0. We call this procedure Data Depth CUSUM (DD-CUSUM).

The following results show the properties of our proposed DD-CUSUM procedure.

Proposition 3 *The DD-CUSUM procedure is affine-invariant.*

Proposition 4 *The DD-CUSUM procedure is asymptotically distribution-free for any continuous multivariate distributions.*

From the proof of Proposition 4, $R_{F_m^*}(\boldsymbol{X}_n^*)$ asymptotically follows Uniform$[0, 1]$ when the process is in control. Therefore, determining the control limit h in our DD-CUSUM procedure can be achieved by simulating data from Uniform$[0, 1]$ and finding h to obtain the desired ARL_0 for any given k. Similar to the COT procedure, the DD-CUSUM procedure is capable of detecting many types of distributional changes including location shifts. However, it is more sensitive to scale increases than location shifts. For detecting location shifts, the SS-CUSUM procedure we propose in the previous section is more powerful than the DD-CUSUM procedure. Therefore, in practice, we recommend using both the SS-CUSUM procedure and DD-CUSUM procedure, with SS-CUSUM for location shifts and DD-CUSUM for scale increases.

2.5 Simulation Studies

In this section, we present some simulation studies to evaluate the performance of our proposed SS-CUSUM and DD-CUSUM procedures. In particular, we will compare them with the MCUSUM procedure proposed by Crosier (1988) and the anti-rank CUSUM (AR-CUSUM) procedure proposed by Qiu and Hawkins (2003). In all the simulation studies, we set the nominal ARL_0 as 200 and the reference sample size as 50000, sufficiently large so that our asymptotic results approximately hold. All the simulation results are based on 10000 replicates. The control limits of different CUSUM procedures to achieve the nominal ARL_0 are all determined through simulation using a bi-section search. The algorithm runs as follows:

Step 1. For any control limit h, we obtain its corresponding in-control average run length (denoted by ARL_0^h) by simulating 10000 in-control sample paths and averaging out the run lengths from these 10000 sample paths. Based on this approach, we first find h_1 such that $\text{ARL}_0^{h_1} < \text{ARL}_0$, and h_2 such that $\text{ARL}_0^{h_2} > \text{ARL}_0$.

Step 2. Find $\text{ARL}_0^{h_3}$ where h_3 is the midpoint of h_1 and h_2.

Step 3. If $\text{ARL}_0^{h_3} < \text{ARL}_0$, assign $h_1 = h_3$. If $\text{ARL}_0^{h_3} > \text{ARL}_0$, assign $h_2 = h_3$;

Step 4. Repeat Steps 2 and 3 until $\text{ARL}_0^{h_3}$ is sufficiently close to ARL_0;

Step 5. Use h_3 as the control limit.

For MCUSUM, the control limit for monitoring a p-dimensional random vector is determined by simulating sample paths from standard p-dimensional multivariate normal

distribution, since MCUSUM is affine invariant and is based on normality assumption. For AR-CUSUM, the control limit depends on the in-control distribution. In our simulation, as in practice, the reference sample is used to estimate the in-control distribution, which in turn, is used to compute the control limit. Therefore, the control limit for AR-CUSUM has to be simulated from each distribution separately. For SS-CUSUM, since it is affine invariant and is asymptotically distribution-free for the elliptical directions family, the control limit of SS-CUSUM for monitoring any p-dimension random vector is determined by simulating sample paths from standard p-dimensional multivariate normal distribution. For DD-CUSUM, since the statistic $R_{F_m^*}(\boldsymbol{X}_n^*)$ we use has an asymptotic Uniform$[0,1]$ distribution and the DD-CUSUM procedure is asymptotically distribution-free, the control limit of DD-CUSUM for monitoring any dimensional random vector from any continuous distribution can be determined by simulating sample paths from Uniform$[0,1]$. In summary, the distribution-free properties of our DD-CUSUM and SS-CUSUM make computing their control limits much simpler compared to AR-CUSUM.

2.5.1 Robustness of SS-CUSUM

As stated in Section 2.3, our proposed SS-CUSUM procedure is asymptotically distribution-free for distributions in the elliptical directions family. In the following simulation study, we will investigate the ARL_0 performance of the SS-CUSUM procedure for different distributions, with particular attention to distributions outside of the elliptical directions family. The distributions we consider are similar to those considered in Zou and Tsung (2011), and they are: (i) p-dimensional standard multivariate normal distribution

(denoted by Norm_p); (ii) p-dimensional standard multivariate t distribution with degrees of freedom 3 (denoted by $t_{p,3}$); (iii) p-dimensional multivariate distribution with independent marginal Cauchy distributions (Cauchy_p); (iv) p-dimensional multivariate distribution with independent marginal chi-square distributions, the degree of freedom of each marginal being 1 (denoted by $\chi^2_{p,1}$); and (v) p-dimensional multivariate gamma distribution with shape parameter γ and scale parameter 1 (denoted by $\text{Gamma}_{p,\gamma}$). Details for generating multivariate gamma random vectors can be found in Stoumbos and Sullivan (2002).

Table 1 shows the simulated ARL_0 of our proposed SS-CUSUM along with their corresponding standard errors (in the parentheses) under different distributions. In the table, the first two columns are corresponding to multivariate normal and t distributions which belong to the elliptical directions family. As expected, all the simulated ARL_0 are very close to the nominal level. The last six columns in the table are corresponding to distributions outside of the elliptical directions family. As we can see from the table, except for the two extremely skewed distributions, $\chi^2_{p,1}$ and $\text{Gamma}_{p,1}$, the other four distributions have simulated ARL_0 close to the nominal level with k as large as 0.5. For the two extremely skewed distributions, when the dimension is not high, the ARL_0 are still close to the nominal one. When the dimension gets higher and the value of k gets larger, there is some deviation from the nominal ARL_0 level. However, if we use $k \leq 0.3$, the ARL_0 values are still well controlled near the nominal level. From the above simulation, we can see that, although the SS-CUSUM procedure is shown to be asymptotically distribution-free for the elliptical directions family, it can achieve reasonable ARL_0 for a variety of distributions outside of the elliptical directions family if we choose relatively small k.

Table 2.1: Simulated ARL_0 of SS-CUSUM for different distributions

k	Norm_2	$t_{2,3}$	Cauchy_2	$\chi^2_{2,1}$	$\text{Gamma}_{2,1}$	$\text{Gamma}_{2,5}$	$\text{Gamma}_{2,10}$	$\text{Gamma}_{2,30}$
0.1	199(1.73)	199(1.75)	199(1.72)	196(1.71)	202(1.75)	200(1.77)	202(1.77)	198(1.73)
0.2	206(1.88)	201(1.85)	199(1.82)	200(1.86)	203(1.86)	201(1.88)	199(1.86)	199(1.83)
0.3	200(1.89)	200(1.89)	203(1.95)	200(1.89)	198(1.89)	197(1.87)	200(1.92)	200(1.90)
0.4	202(1.95)	198(1.93)	201(1.92)	200(1.94)	201(1.90)	204(1.95)	201(1.91)	201(1.92)
0.5	202(1.97)	198(1.92)	199(1.94)	193(1.89)	198(1.94)	201(1.91)	199(1.90)	200(1.97)
k	Norm_5	$t_{5,3}$	Cauchy_5	$\chi^2_{5,1}$	$\text{Gamma}_{5,1}$	$\text{Gamma}_{5,5}$	$\text{Gamma}_{5,10}$	$\text{Gamma}_{5,30}$
0.1	201(1.66)	200(1.62)	198(1.61)	200(1.66)	199(1.64)	200(1.66)	199(1.64)	200(1.64)
0.2	201(1.83)	202(1.83)	202(1.83)	202(1.56)	199(1.79)	199(1.83)	198(1.81)	196(1.80)
0.3	200(1.88)	200(1.92)	201(1.90)	195(1.84)	198(1.88)	198(1.87)	196(1.84)	200(1.91)
0.4	200(1.94)	198(1.92)	201(1.92)	184(1.77)	195(1.88)	197(1.89)	198(1.89)	197(1.90)
0.5	199(1.92)	199(1.93)	196(1.85)	169(1.62)	184(1.79)	192(1.87)	200(1.95)	198(1.91)
k	Norm_{10}	$t_{10,3}$	Cauchy_{10}	$\chi^2_{10,1}$	$\text{Gamma}_{10,1}$	$\text{Gamma}_{10,5}$	$\text{Gamma}_{10,10}$	$\text{Gamma}_{10,30}$
0.1	198(1.52)	203(1.60)	201(1.55)	202(1.56)	202(1.55)	200(1.56)	199(1.55)	195(1.51)
0.2	198(1.75)	203(1.85)	199(1.77)	202(1.79)	201(1.81)	207(1.90)	198(1.78)	198(1.78)
0.3	198(1.89)	199(1.88)	202(1.91)	185(1.70)	192(1.79)	201(1.92)	199(1.83)	201(1.89)
0.4	201(1.93)	194(1.90)	203(1.97)	174(1.69)	183(1.74)	194(1.86)	198(1.92)	202(1.94)
0.5	197(1.90)	198(1.88)	190(1.85)	152(1.48)	166(1.60)	189(1.83)	191(1.87)	194(1.88)

2.5.2 Location Shift Detection

As mentioned earlier, our proposed DD-CUSUM is more sensitive to scale increases than location shifts. In our simulation studies, it has been confirmed that SS-CUSUM is more powerful than DD-CUSUM for detecting location shifts. Therefore, we recommend using SS-CUSUM for detecting location shifts in practice. In this section, we present a simulation study for comparing the detection power of SS-CUSUM with two existing methods, MCUSUM and AR-CUSUM, for detecting location shifts under different distribution settings. The distributions we consider are Norm_5, $t_{5,3}$, Cauchy_5, $\chi^2_{5,1}$ and $\text{Gamma}_{5,1}$. The distribution notations are the same as the ones in the previous section. The first type of location shifts we consider is the location shift in the first component with size of b varying from 0 to 3 with an increment of 0.5. For AR-CUSUM, we have to choose which subset of the anti-ranks to use. For computational simplicity, we use single anti-rank, and the single anti-rank we consider is the first anti-rank and last anti-rank, which is recommended by Qiu and Hawkins (2003). To make our simulations close to the situations in practice, for each setting we introduce the location shift after 50 observations. It is possible that some of the CUSUM procedures will trigger an alarm (false alarm) before the 50th observation in some of the sample paths. To eliminate the effect of those false alarms on our power evaluation, the sample paths in which any of the CUSUM procedures triggers an alarm before the 50th observation are discarded. Therefore, in the simulated sample paths we consider here all the CUSUM procedures signal after the 50th observation. The detection power of different CUSUM procedures is then compared by the average signal time after the 50th observation. This average signal time is sometimes called steady-state ARL (SSARL) in the literature

(see, for example, Hawkins and Olwell (1998)). Table 2.2 lists the simulated SSARL of each procedure from 10,000 replications under the Norm_5 distribution. It also includes the simulated ARL_0. Since every CUSUM procedure involves the tuning parameter k, Table 2.2 lists the results with different choices of k for each procedure. From the table, we can see that all the four CUSUM procedures can achieve the desired ARL_0. By comparing the four procedures, it is obvious that SS-CUSUM and MCUSUM both outperform the two AR-CUSUM procedures. MCUSUM performs the best as we expected, since MCUSUM is designed specifically for the multinormal distribution. However, SS-CUSUM, the non-parametric counterpart, performs almost equally well as MCUSUM in this multinormal case.

We also obtain similar tables as Table 2.2 which contain the SSARL performance of the four CUSUM procedures under other distributions. One can see Appendix B for the complete results. We here only present the SSARLs of SS-CUSUM with $k = 0.2$ and 0.3, the SSARLs of MCUSUM with $k = 0.2$ and the SSARLs of AR-CUSUM with $k = 0.1$, since those particular choices of k for each procedure give the procedure the best or nearly the best detection power across different simulation settings.

Table 2.2: Power comparison for location shifts: Norm_5

$p = 5$	SS-CUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	203.5(1.67)	201.1(1.85)	203.4(1.92)	194.5(1.88)	200.4(1.98)
0.5	34.1(0.19)	33.5(0.23)	39.7(0.32)	48.5(0.43)	58.2(0.54)
1.0	17.4(0.07)	14.2(0.06)	13.5(0.07)	14.3(0.09)	16.1(0.11)
1.5	12.6(0.05)	9.7(0.03)	8.7(0.03)	8.2(0.03)	8.3(0.04)
2.0	10.4(0.04)	7.9(0.03)	6.8(0.02)	6.3(0.02)	6.0(0.02)
2.5	9.2(0.03)	6.9(0.02)	6.0(0.02)	5.4(0.01)	5.0(0.01)
3.0	8.6(0.03)	6.5(0.02)	5.5(0.01)	4.9(0.01)	4.6(0.01)
$p = 5$	MCUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	203.0(1.44)	202.6(1.64)	199.3(1.73)	206.1(1.88)	204.9(1.91)
0.5	36.6(0.18)	31.5(0.17)	29.4(0.18)	29.8(0.19)	30.6(0.22)
1.0	19.2(0.08)	15.5(0.07)	13.5(0.06)	12.6(0.06)	11.9(0.06)
1.5	13.1(0.05)	10.3(0.04)	8.8(0.03)	8.0(0.03)	7.3(0.03)
2.0	9.9(0.04)	7.9(0.03)	6.6(0.02)	6.0(0.02)	5.4(0.02)
2.5	8.0(0.03)	6.3(0.02)	5.3(0.02)	4.7(0.02)	4.3(0.01)
3.0	6.8(0.02)	5.3(0.02)	4.5(0.02)	4.0(0.01)	3.6(0.01)
$p = 5$	AR-CUSUM First				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	199.7(2.63)	200.1(2.40)	200.3(2.27)	197.3(2.09)	201.3(2.15)
0.5	106.8(0.96)	129.3(1.30)	134.2(1.34)	131.4(1.36)	131.5(1.30)
1.0	51.7(0.34)	72.5(0.63)	96.5(0.97)	97.6(1.00)	89.4(1.05)
1.5	39.1(0.22)	50.4(0.37)	75.0(0.74)	86.0(0.88)	89.5(0.93)
2.0	35.4(0.19)	43.4(0.28)	65.3(0.59)	80.1(0.84)	84.3(0.88)
2.5	34.5(0.18)	42.1(0.27)	62.0(0.57)	78.4(0.82)	82.4(0.84)
3.0	34.1(0.17)	41.9(0.26)	61.2(0.54)	77.0(0.80)	81.7(0.83)
$p = 5$	AR-CUSUM Last				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	200.5(2.71)	195.4(2.33)	203.8(2.30)	193.3(2.05)	196.1(2.10)
0.5	72.1(0.57)	87.9(0.77)	107.2(0.99)	123.9(1.15)	153.4(1.45)
1.0	24.2(0.13)	24.5(0.15)	25.9(0.17)	28.6(0.20)	32.5(0.25)
1.5	14.2(0.07)	13.3(0.06)	13.2(0.06)	13.4(0.07)	14.1(0.08)
2.0	10.5(0.04)	9.6(0.04)	9.2(0.04)	9.1(0.04)	9.3(0.04)
2.5	8.8(0.03)	8.0(0.03)	7.6(0.03)	7.4(0.03)	7.6(0.03)
3.0	8.1(0.03)	7.3(0.03)	6.8(0.03)	6.8(0.03)	6.8(0.04)

Figure 2.1 shows the SSARL curves of the four procedures with those particular choices of k for the selected distributions. In all the four plots, the SSARLs are plotted on log scale. Figures 2.1 (a) and (b) show the results for the $t_{5,3}$ and Cauchy_5 distributions, both of which are heavy tailed. MCUSUM is not shown in these two plots since it fails to achieve nominal ARL_0 level. In contrast, both SS-CUSUM and AR-CUSUM can achieve the desired ARL_0. However, SS-CUSUM outperforms AR-CUSUM in both of the cases. Figures 2.1 (c) and (d) show the SSARL curves for the skewed distributions $\chi^2_{5,1}$ and $\text{Gamma}_{5,1}$. Although SS-CUSUM is not distribution-free under these two distributions, because we select small values of k SS-CUSUM can still achieve the nominal ARL_0 as shown in the earlier robustness study. From Figures 2.1 (c) and (d), again we can see that the detection power of our SS-CUSUM dominates that of MCUSUM and AR-CUSUM in both cases.

We also carry out a simulation study to compare the detection power of the four CUSUM procedures for location shifts with an equal magnitude in all the components. Without loss of generality, we focus on the downward shifts. We choose the shift magnitude b in every component to vary from 0.2 to 1 with an increment of 0.2. Based on those b's, the choices of k for each of the four CUSUM procedures from the above one-component location shift study will still give the procedures the best or nearly the best detection power across different simulation settings. The SSARL curves of the four CUSUM procedures with those particular choices of k under different distributions are presented in Figure 2.2. Similar to the above one-component location shift study, for the two heavy tailed distributions, $t_{5,3}$ and Cauchy_5, MCUSUM fails to achieve the nominal ARL_0, therefore it is not shown in Figures 2.2 (b) and (c). From all the five plots, we can see that the performance of our

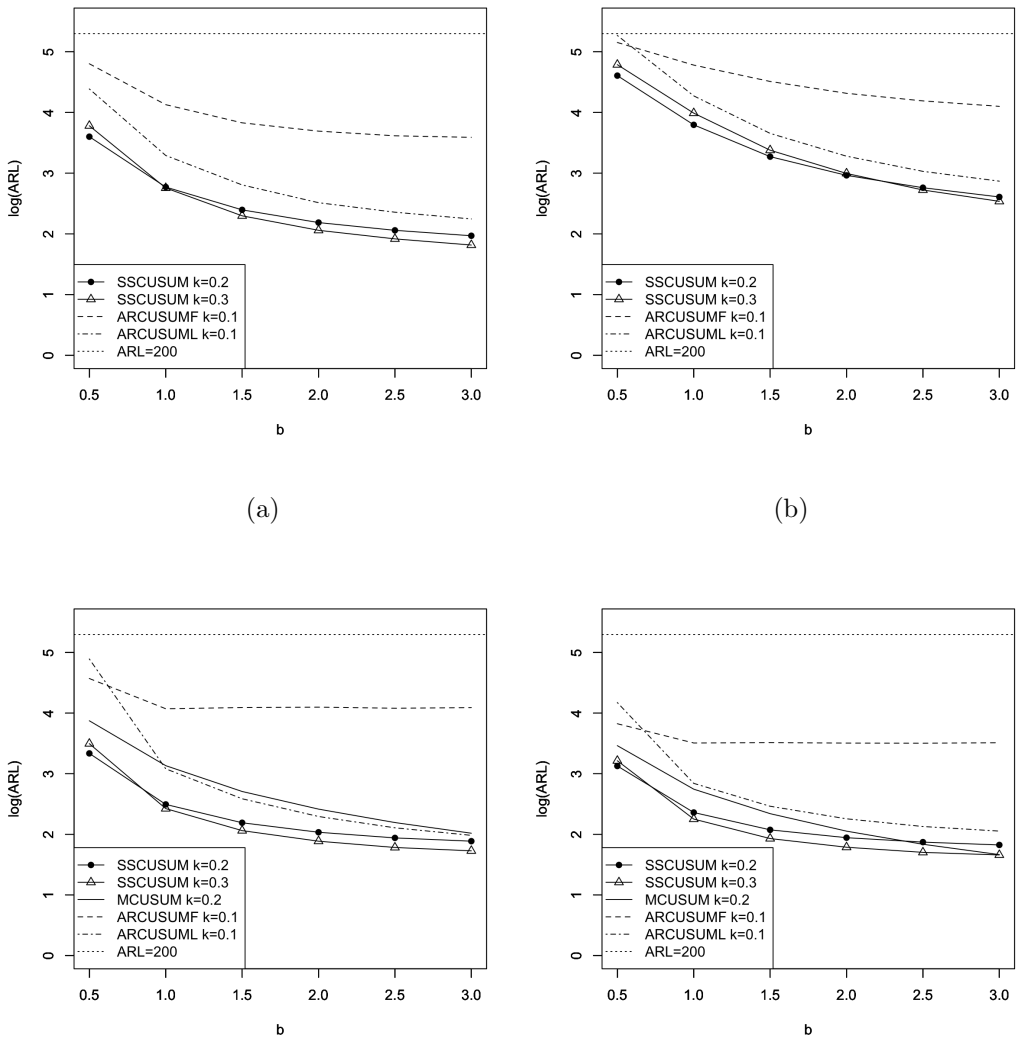


Figure 2.1: Power comparision of SS-CUSUM, MCUSUM, AR-CUSUMF (AR-CUSUM with the first anti-rank), and AR-CUSUML (AR-CUSUM with the last anti-rank) with a shift of b in the first component under: (a) $t_{5,3}$; (b) Cauchy$_5$; (c) $\chi^2_{5,1}$; (d) Gamma$_{5,1}$.

SS-CUSUM is consistently among the best, especially surpasses the performance of the last anti-rank AR-CUSUM in most of the cases. It is worth pointing out that the out-of-control SSARLs for the first anti-rank AR-CUSUM under the five distributions are all larger than the in-control ARL_0, which implies that the first anti-rank AR-CUSUM is powerless for detecting this kind of location shifts. This phenomenon was also briefly mentioned in Qiu and Hawkins (2003) and they referred to it as bias phenomenon, similar to the "biased" statistical test.

In the following we provide some brief explanation for this biased phenomenon. We first write the new observation $\boldsymbol{X}_n$ as $(X_{n1}, ..., X_{n5})'$ and define its observed first anti-rank vector as $\boldsymbol{\eta}_n = (\eta_{n0}, \eta_{n1}, ..., \eta_{n5})'$, where

$$\eta_{n0} = I\{0 \text{ is the smallest among } \{X_{n1}, ..., X_{n5}, 0\}\},$$

$$\eta_{nj} = I\{X_{nj} \text{ is the smallest among } \{X_{n1}, ..., X_{n5}, 0\}\}, \text{ for } j = 1, ..., 5,$$

and $I\{A\}$ is the indicator function and takes 1 if A is true and 0 otherwise. Similarly we can define the expected first anti-rank vector when the process is in control as $\boldsymbol{d}_0 = (d_0, d_1, ..., d_5)'$, where

$d_0 = P\{0 \text{ is the smallest among } \{X_{n1}, ..., X_{n5}, 0\} \text{ when the process is in control } \} = E_0(\eta_{n0})$,

$d_j = P\{X_{nj} \text{ is the smallest among } \{X_{n1}, ..., X_{n5}, 0\} \text{ when the process is in control}\} = E_0(\eta_{nj})$,

for $j = 1, ..., 5$. If we use the Norm_5 distribution as an example,

$$\boldsymbol{d}_0 = (0.03125, 0.19375, 0.19375, 0.19375, 0.19375, 0.19375)'.$$

Given a downward shift of 1 in all the 5 components for the Norm_5 distribution, the observed first anti-rank vector $\boldsymbol{\eta}_n$ has the expected value

$$\boldsymbol{d}_1 = E_1(\boldsymbol{\eta}_n) = (0.0001, 0.19998, 0.19998, 0.19998, 0.19998, 0.19998)',$$

which is quite close to $\boldsymbol{d}_0$. Loosely speaking, the first antirank AR-CUSUM is roughly monitoring $\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2$. Since $\|E_1(\boldsymbol{\eta}_n) - \boldsymbol{d}_0\|^2 = \|\boldsymbol{d}_1 - \boldsymbol{d}_0\|^2 = 0.0011644$, which is very small, it explains why it is difficult for the first anti-rank AR-CUSUM to detect this kind of downward location shifts.

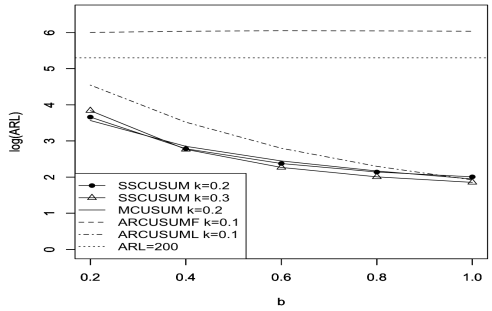
Next we explain why it takes longer for the first anti-rank AR-CUSUM to trigger an alarm when the process is out of control than when it is in control (the biased phenomenon). We still use the Norm_5 distribution as an example. As we mention above, loosely speaking, the monitoring statistic for the first anti-rank AR-CUSUM roughly accumulates $\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2$. If one of the X_{nj} $(j = 1, ..., 5)$ is the smallest among $\{X_{n1}, ..., X_{n5}, 0\}$, $\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2 = 0.801$. If 0 is the smallest among $\{X_{n1}, ..., X_{n5}, 0\}$, $\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2 = 1.126$. Therefore, when the process is in control, $E\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2 = 0.801 \times 0.96875 + 1.126 \times 0.03125 = 0.811$, and when the process is out of control with a downward shift of 1 in all the 5 components, $E\|\boldsymbol{\eta}_n - \boldsymbol{d}_0\|^2 = 0.801 \times 0.9999 + 1.126 \times 0.0001 = 0.801 < 0.811$. This implies that the monitoring statistic is more likely to go beyond the control limit when the process is in control than when the process is out of control. Therefore, it takes longer to trigger the alarm when the process is out of control than when the process is in control.

In the above simulation, the last anti-rank AR-CUSUM has some power for detecting the downward shifts with an equal magnitude in all the components, while the first anti-rank AR-CUSUM is powerless for this purpose. Using the similar argument as above,

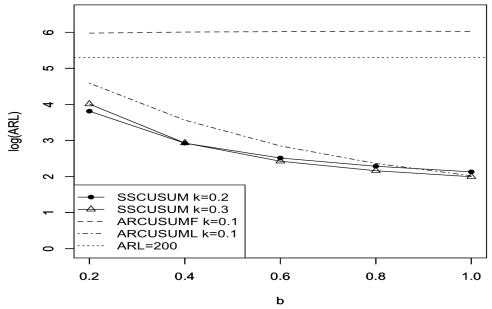
we can also see that the last anti-rank AR-CUSUM will be powerless for detecting the upward shifts with an equal magnitude in all the components. Therefore, which anti-rank to use is very critical for the performance of AR-CUSUM. If the anti-rank is not appropriately selected, the AR-CUSUM procedure can be useless. However, how to choose the anti-rank depends on what kind of location shifts the process will encounter, which is usually unknown in practice. In contrast, our SS-CUSUM can achieve good detection power for different types of location shifts, and it even has comparable performance with MCUSUM when the underlying distribution is multinormal.

2.5.3 Scale Increase Detection

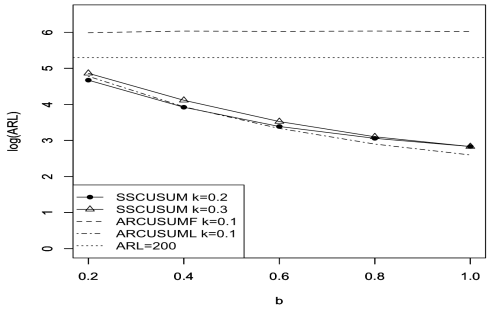
In this section, we consider the scale increase detection, i.e. detecting increases in variability of the process. Based on our simulations, SS-CUSUM is not as powerful as our DD-CUSUM for detecting this kind of change. Therefore, we recommend using DD-CUSUM for detecting scale increases in practice. In this section, we present some simulation studies to investigate the performance of DD-CUSUM for scale increase detection under different distributions. The distributions considered here are Norm_5, $t_{5,3}$, Cauchy_5, $\chi^2_{5,1}$, and $\text{Gamma}_{5,1}$. We consider two scale increase scenarios: (a) scale change by b times in all components, and (b) scale change by b times only in the first component. In both of the cases, b varies from 1 to 8, with an increment of 2. Similar to the previous simulation study for location changes, we introduce the scale change after the 50th observation. The SSARL will be reported for each scale change setting.



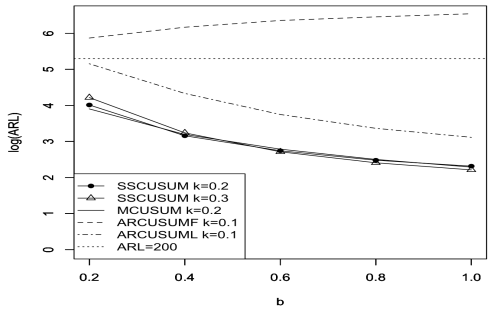
(a)



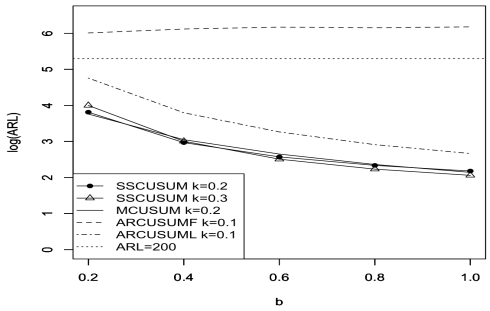
(b)



(c)



(d)



(e)

Figure 2.2: Power comparision of SS-CUSUM, MCUSUM, AR-CUSUMF (AR-CUSUM with the first anti-rank), and AR-CUSUML (AR-CUSUM with the last anti-rank) with a downward shift of b in all the components under: (a) Norm_5; (b) $t_{5,3}$; (c) Cauchy_5; (d) $\chi^2_{5,1}$; (e) $\text{Gamma}_{5,1}$.

For the first scale change scenario where the scales increase by the same magnitude in all components, AR-CUSUM will not have any detection power. This is due to the fact that if all components have the same scale increase, the probability vector for any anti-rank will not change, so that the scheme will not signal any abnormality. However, DD-CUSUM can easily detect this type of scale increase. The simulated ARLs for DD-CUSUM are displayed in Table 2.3. From the table we can see that DD-CUSUM can achieve the desired ARL_0 when the process is in control under all the five distributions. When there is a scale increase, the out-of-control SSARLs decrease rapidly from 200, which indicates the good power of DD-CUSUM for detecting the change.

Next, we would like to compare the performance of DD-CUSUM with AR-CUSUM when there is a scale increase in only one component. This time, AR-CUSUM will have detection power, since scale increases in only one component will alter the order of components, and affect the probability vector of the anti-ranks. We still use the previous distribution settings. The simulated SSARLs are listed in Table 2.4. To save the space, we only show the results for the first anti-rank AR-CUSUM, since its performance is as good as or better than the performance of the last anti-rank AR-CUSUM in our simulation studies. As we can see from the table, for symmetric distributions such as Norm_5, $t_{5,3}$ and Cauchy_5, DD-CUSUM completely dominates AR-CUSUM. However, for $\chi^2_{5,1}$, AR-CUSUM performs better than DD-CUSUM. For $\text{Gamma}_{5,1}$ distribution, the two procedures have comparable performance. AR-CUSUM does better under the $\chi^2_{5,1}$ distribution because the extremely right skewness enables the scale increase in one component to significantly alter the probability distribution of the first anti-rank. As a result, AR-CUSUM with the first

Table 2.3: Detection power for DD-CUSUM: scale increases in all components

Norm_5

b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
1	193.1(1.85)	194.6(1.94)	188.3(1.88)	206.6(2.05)
2	6.8(0.04)	6.6(0.05)	6.2(0.05)	7.1(0.06)
4	3.9(0.01)	3.2(0.01)	2.6(0.01)	2.6(0.01)
6	3.5(0.01)	2.9(0.01)	2.2(0.01)	2.2(0.01)
8	3.4(0.01)	2.8(0.01)	2.0(0.01)	2.0(0.01)

$t_{5,3}$

b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
1	183.1(1.79)	197.1(1.91)	216.2(2.08)	193.6(1.95)
2	13.7(0.10)	18.6(0.16)	27.7(0.26)	35.1(0.34)
4	5.7(0.03)	5.8(0.03)	7.1(0.05)	9.1(0.08)
6	4.6(0.02)	4.1(0.02)	4.4(0.03)	5.2(0.04)
8	4.1(0.01)	3.5(0.01)	3.4(0.02)	3.8(0.02)

Cauchy_5

b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
1	196.7(1.88)	206.7(2.03)	199.6(1.96)	193.8(1.95)
2	34.1(0.30)	56.5(0.54)	81.6(0.79)	97.2(0.95)
4	11.9(0.07)	18.2(0.15)	32.1(0.30)	48.8(0.48)
6	8.4(0.04)	11.2(0.08)	19.6(0.18)	33.2(0.32)
8	7.1(0.03)	8.3(0.05)	13.9(0.12)	25.2(0.24)

$\chi^2_{5,1}$

b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
1	186.2(1.80)	191.0(1.90)	201.9(1.99)	200.4(1.99)
2	18.3(0.15)	21.6(0.20)	24.7(0.24)	28.3(0.27)
4	7.5(0.05)	7.5(0.05)	7.5(0.06)	8.4(0.07)
6	5.8(0.03)	5.3(0.03)	5.0(0.04)	5.3(0.04)
8	5.0(0.02)	4.4(0.02)	4.0(0.02)	4.1(0.03)

$\text{Gamma}_{5,1}$

b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$
1	194.6(1.90)	199.0(1.96)	199.4(1.98)	201.8(2.02)
2	14.5(0.11)	16.2(0.14)	18.0(0.16)	21.0(0.20)
4	6.0(0.03)	5.6(0.03)	5.3(0.04)	5.9(0.05)
6	4.6(0.02)	4.1(0.02)	3.6(0.02)	3.8(0.03)
8	4.1(0.02)	3.5(0.01)	3.0(0.02)	3.1(0.02)

anti-rank is more powerful in detecting this kind of change.

To investigate the performance of DD-CUSUM and AR-CUSUM under moderately skewed distributions, we conduct a power comparison study under $\chi^2_{5,3}$, $\chi^2_{5,5}$, $\text{Gamma}_{5,3}$ and $\text{Gamma}_{5,5}$. The results are shown in Table 2.5. It is obvious from the table that DD-CUSUM outperforms AR-CUSUM in all the cases. When the distribution becomes less skewed, the power advantage of DD-CUSUM over AR-CUSUM becomes more significant.

The conclusion from the above simulation study is that unless you have a-priori concern for extreme skewed distributions, we recommend the DD-CUSUM procedure. If the potential for extreme skewed distributions exists, then we recommend running both the DD-CUSUM and AR-CUSUM procedures in parallel.

Table 2.4: Power comparison for scale increases in one component

Norm_5

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	198.8(1.93)	201.4(1.99)	200.3(2.00)	205.1(2.72)	200.9(2.39)	206.0(2.32)
2	41.0(0.38)	47.4(0.45)	48.2(0.46)	175.9(1.73)	185.6(1.84)	189.5(1.87)
4	14.3(0.11)	15.4(0.13)	14.5(0.13)	83.4(0.71)	97.3(0.90)	111.7(1.08)
6	9.9(0.07)	9.9(0.08)	8.7(0.07)	60.0(0.47)	69.8(0.60)	80.8(0.74)
8	8.1(0.05)	7.8(0.06)	6.8(0.05)	50.8(0.38)	57.0(0.47)	65.7(0.58)

$t_{5,3}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	190.5(1.86)	195.6(1.93)	196.7(1.95)	200.3(2.72)	199.5(2.42)	199.4(2.27)
2	74.2(0.73)	96.2(0.96)	111.9(1.12)	173.0(1.65)	186.1(1.87)	187.2(1.83)
4	29.6(0.26)	39.6(0.37)	50.8(0.50)	84.3(0.71)	97.4(0.89)	111.0(1.06)
6	18.7(0.15)	24.3(0.22)	30.3(0.29)	60.4(0.48)	69.3(0.61)	80.8(0.74)
8	14.3(0.11)	16.9(0.15)	21.0(0.19)	50.1(0.37)	57.0(0.47)	65.6(0.58)

Cauchy_5

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	194.0(1.85)	207.7(2.03)	195.3(1.94)	199.0(2.66)	198.6(2.38)	204.4(2.29)
2	121.5(1.20)	153.4(1.53)	157.6(1.60)	240.7(2.44)	222.3(2.24)	218.4(2.17)
4	72.0(0.70)	104.1(1.03)	121.1(1.20)	160.1(1.53)	171.7(1.66)	179.8(1.81)
6	51.7(0.49)	81.3(0.79)	98.4(0.98)	122.5(1.11)	138.2(1.33)	151.2(1.50)
8	41.2(0.38)	66.6(0.64)	87.2(0.87)	101.6(0.91)	117.7(1.09)	134.7(1.49)

$\chi^2_{5,1}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	193.8(1.87)	199.6(1.97)	204.3(2.04)	202.1(2.24)	201.8(2.17)	203.1(2.06)
2	87.8(0.87)	101.3(1.01)	107.1(1.07)	30.9(0.19)	31.2(0.21)	32.3(0.24)
4	41.9(0.39)	47.6(0.46)	49.0(0.48)	20.9(0.11)	20.1(0.12)	19.5(0.12)
6	29.1(0.27)	32.1(0.30)	32.2(0.31)	19.0(0.10)	17.5(0.09)	16.9(0.10)
8	22.9(0.20)	25.2(0.23)	24.4(0.23)	17.7(0.09)	16.6(0.09)	15.7(0.09)

$\text{Gamma}_{5,1}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	188.0(1.81)	198.7(1.95)	190.2(1.87)	202.3(2.65)	201.7(2.35)	203.0(2.22)
2	69.8(0.68)	79.8(0.78)	80.7(0.81)	52.3(0.39)	61.7(0.51)	78.9(0.71)
4	27.0(0.24)	30.7(0.29)	30.3(0.28)	27.7(0.16)	29.9(0.20)	34.3(0.25)
6	17.3(0.14)	19.1(0.17)	17.7(0.16)	24.1(0.14)	24.6(0.15)	27.2(0.18)
8	13.4(0.10)	14.1(0.12)	13.2(0.11)	22.1(0.12)	22.1(0.13)	24.7(0.16)

Table 2.5: Power comparison for scale increases in one component

$\chi^2_{5,3}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	196.2(1.92)	200.5(1.96)	194.9(1.92)	201.6(2.52)	199.6(2.17)	201.7(2.13)
2	60.5(0.58)	67.1(0.65)	69.9(0.68)	75.6(0.62)	110.2(1.04)	175.0(1.71)
4	20.7(0.18)	22.3(0.21)	22.0(0.21)	35.9(0.23)	44.3(0.32)	66.1(0.55)
6	12.9(0.10)	13.5(0.11)	12.7(0.11)	28.5(0.17)	34.1(0.23)	47.2(0.35)
8	10.0(0.07)	9.8(0.08)	9.1(0.08)	26.0(0.15)	29.8(0.19)	39.7(0.28)

$\chi^2_{5,5}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	203.6(1.96)	199.7(1.97)	198.7(1.97)	202.5(2.57)	200.2(2.17)	202.6(2.19)
2	48.2(0.46)	53.0(0.53)	55.7(0.53)	96.1(0.85)	129.5(1.24)	175.6(1.74)
4	14.0(0.11)	14.5(0.13)	13.9(0.13)	42.1(0.29)	52.8(0.42)	75.5(0.67)
6	8.8(0.06)	8.5(0.07)	7.9(0.07)	32.9(0.21)	38.5(0.28)	52.2(0.42)
8	6.9(0.04)	6.4(0.04)	5.7(0.04)	29.5(0.18)	33.0(0.23)	43.5(0.33)

$\text{Gamma}_{5,3}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	187.4(1.80)	194.9(1.93)	198.5(1.93)	197.4(2.66)	203.0(2.44)	201.3(2.23)
2	40.6(0.39)	47.1(0.45)	50.2(0.49)	104.8(0.94)	123.2(1.16)	134.2(1.32)
4	11.5(0.08)	11.9(0.10)	11.8(0.10)	44.7(0.31)	50.7(0.40)	56.7(0.48)
6	7.3(0.04)	7.2(0.05)	6.5(0.05)	34.9(0.23)	37.1(0.27)	41.8(0.33)
8	5.9(0.03)	5.4(0.03)	4.8(0.03)	30.7(0.19)	31.5(0.22)	34.9(0.26)

$\text{Gamma}_{5,5}$

$p = 5$	DD-CUSUM			AR-CUSUM First		
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.1$	$k = 0.2$	$k = 0.3$
1	204.1(1.97)	202.9(1.99)	198.6(1.98)	200.4(2.69)	200.9(2.37)	196.9(2.21)
2	30.7(0.28)	34.0(0.32)	34.0(0.32)	130.2(1.20)	143.8(1.38)	156.3(1.53)
4	7.8(0.05)	7.7(0.06)	7.1(0.06)	53.9(0.39)	60.7(0.51)	69.7(0.62)
6	5.3(0.03)	4.7(0.03)	4.1(0.03)	40.6(0.27)	43.6(0.33)	50.4(0.43)
8	4.5(0.02)	3.9(0.02)	3.2(0.02)	35.6(0.22)	37.1(0.26)	41.6(0.33)

Chapter 3

Nonparametric CUSUM Control Chart for Autocorrelated Processes

3.1 Introduction

The development of modern sampling technology enables one to sample data at a high frequency from a process. However, the more complex autocorrelation structure within the high frequency observations creates challenge to the process monitoring procedure. One example of this process is illustrated in Figure 3.1 (a), which demonstrates a portion of CPU usage records of a network server that motivates this work. The full data consist of the CPU usage of a network server from May 31st to November 21st in 2010, with measurements recorded every 5 minutes. The data shown in Figure 3.1 (a) is the snapshot of the CPU usage for one week. One can find the histogram of the data in Figure 3.1 (b) and the Autocorrelation Function (ACF) in Figure 3.1 (c). From the Figure 3.1 (c), we can see that

there is a slowly decaying autocorrelation within the observations. In fact, many network data demonstrate such correlation structure, and it is sometimes considered as Long-Range Dependence (LRD) (Park et al. (2011)).

It has been shown that for the conventional control charts (such as the original versions of Shewhart chart, Cumulative Sum (CUSUM) chart, and Exponentially Weighted Moving Average (EWMA) chart), deviation from either the distributional assumption or the iid assumption or both may result in either more frequent false alarms (e.g. failing to control the in-control average run length (denoted by ARL_0)) or weakened anomaly detection power (e.g. enlarging the out-of-control average run length (denoted by ARL_1)) (see Johnson and Bagshaw (1974) and Black et al. (2011) for example).

There have been many studies in developing SPC procedure for serially correlated observations with the normality assumption. Most of them are the so called "residual-based" methods, where a parametric model is assumed to describe the correlation structure within the data, and the approximately uncorrelated residuals are extracted for use in a control chart (see Harris and Ross (1991), Runger and Willemain (1995), and Cheng and Thaga (2005) for details). The parametric models used could be Box and Jenkins time series models (i.e. autoregressive (AR) models and autoregressive moving average (ARMA) models) for stationary processes with a fast decaying auto-correlation. For the LRD cases, a popular model is the fractional ARIMA model (details in Section 3.4). However, the model assumption used in this approach is difficult to justify in the real application. Especially for the LRD case where a fractional ARIMA model is utilized, the estimate of the fractional differencing parameter d is based on the maximum likelihood (ML) method, where

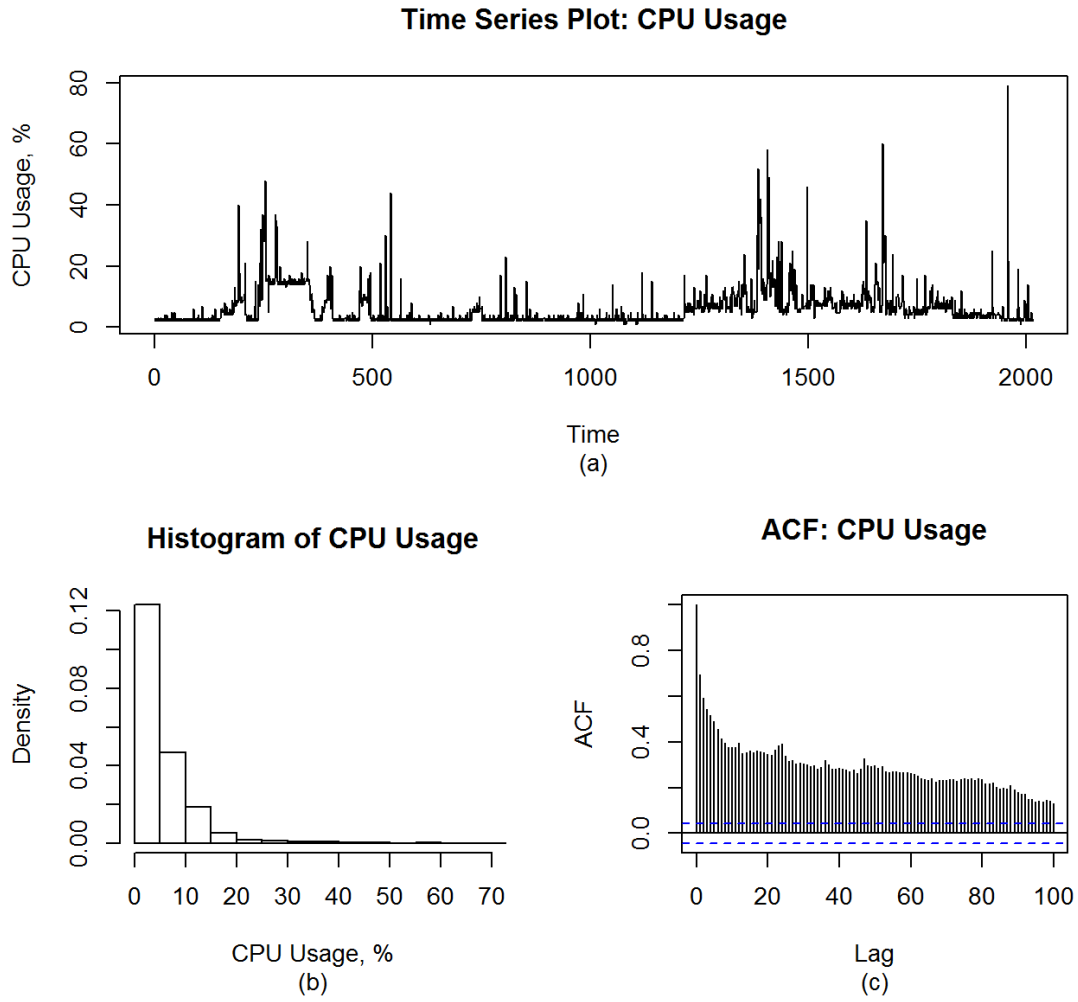


Figure 3.1: Real CPU Usage Data: (a) illustrates the CPU usage data, (b) demonstrates the histogram of the data in (a), and (c) represents the autocorrelation function (ACF) of the data in (a)

a distributional assumption is required. A deviation from this assumption would result in unreliable parameter estimates, which in turn affects the correlation structure of the estimated residuals. Therefore, the ARL_0 and ARL_1 performance of the control charts could be compromised ultimately. The control charts used are the variants of classic Shewhart chart, CUSUM chart, or EWMA chart.

Another approach for handling correlation is to use a data transformation technique to diminish its magnitude. For instance, Runger and Willemain (1995) proposed the weighted and unweighted Batch Means chart (referred to as WBM and UBM chart, respectively). The WBM chart assumes some specific correlation structures in the data (e.g. first order autoregressive (AR(1)) model) and transform the data accordingly to eliminate the autocorrelation. In some applications, including network surveillance, it may not always be possible to find such a transformation, especially where LRD is present. The UBM chart does not make correlation structure assumption. However, its performance was only evaluated with data from an AR(1) model. The performance and validity of UBM under more complex correlation structure is unknown. Besides what has been discussed above, both of these methods could only solve the problem where the correlation decays fairly fast. For the situation where the correlation sustains for a longer period of time (i.e. LRD), it would be difficult for these methods to transform the data and break up the autocorrelation successfully.

To overcome the previously mentioned difficulties, we propose a wavelet-based nonparametric CUSUM control chart in this chapter. Wavelet analysis has been applied to a variety of fields in statistics, such as nonparametric regression, density estimation, and

time series analysis. In the SPC field, studies concentrate on either using wavelet analysis to denoise and pre-process the data before a SPC method is conducted, or relying on the multi-scale properties of the wavelet based techniques to detect different types of changes in the process (see Ganesan et al (2010) for a review). However, studies that apply wavelet analysis in control charts for serially correlated process, especially one possessing LRD is still rare in the literature. In this chapter, we propose a SPC procedure which utilizes the approximate decorrelation property of wavelets that makes the procedure robust to different autocorrelation structures. At the same time, the procedure incorporates the SS-CUSUM control chart proposed in Chapter 2. It is therefore robust to the situations where observations do not follow a normal distribution.

The rest of the chapter is organized as follows. Since wavelet analysis is the key component of our procedure, we give a brief review in Section 3.2. Our proposed methodology is detailed in Section 3.3. In Section 3.4.1 and 3.4.2, we report simulation studies that compare the ARL_0 and ARL_1 performance between our propsed method and the residual based method. Some practical issues are discussed in Section 3.4.3. Finally, in Section 3.5, we will revisit the CPU usage example to demonstrate the real application of our proposed procedure.

3.2 Wavelets

Assume we have a discrete signal $\boldsymbol{y} = (y_1, y_2, \cdots, y_N)'$. For simplicity, we further assume $N = 2^n$. The wavelet tranformation of the signal involves passing the sequence through two filters, known as low pass filter (denoted as $\mathbf{h}$) and high pass filter (denoted as

$\mathbf{g}$), respectively. The number of nonzero elements in $\mathbf{h}$ and $\mathbf{g}$ is called the filter length. The filters take the input sequence $\boldsymbol{y}$ and produce two series of coefficients via convolution product $\mathbf{h} * \boldsymbol{y}$ and $\mathbf{g} * \boldsymbol{y}$, where $*$ denotes the convolution product. The resulting series of coefficients are referred to as the first scale scaling coefficients (denoted as $\mathbf{c}_1 = (c_{1,1}, \cdots, c_{1,N/2})'$) and the first scale wavelet coefficients (denoted as $\mathbf{w}_1 = (w_{1,1}, \cdots, w_{1,N/2})'$), respectively. Note that the scaling coefficients and the wavelet coefficients both are half the length of the original sequence. The scaling coefficients can be considered as an approximation of the original input sequence $\boldsymbol{y}$, while the wavelet coefficients provide information about how far the scaling coefficients are from the original series.

The wavelet transformation described above could be conducted iteratively in order to decompose the signal to a specific scale j ($j \leq n$). This can be achieved by sequentially applying the low and high pass filters to the previous scale coefficients. This method is known as the "Pyrimidal Algorithm". More precisely, the vector of the coefficients at scale j can be computed recursively as $\mathbf{c}_i = \mathbf{h} * \mathbf{c}_{i-1}$ and $\mathbf{w}_i = \mathbf{g} * \mathbf{c}_{i-1}$ with $i = 1, 2, \cdots, j$ and $\mathbf{c}_0 = \boldsymbol{y}$. The scaling coefficients at the jth scale are known as the scaling coefficients at the coarsest scale. Together, all j scales of wavelet coefficients and the scaling coefficients at the coarsest scale contain all the information in $\boldsymbol{y}$.

There are many different wavelets in the literature. A simplest example is the Haar wavelet proposed by Alfréd Haar in 1909. The low pass and high pass filters for Haar wavelet are $\mathbf{h} = (\frac{1}{\sqrt{2}} \ \frac{1}{\sqrt{2}})$ and $\mathbf{g} = (\frac{1}{\sqrt{2}} \ -\frac{1}{\sqrt{2}})$, respectively. Hence the Harr wavelet has a filter length of 2. For a more complete list of wavelets and their corresponding filters, one can consult Daubechies (1988).

Wavelets have many nice properties which make them widely applicable to many fields in statistics. The most appealing property for our application is the approximate decorrelation property that yields wavelet coefficients approximately uncorrelated even if the original data is serially correlated (Aradhye et al. (2003)). This phenomenon can be explained by the fact that the wavelets are approximate eigenfunctions of many mathematical operators (Beylkin et al. (1991)). Even for fractionally differenced series which possess LRD, the correlation structure of wavelet coefficients within each scale can often be modeled adequately by an autoregressive model of order 1 (AR(1)) (Craigmile and Percival (2005)). In Figure 3.2, we show the ACF of Haar wavelet coefficients computed on data illustrated in Figure 3.1. From the figures, we can see that the ACF of wavelet coefficients no longer present LRD, and the correlation structure may be described adequately by an AR(1) model. This empirical result as well as simulation results shown in Section 3.4 suggest we could model wavelet coefficients using an AR(1) model, and then extract the approximately uncorrelated residuals for input into a suitable control charts.

However, using the uncorrelated residuals at each scale to build separate control charts would result in a different comparison problem since the between-scale correlation cannot be neglected. As a result, it would be difficult to control the overall ARL_0. One possible way to account for the between scale correlation and successfully control the overall ARL_0 is to put residuals from different scales into a multivariate vector and monitor the process using a multivariate control chart. In this dissertation, we use the SS-CUSUM control chart introduced in Chapter 2.

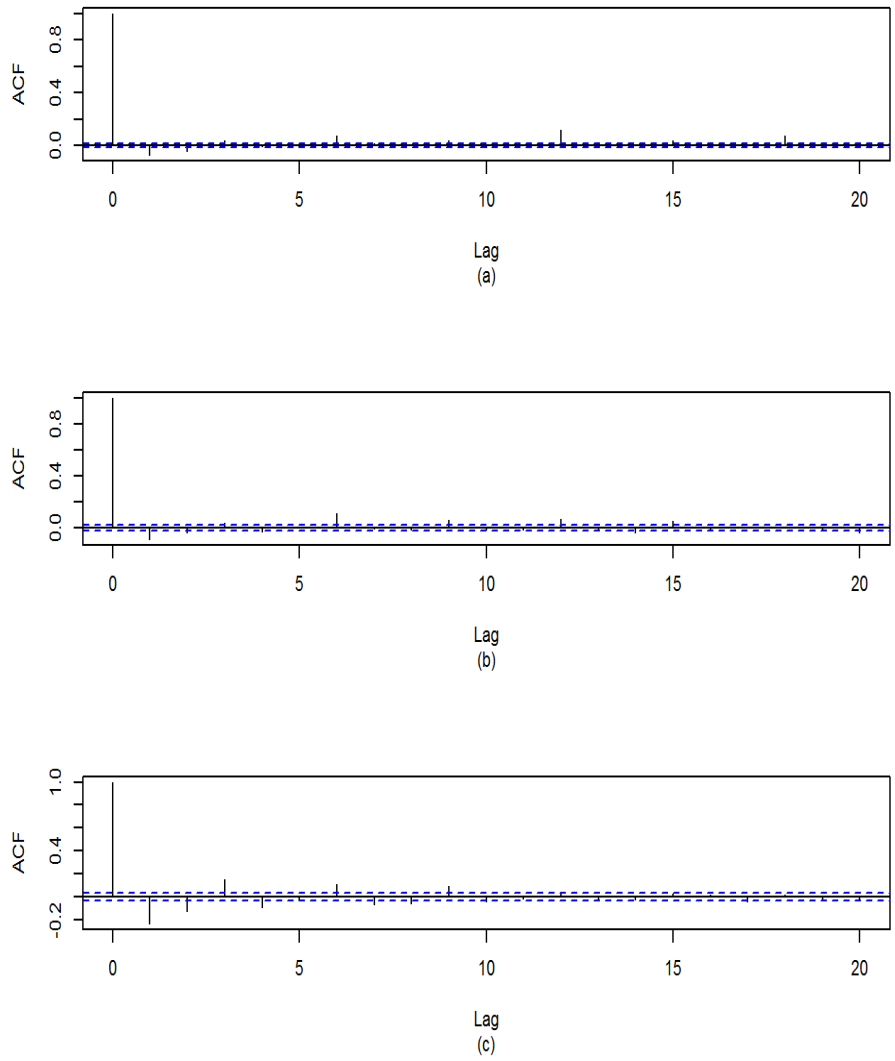


Figure 3.2: ACF of Wavelet Coefficients: (a) - (c) represents the ACFs of wavelet coefficients from data in Figure 3.1 (a) at scale 1, 2, and 3, respectively.

3.3 Wavelet-Based SS-CUSUM Control Chart

In this section, we detail our wavelet-based nonparametric SS-CUSUM procedure for processes with serially correlated observations. Assume that we have a reference sample $(y_1, \cdots, y_N)$. For simplicity, we assume $N = 2^n$ throughout. Further denote the new observations by $(x_1, x_2, \cdots)$, which comes from the same mechanism as the reference sample if there is no change in the process. Suppose that our analysis requires a l-scale ($l < n$) wavelet decomposition (details of determining the decomposition scale is given in Section 3.4.3). Then from the discussion in Section 3.2, every 2^l observations would produce 2^{l-1} first scale wavelet coefficients, 2^{l-2} second scale wavelet coefficients, etc. and one l scale scaling coefficient. For instance, reference observations $(y_1, \cdots, y_{2^l})$ could produce $(w_{1,1}, \cdots, w_{1,2^{l-1}}, w_{2,1}, \cdots, w_{2,2^{l-2}}, \cdots, w_{l,1}, c_{l,1})$. These 2^l coefficients contain all the information in the 2^l observations. Therefore, we put every 2^l reference observations into one batch, so that the reference sample is grouped into 2^{n-l} batches. From each batch, there are 2^{l-i} ith scale wavelet coefficients, $i = 1, \cdots, l$ and one scaling coefficient produced in the wavelet decomposition. If we group the wavelet coefficients at the same scale across all the 2^{n-l} batches, for the ith scale wavelet coefficients, we have $(w_{i,1}, \cdots, w_{i,2^{l-i}}, w_{i,2^{l-i}+1}, \cdots, w_{i,2^{l-i+1}}, \cdots, w_{i,2^{n-i}-2^{l-i}+1}, \cdots, w_{i,2^{n-i}}), i = 1, \cdots, l$, and for the coarest level scaling coefficients we have $(c_{l,1}, \cdots, c_{2^{n-l}})$.

Within each scale of wavelet coefficients, due to Craigmile et al. (2005), an AR(1) model is sufficient to account for the within-scale auto-correlation of the wavelet coefficients. Hence, we fit l separate AR(1) models for each of the l scales of the wavelet coefficients. For instance, at scale $i, i = 1, \cdots, l$, the AR(1) model can be written as

$$w_{i,t} - \mu_{w_i} = \alpha_i(w_{i,t-1} - \mu_{w_i}) + \epsilon_{i,t}, \quad t = 1, \cdots, 2^{n-i}$$

where the μ_{w_i} is the mean of $w_{i,t}$ and the $\epsilon_{i,t}$ are iid random variables with mean 0 and standard deviation σ_{ϵ_i}. The estimate of the autoregressive coefficient α_i (denoted by $\hat{\alpha}_i$) and μ_{w_i} (denoted by $\hat{\mu}_{w_i}$) can be obtained by Least Square (LS) method. The estimated residuals can be found by

$$\hat{e}_{i,t} = w_{i,t} - \hat{w}_{i,t}, \quad t = 1, \cdots, 2^{n-i} \tag{3.1}$$

where $\hat{w}_{i,t} = \hat{\alpha}_i(w_{i,t-1} - \hat{\mu}_{w_i})$ is the predicted value of $w_{i,t}$. The residuals for the ith scale are recorded as $\hat{\mathbf{e}}_i = (\hat{e}_{i,1}, \cdots, \hat{e}_{i,2^{n-i}})'$.

For the coarsest level scaling coefficients denoted as $\mathbf{c}_l = (c_{l,1}, \cdots, c_{l,2^{n-l}})'$, since they can be considered as an approximation of the original data, the correlation structure within the coefficients depends on the original data. Based on our observation, the scaling coefficients are non-stationary in our real data. Therefore, we propose to use the following autoregressive integrated moving average model with $p = 1, d = 1, q = 1$ (ARIMA(1, 1, 1)) to account for its auto-correlation structure. If we denote the one-step differencing series by $D_t = c_{l,t} - c_{l,t-1}$, the ARIMA(1, 1, 1) model can be written as

$$D_t = \beta D_{t-1} + \epsilon_{c,t} + \gamma \epsilon_{c,t-1}$$

where β is the autoregressive coefficient and γ is the moving average coefficient. The $\epsilon_{c,t}$ are iid random variables with mean 0 and standard deviation σ_{ϵ_c}. The estimates of β and

γ (denoted as $\hat{\beta}$ and $\hat{\gamma}$, respectively) can be obtained similarly using the LS method. Then the estimated residuals can be calculated as

$$\hat{e}_{c,t} = D_t - \hat{D}_t \tag{3.2}$$

where $\hat{D}_t = \hat{\beta} D_{t-1} + \hat{\gamma}\hat{e}_{c,t-1}$ is the predicted value of D_t. The residuals are then recorded as $\hat{\mathbf{e}}_c = (\hat{e}_{c,1}, \cdots, \hat{e}_{c,2^{n-j}})$.

Once we obtain the residuals $\hat{\mathbf{e}}_i, (i = 1, \cdots, l)$, and $\hat{\mathbf{e}}_c$, we group all the residuals from the same batch of original data into the same vector. For instance, residuals $(\hat{e}_{1,2^{l-1}(m-1)+1}, \cdots, \hat{e}_{1,2^{l-1}m}, \hat{e}_{2,2^{l-2}(m-1)+1}, \cdots, \hat{e}_{2,2^{l-2}m}, \cdots, \hat{e}_{i,2^{l-i}(m-1)+1}, \cdots, \hat{e}_{i,2^{l-i}m}, \cdots, \hat{e}_{l,m}, \hat{e}_{c,m})$ are all originated from the mth batch of observations $(y_{2^l(m-1)+1}, \cdots, y_{2^l m}), m = 1, \cdots, 2^{n-l}$. Therefore, we put these 2^l residuals into a vector to represent the information contained in $(y_{2^l(m-1)+1}, \cdots, y_{2^l m})$. We denote $\mathbf{Y}_m = (\hat{e}_{1,2^{l-1}(m-1)+1}, \cdots, \hat{e}_{1,2^{l-1}m}, \cdots, \hat{e}_{l,m}, \hat{e}_{c,m}), m = 1, \cdots, 2^{n-l}$. Then $(\mathbf{Y}_1, \cdots, \mathbf{Y}_{2^{n-l}})$ can be considered as the reference sample in our procedure. Since, after the wavelet transformation and time series model fitting, the residuals in the vectors are approximately decorrelated, $(\mathbf{Y}_1, \cdots, \mathbf{Y}_{2^{n-l}})$ can be considered uncorrelated, and the SS-CUSUM procedure can be applied. The solutions $(\hat{\boldsymbol{\theta}}_{2^{n-l}}, \hat{A}_{2^{n-l}})$ can be found based on equations (2.2) and (2.3).

For the new observations, we need to wait until there are 2^l observations before we can apply the same wavelet decomposition as we did for the reference data. For every 2^l new observations collected, we apply (3.1) and (3.2) to each corresponding scale of wavelet and scaling coefficients to find the residuals. The 2^l residuals are put into an 2^l-dimensional vector similarly as for the reference sample and we denote it as $\mathbf{X}$. Then the transformation

defined in (2.1) is applied to $\mathbf{X}$, and the resulting $\mathbf{X}^*$ is used in the SS-CUSUM procedure described in Section 2.3. If there is a mean shift in the new observations, this change can be reflected in all scales of wavelet and scaling coefficients. The means of the estimated residuals for each scale would change, which can be detected by the SS-CUSUM control chart. Note that all the 2^l wavelet and scaling coefficients are important in our procedure. The wavelet coefficients are sensitive to the abrupt mean change in the process, while the scaling coefficients are better in detecting persistent change (Aradhye et al. (2003)). In our application where the mean shift is persistent, the scaling coefficients contribute more in the final detection. However, the wavelet coefficients would pick up the change immediately after its happenning, so that the test statistic could be raised toward the control limit, and as a result, the detection time could be shortened.

When modeling the correlation structures of the wavelet coefficients, we use the AR(1) models due to the recommendation from Craigmile et al. (2005). We also use ARIMA(1, 1, 1) model to model the scaling coefficients due to their non-stationary characteristics. One could use other more sophisticated models such as AR(p) ($p > 1$), ARMA(p, q), or ARIMA(p, d, q) models to account for the correlation structures. Our choice of using fixed order AR and ARIMA models here could help make our procedure easier to automate and more convenient to use for practioners. Our simulation studies in Section 3.4 show that this choice of models works appropriately under a variety of settings.

One drawback of our procedure is that there is a delay of 2^l in our detection due to the grouping of the new observations. Therefore, the choice of l is important for the detection power of our scheme. In order to reduce the delay, we should choose a smaller

l. However, the larger the l is, the better the decorrelation power. Therefore, one should choose an l that balances the time delay and the decorrelation power. Generally speaking, when the correlation is known *a priori* to be decaying faster, one should choose a smaller l, while when the correlation is known to be decaying very slowly, one should choose a larger l accordingly. A method to choose l based on multivariate run test proposed by Paindaveine (2009) will be discussed in Section 3.4.3.

Another problem worth discussing is the choice of wavelet filter. According to Craigmile et al. (2005), longer wavelet filters will asymptotically decorrelate the between-scale wavelet coefficients. However, the increase in wavelet filter length would lead to an increase in the covariance in the within-scale wavelet coefficients. Since our goal is to decorrelate the within-scale wavelet coefficients, and the correlation of between-scale coefficients can be accounted for by the multivariate control chart, we should choose a shorter wavelet filter. Therefore, in this chapter, we choose to use Haar wavelet, which has the shortest filter length.

3.4 Simulation Studies

In this section, we report simulation studies to evaluate our proposed method, and compare it to the method which uses nonparametric control charts based on residuals from fractional ARIMA models. Fractional ARIMA models are proposed by Granger and Joyeus (1980) and Hosking (1981) to model the LRD time series. Before we detail the simulation studies, we first briefly review the fractional ARIMA model.

For a time series $\{y_t\}_{t=1}^m$, we denote B as the backshift operator where $B^j y_t = y_{t-j}$,

and define polynomials

$$\phi(x) = 1 - \sum_{j=1}^{p} \phi_j x^j$$

and

$$\psi(x) = 1 - \sum_{j=1}^{q} \psi_j x^j.$$

We can write the commonly used ARIMA(p, d, q) model as

$$\phi(B)(1 - B)^d y_t = \psi(B)\epsilon_t \tag{3.3}$$

where d can only take integer vlaues, and ϵ_t is Gaussion noise. The process can be transformed into an ARMA(p, q) process if we difference y_t d times. However, (3.3) can also be generalized natually by allowing d to take any real values. To achieve this, first note that $(1 - B)^d$ can be written as

$$(1 - B)^d = \sum_{k=0}^{d} \binom{d}{k} (-1)^k B^k$$

with the binomial coefficients

$$\binom{d}{k} = \frac{d!}{k!(d-k)!} = \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)}$$

where $\Gamma(\cdot)$ denotes the gamma function. Since gamma function is also defined for all real numbers, we can formally define $(1 - B)^d$ for any real number d by

$$(1 - B)^d = \sum_{k=0}^{\infty} \binom{d}{k} (-1)^k B^k = \sum_{k=0}^{\infty} \frac{\Gamma(d+1)}{\Gamma(k+1)\Gamma(d-k+1)} (-1)^k B^k \tag{3.4}$$

By doing so, the commonly used ARIMA(p, d, q) model in (3.3) is generalized to a fractional ARIMA model using (3.4). In the LRD context, the most interesting range for d is $0 \leq d < \frac{1}{2}$, since the process is stationary when d is within this range (Beran (1994)).

The parameter estimates of the fractional ARIMA models can be obtained using normal maximum likelihood (ML) methods described in Haslett and Raftery (1989). An automatic ML based parameter estimate procedure can be found in Hyndman and Khandakar (2008), which we use to get the parameter estimates in our simulation studies. The algorithm first assumes a fractional ARIMA (2,d,0) model, and uses normal MLE method to find d, which is used to difference the time series. An ARMA (p, q) model is then selected using the differenced time series. Then finally the full model is re-estimated using MLE method with the value of (p, q) from last step.

After the parameter estimates of the fractional ARIMA model are found, the estimated residuals can be obtained by subtracting the predicted value $\hat{y}_t$ from the observed value y_t. The predicted value $\hat{y}_t$ is calculated using method described in Peiris and Perera (1987).

As discussed in Section 3.1, a SPC procedure can be formulated by charting the approximately uncorrelated residuals from the fractional ARIMA model. The control charts we use in our simulation studies are distribution-free charts. There are two schemes we use. The first one is proposed by Willemain and Runger (1996) (referred to as "Runger Chart" throughout). It is a Shewhart-type chart based on the empirical distribution estimated from the reference sample. This method basically uses the empirical order statistics as the control limits. If ARL_0 is set to be a, then the control limits are $(Y_{(\frac{m}{2a})}, Y_{(m-\frac{m}{2a})})$, where $Y_{(i)}$ denotes the ith order statistic for reference sample $\{Y_t\}_{t=1}^m$.

The second procedure is the modification of the DD-CUSUM chart we proposed in Chapter 2 (referred to as "Rank CUSUM Chart" throughout). This procedure was proposed

originally as a multivariate control chart based on data depth, the multivariate version of rank. We modify the procedure by using the univariate rank directly. The procedure can be summarized as follows.

If we have a reference sample $(Y_1, \cdots, Y_m)$ and we define the R statistic of a new observation X_n as $R_n = \frac{\text{number of } Y_i < X_n}{m+1}$, then the R_n asymptotically follows a uniform distribution on $(0, 1)$ as $m \to \infty$, when X_n is from the same distribution as the reference sample (see McDonald (1990)). Then the Rank CUSUM procedure can be defined as

$$S_n^+ = \max(0, S_{n-1}^+ - (0.5 - R_n) - k)$$

$$S_n^- = \min(0, S_{n-1}^- - (0.5 - R_n) + k)$$

where $S_0^{\pm} = 0$ and $k > 0$. The procedure triggers an alarm when $S_n^+ > h$ or $S_n^- < -h$, where h is the control limit predetermined based on the choice of k and the desired ARL_0. One can use the bisection search algorithm discussed in Section 2.5 to obtain h.

3.4.1 ARL_0 Performance Evaluation

In this section, we evaluate the ARL_0 performance of our proposed method and compare it with the Runger chart and Rank CUSUM chart utilizing the estimated residuals from fractional ARIMA models. We also consider a situation where data is generated by a generalized linear mixed model.

We first consider fractional ARIMA(1, d, 1) and fractional ARIMA(3, d, 2) models. We choose $d = 0.2$ and 0.3. For $p = 1, q = 1$, we choose $\phi_1 = 0.5$ and $\psi_1 = 0.3$. For $p = 3, q = 2$, we set $(\phi_1, \phi_2, \phi_3) = (0.5, 0.3, 0.1)$ and $(\psi_1, \psi_2) = (0.4, 0.2)$. There are two distributions we used as the innovations to generate the time series. One is the stan-

dard normal distribution. The second one is a centered poisson distribution. The latter innovations are obtained by subtracting the mean and dividing the standard deviation of a random sample generated from a poisson distribution with mean 3. The reference sample size is chosen to be $m = 32000$. A discussion on the choice of reference sample size can be found in Section 3.4.3

To evaluate the ARL_0 performance, we generate 10000 sample paths for the test data using the same settings as the reference sample. For each sample path, a run length is recorded. Then the conditional ARL_0 is calculated by averaging the 10000 run lengths. This process is repeated for 100 times to give an approximate distribution of the conditional ARL_0. The desired ARL_0 is set to be 1000.

We apply the procedure described in Section 3.3 with $l = 4$ and $l = 5$ for $d = 0.2$ and $l = 5$ and $l = 6$ for $d = 0.3$. These choices of decomposition scales are discussed in Section 3.4.3. The k parameter for SS-CUSUM in our proposed procedure is chosen to be 0.2 or 0.3, as recommended by Li et al. (2013) to achieve robustness under different distributions. We also apply the Runger chart and Rank CUSUM chart as discussed previously. The k parameter for the Rank CUSUM chart is chosen to be 0.2. The results for fractional ARIMA (1, 0.2, 1) with standard normal innovations is displayed in Figure 3.3 (a). In the figure, we can see six boxplots representing the approximate distributions of conditional ARL_0 using different procedures.

From the figure we can see that our proposed wavelet-based method has a better control with most of the ARL_0 within 20% of the nominal level. The ARL_0 from the Runger chart and Rank CUSUM chart are around the nominal level, but the variation is

much larger than our proposed method. In fact, both of these two methods require more reference sample to achieve a similar ARL_0 control as our method. The decomposition scale also affect the ARL_0 control of our proposed method. A higher scale will result in a better control, due to the fact that higher scale of wavelet decomposition has a better decorrelation power. However, a lower scale of decomposition will have reduced detection time. Therefore, practitioners could choose the decomposition scale based on their priorities.

The result for the situation where data are generated from a fractional ARIMA (3, 0.2, 2) model with standard normal innovation is collected in Figure 3.3 (b). From the figure we can see significantly more variability in the conditional distribution of ARL_0 for the Runger chart and Rank CUSUM chart. This may be due to the fact that the reference sample is not large enough to fit the model well. As a result, the fitted fractional ARIMA models can be very different from the fractional ARIMA models that simulate the data. The misspecification of the fractional ARIMA model would result in a significant correlation presented in the estimated residuals, and ultimately affect the ARL_0 performance. On the opposite, our proposed procedure still appropriately controls the ARL_0 at the desired level. Certainly, the performance of control charts based on residual estimates from fractional ARIMA model fitting could be improved if a model close to the real model is used. However, it would be difficult to obtain the correct model. In real applications, the true model is seldomly known.

Figures 3.3 (c) and (d) present the results for fractional ARIMA (1, 0.2, 1) and (3, 0.2, 2) models with centered poisson innovations. The methods based on estimated residuals from fractional ARIMA models generally fail to control the ARL_0. This is due to

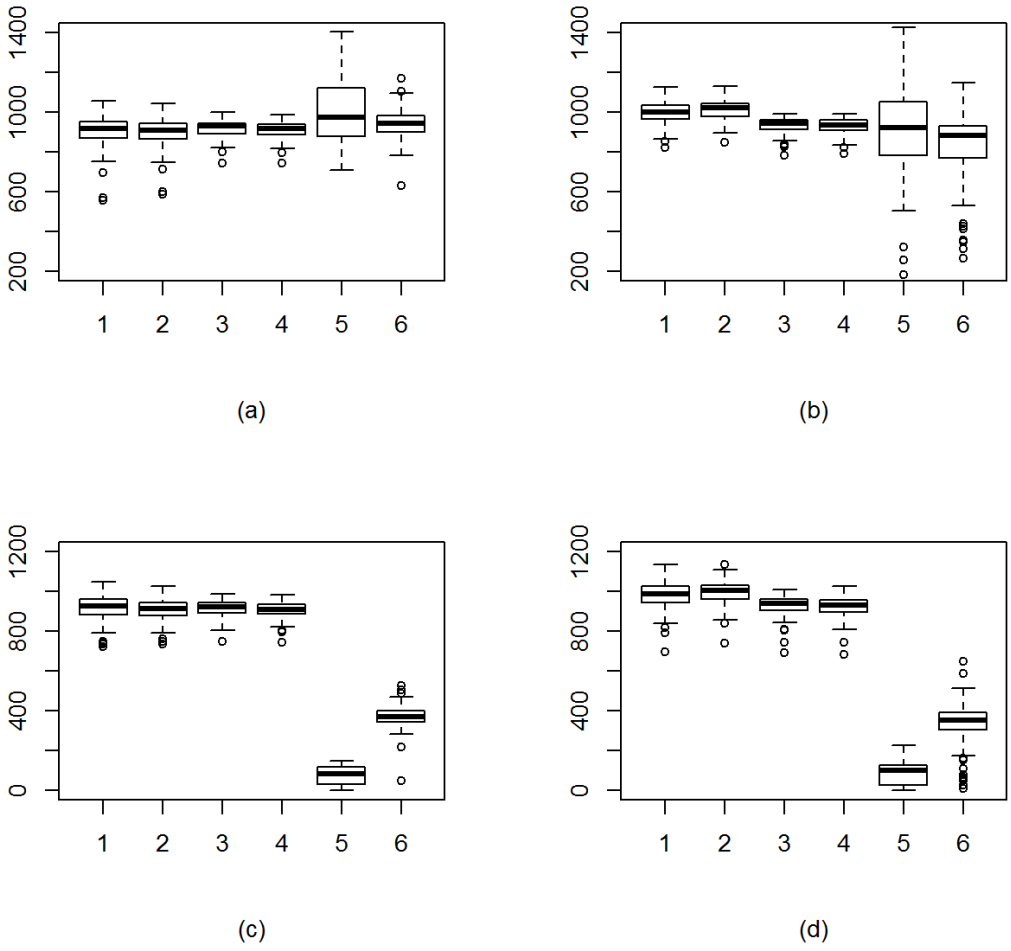


Figure 3.3: ARL_0 Performance for Data Generated from Fractional ARIMA (p,0.2,q) Models: (a) Normal innovation, $p = 1, q = 1$, (b) Normal innovation, $p = 3, q = 2$, (c) Poission innovation, $p = 1, q = 1$, and (d) Poisson innovation, $p = 3, q = 2$. The six boxplots, from left to right, represent the proposed method with $l = 4$ and $k = 0.2$, with $l = 4$ and $k = 0.3$, with $l = 5$ and $k = 0.2$, with $l = 5$ and $k = 0.3$, Runger chart, and Rank CUSUM chart, respectively.

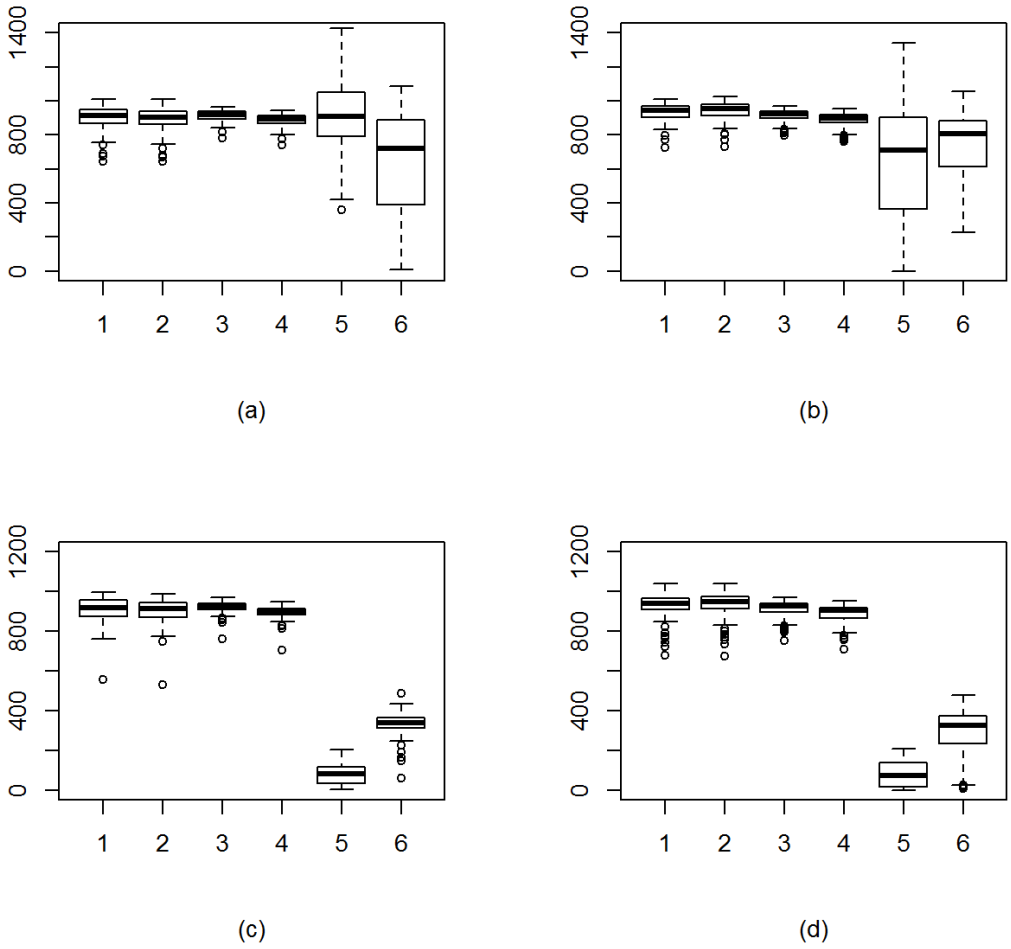


Figure 3.4: ARL_0 Performance for Data Generated from Fractional ARIMA (p,0.3,q) Models:(a) Normal innovation, $p = 1, q = 1$, (b) Normal innovation, $p = 3, q = 2$, (c) Poission innovation, $p = 1, q = 1$, and (d) Poisson innovation, $p = 3, q = 2$. The six boxplots, from left to right, represent the proposed method with l=5 and k=0.2, with l=5 and k=0.3, with l=6 and k=0.2, with l=6 and k=0.3, Runger chart, and Rank CUSUM chart, respectively.

the fact that the parameter estimates of the fractional ARIMA models are based on the ML methods, which requires normality assumption. The deviation from normality affects the parameter estimates, especially the estimate in the differencing parameter d. Therefore, the estimated residuals are not approximately uncorrelated. On the other hand, our proposed method still controls the ARL_0 at the nominal level, with almost all the ARL_0 within 20% of the desired level.

Results for the cases described above but with $d = 0.3$ are displayed in Figure 3.4. We can see they are very similar to what we observed in fractional ARIMA (p, 0.2, q) case.

Another data generating scheme is considered in our simulation studies to mimic the data we observed in our real example illustrated in Figure 3.1 (a). We first assume the data in each week are independent and have the same pattern with fixed hourly means which can be estimated from the real data. There are 168 hours per week, so we denote the hourly means by $\{\mu_t\}_{t=1}^{168}$. Then correlated random effects $\boldsymbol{\delta} = (\delta_1, \cdots, \delta_{168})'$ following a multivariate normal distribution with mean $\mathbf{0}$ and variance $\boldsymbol{\Sigma}$ are added to μ_t. The correlation between δ_i and δ_j is ρ^{j-i} for all $j > i, i = 1, \cdots, 167$. We choose $\rho = 0.5$ or 0.9. For each hour there are 12 observations. Hence, there are 2016 observations in a week. If we denote these observations by $\{Y_k\}_{k=1}^{2016}$, and for $k = 12(t-1)+1, \cdots, 12(t-1)+12$, $t = 1, \cdots, 168$ we set

$$E(Y_k|\delta_t) = \mu_t + \delta_t, \tag{3.5}$$

then we can generate data Y_k from a normal distribution with mean defined in (3.5) and standard deviation 1, or a poisson distribution with mean defined in (3.5). This data generating scheme coincides with the (Generalized) Linear Mixed Model ((G)LMM). We

can repeat the previous process to generate observations for multiple weeks.

To demonstrate the goodness-of-fit of the model to the real data, we simulated 1000 weeks of data using the above data generating scheme with Poisson distribution and $\rho = 0.9$. One can see Figure 3.5 for the comparison between the simulated data and one week of the true data. From the figure, we can see that the true data is well contained within our simulated data. There are only a few outliers. We also examined other weeks of true data, and found similar results. Hence, we could conclude that our data generating scheme could describe the real data well.

We generate 16 weeks of data using this scheme as the reference samples, so that the effective reference sample depth is approximately 32000. Similarly as the previous cases, for each reference sample generated, we use 10000 test samples to evaluate the conditional ARL_0. This process is repeated 100 times to obtain the approximate distribution of the conditional ARL_0. We apply our proposed methods, Runger chart, and Rank CUSUM chart. The residuals for Runger chart and Rank CUSUM charts are obtained by fitting fractional ARIMA models estimated from the data. The results are shown in Figure 3.6. From the figures, we can see that our proposed method is controlling the ARL_0 properly. The Runger chart and Rank CUSUM chart fail to control the ARL_0. The Rank CUSUM chart has a much smaller ARL_0 than the desired one. The Runger chart has a well controlled median ARL_0. However, the large variation in the conditional ARL_0 distribution makes its practical usefulness questionable. The reason for both of the residual-based methods failing to control the ARL_0 may be due to the fact that the fractional ARIMA model could not decorrelate the data thoroughly, since the data generating scheme is different from the

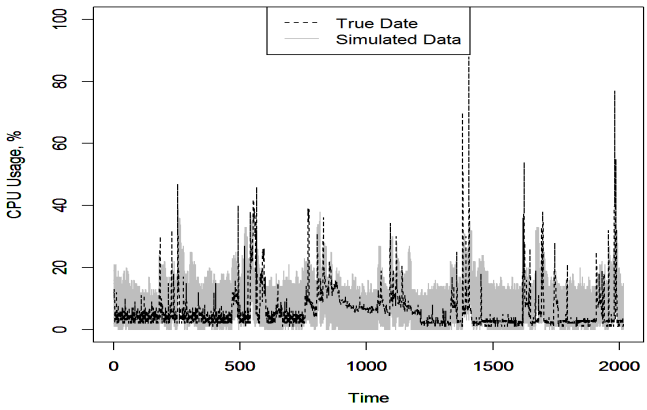


Figure 3.5: Goodness-of-fit of the GLMM Model to the Real Data

fractional ARIMA model, which would result in a model misspecification.

3.4.2 ARL_1 Performance Evaluation

From Section 3.4.1, we can see that both Runger chart and Rank CUSUM chart using estimated residuals from fractional ARIMA model generally do not control the ARL_0 well. On the other hand, our proposed method is working properly under all simulation settings. In this section, we will compare the ARL_1 performance of these three control charts. Note that we only include the ARL_1 performance for each control chart when it has appropriate ARL_0 control.

We simulate data using the same two schemes as in Section 3.4.1. For test data, we use a step change in the mean. That is, we add a change δ to the test data. The magnitude of δ is $(0.6, 1.2, 1.8, 2.4, 3)$. We also assume that the change will persist until the alarm is triggered. We still use 10000 sample paths to find the ARL at each value of

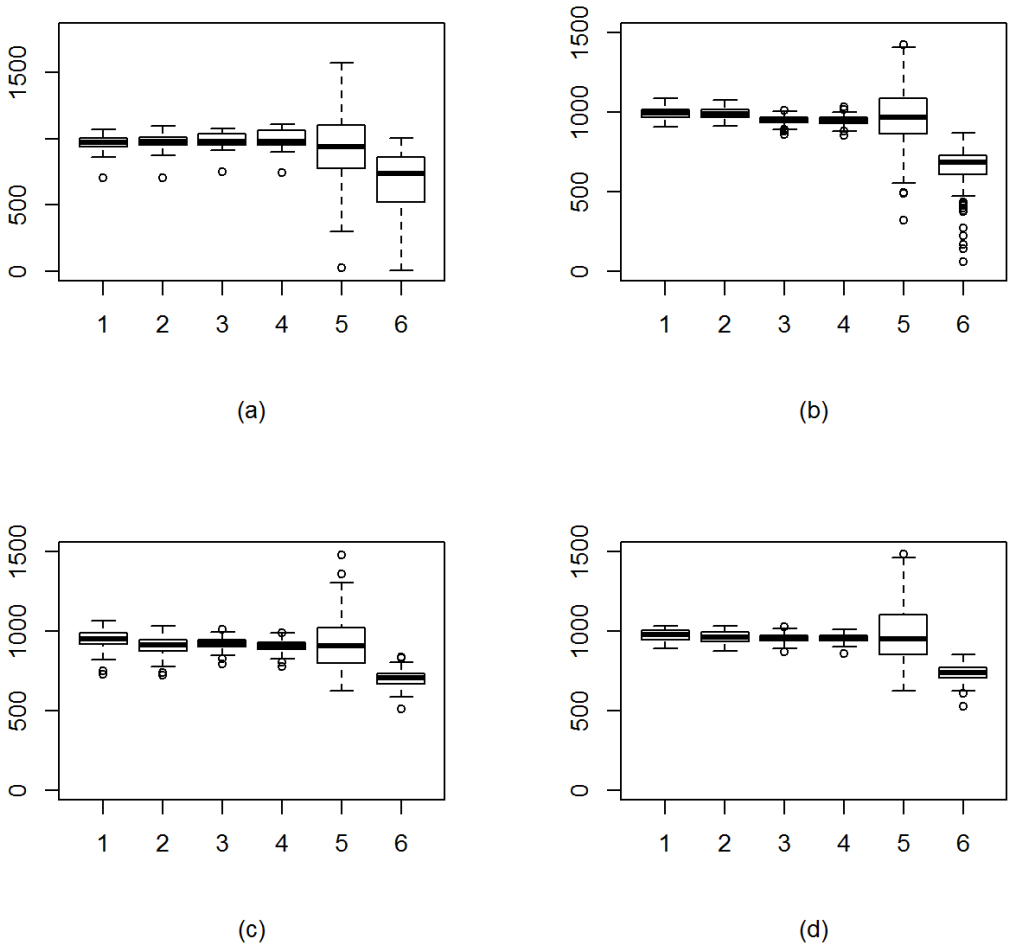


Figure 3.6: ARL_0 Performance for Data Generated from GLMM Model: (a) Normal $\rho = 0.9$, (b) Normal $\rho = 0.5$, (c) Poisson $\rho = 0.9$, and (d) Possion $\rho = 0.5$. The six boxplots are in the same order as in Figure 3.3

δ. Then the process is repeated 100 times, and an average of the ARL estimates is taken which represents the unconditional ARL at each value of δ.

The results for data simulated from fractional ARIMA model with $d = 0.2$ are shown in Table 3.1. From the table we can see that for the data simulated with normal innovations, the Rank CUSUM has the best detection power among all the three methods, with a slight advantage over our proposed method. Both of these two methods outperform the Runger chart by a significant margin. However, if we take the ARL_0 control into consideration, with a much smaller variation in the conditional ARL_0 distribution, our proposed scheme should still be the method of choice. When it comes to the poisson innovation scenario, the two residual based methods totally fail to control the ARL_0. Hence there is no fair ARL_1 comparison between the three methods. However, we can see that our proposed method has a decent performance in the sense that, as δ increases, the ARL_1 quickly decreases. The results for data simulated from fractional ARIMA model with $d = 0.3$ are similar to what we saw above, and are collected in Table 3.2. The Rank CUSUM chart could not control the ARL_0 well, so we exclude their ARL_1 performance in the comparison. From the results, we can see that our proposed method dominates its competitor in the ARL_1 performance. The results for data simulated from GLMM model are shown in Table 3.3. From the table we can see that our proposed method has a superior ARL_1 performance than its competitors when the ARL_0 is properly controlled, especially if we choose a smaller decomposition level. Therefore, from the simulation results shown above, our method not only controls ARL_0 better, but also has a similar, if not better, detection power than its competitors under most of the simulation settings.

3.4.3 Implementation Issues

In this section, we discuss some implementation issues in our proposed method. The first problem is how to choose the scale of wavelet decomposition. We use a method based on a multivariate run test proposed by Paindaveine (2009). The method is detailed as follows.

We use the procedure described in Section 3.3 to obtain the residual vectors $\mathbf{Y}_k$ from the reference sample $(y_1, \cdots, y_N)$ at scale l. Then the multivariate run test is applied to the sample $\{\mathbf{Y}_k\}_{k=1}^{2^{n-l}}$ to determine if there is correlation between these vectors. The run test is conducted using every r multivariate observations, $r = 1, 2, \cdots q, q \leq 2^{n-1}$, respectively, and we denote the test statistic by Q_r, which is given as follows:

$$Q_r = p^2 \frac{1}{n' - 1} \sum_{s,t=2}^{n'} \mathbf{U}'_s \mathbf{U}_t \mathbf{U}'_{s-1} \mathbf{U}_{t-1} \tag{3.6}$$

where p is the dimension of the vector, n' is the number of the vectors, and $\mathbf{U}_k$ is the spatial sign of $\mathbf{Y}_k^*$ discussed in Section 2.2.3. If $Q_r > \chi^2_{p^2,1-\alpha}$, then there is correlation between the multivariate observations at lag r at significance level α. Therefore, Q_r can be seen as the multivariate counterpart of ACF of the univariate time series at lag r. A plot of Q_r versus the lag r can be used to determine if there is autocorrelation between the multivariate observations. We repeat this process for any scale $l = 2, 3, \cdots$. Then the scale of decomposition is chosen to be the smallest scale where its test statistics at all lags are insignificant (or near insignificant). Figure 3.7 is an example. The data are generated from the same fractional ARIMA (1, 0.2, 1) model described in Section 3.4.1. The significance level for the run test is set to be $\alpha = 0.05$. From the figures, we can see that at scale 4,

the statistics at all lags are almost insignificant. This result actually validates our choice of decomposition scale in our simulation study.

Another issue we discuss in this section is the reference sample size for our method to performe appropriately, especially for the ARL_0 to be controlled at the desired level. A simulation study is conducted to investigate the effect of reference sample size on the ARL_0 control. We use the fractional ARIMA(1, 0.2, 1) model described in Section 3.4.1 with standard normal innovations as an example. The reference sample sizes are set to be $n = 16000, 32000$ and 64000. The nominal ARL_0 is set to be 1000 as before. Then we apply our proposed method as well as the two competitors to evaluate the ARL_0 performance at each reference sample size as described previously. The results are shown in Figure 3.8. It can be seen that the ARL_0 performance for all the methods gets better as the reference sample size increases. For our proposed method, this is due to the fact that the SSCUSUM control chart involved in our procedure is asympototically distribution free. A larger sample size is essential for the control chart to maintain a desired level of ARL_0. For the two residual-based methods, the larger sample size would help in the fractional ARIMA model fitting, which would lead to approximately uncorrelated residuals.

At the same time, we also discover the phenomenon that the ARL_0 control of our proposed procedure is affected by the value of the target ARL_0. For a smaller target ARL_0, a smaller sample size is required to achieve appropriate control, while a larger target ARL_0 is in need for a larger reference sample size. The reason is that our procedure only approximately decorrelate the signal. Therefore, after the wavelet transformation and time series model fitting, there might still be correlation remaining in the multivariate vectors.

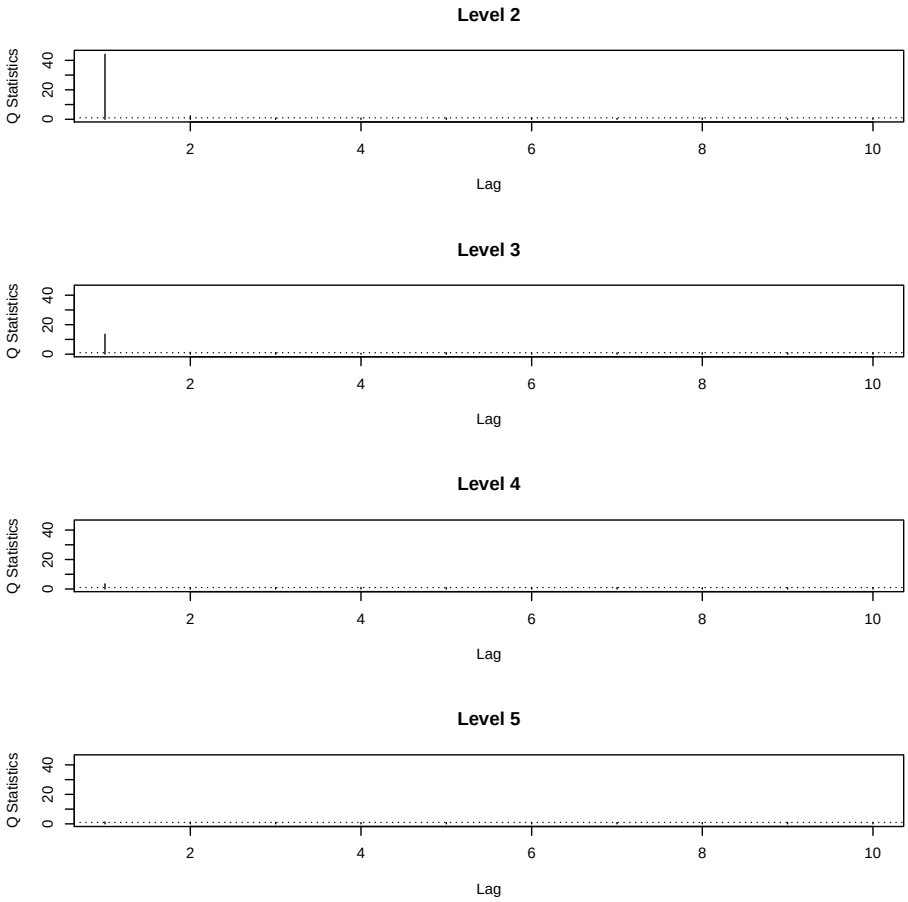


Figure 3.7: Multivariate Run Test for Decomposition Scale Determination.

For smaller target ARL_0, the impact of remaining correlation is not significant, whereas the correlation may accumulate and affect the ARL significantly for a longer series of data (i.e. larger target ARL_0). Therefore, if the desired ARL_0 is larger, the practitioners will need a larger reference sample. Our experiments with $ARL_0 = 1000$ suggest reference sample sizes on the order of 30000 to 50000 is needed.

The third issue we discuss is the missing value treatment. It is very common for the real data to contain a substantial portion of missing values. The missing values in the reference sample could be addressed by using the existing information in the sample. Some commonly used techniques are mean imputation, or imputation based on some parametric models. As long as the practitioners consider the assumption for the imputation justifiable, any imputation method could be applied. For the test sample, where the mean could potentially shift, one could not use the information from reference sample for the imputation purpose. We assume that for the network data, the means for consecutive observations are similar. Hence, we could substitute the missing values with the average of their non-missing neighbors, if there are not many consecutive missing values. If there is a block of observations missing (for instance 4 or more consecutive values are missing), then we would suspend our monitoring process until we have sufficient observations to resume. Before we could resume the monitoring process, it is necessary that we have a burn-in period that allows us to find reliable residual estimates from the ARIMA(1, 1, 1) model for the scaling coefficients. A longer burn-in period may result in better residual estimates, while on the other hand, delay the out of control detection.

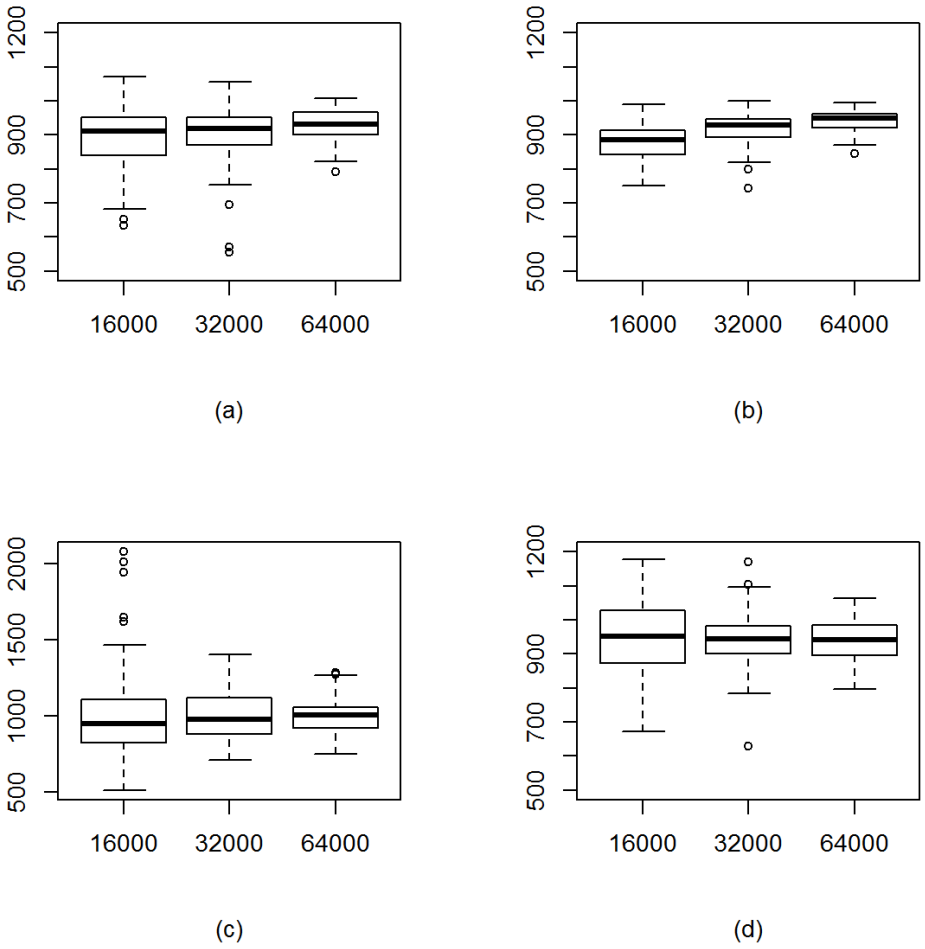


Figure 3.8: Sample Size Comparison for: (a) Wavelet SSCUSUM Chart with l=4, (b) Wavelet SSCUSUM Chart with l=5, (c) Runger Chart, and (d) Rank CUSUM Chart

3.5 Real Data Example

In this section, we use the data illustrated in Figure 3.1 (a) to demonstrate the implementation of our proposed method, and compare it with the Runger and Rank CUSUM charts that utilize the estimated residuals from fractional ARIMA models. The data consist of the CPU usage of a network server from May 31st to November 21st in 2010, with measurements recorded every 5 minutes. During October 11th to November 7th, there are four weeks of data missing. Therefore, in total, there are 21 weeks of data available. We take the data from the first 18 consecutive weeks (May 31st to Oct 3rd) as the reference sample to calibrate our procedure as described in Section 3.3. The last three weeks (week starting from Oct 4th to Oct 10th and two consecutive weeks starting from Nov 8th to Nov 21st) are treated as the test sample. From Figure 3.1 (b) and (c), we can see that the data deviate from normality, and there is LRD presented within the observations. Therefore, our proposed wavelet-based SS-CUSUM control chart is appropriate to monitor the process. We also include Runger chart and Rank CUSUM chart for performance comparison.

Before we could use the proposed method to monitor the process, the massive number of missing values is treated as follows. For the reference sample, we assume the observations within the same hour during the week share the same mean, we could substitute the missing values with the corresponding hourly means. And for the test sample, we use the method discussed in Section 3.4.3. For this real application, during the three monitoring weeks, the first week has 1 missing value, and the third week has 4 non-consecutive missing values. Hence we just use the averages of the neighboring values to impute the missing values in those two weeks. However, between the first and second week in our test sample, there

are four weeks of data missing. And even for the second week itself, we have 96 consecutive missing values at the beginning of the week. Therefore, we suspend our monitoring process during this time, and use the observations from the rest of the second week as our burn-in period. We use this long burn-in period just for more accurate residual estimates. Our monitoring process resumes at the beginning of the third week for the test sample.

We use an $\text{ARL}_0 = 1000$ for the three control charts, which is equivalent to approximately 2 false alarms per week. We use the method described in Section 3.4.3 to determine the decomposition scale. The result is shown in Figure 3.9. From the figure, a 5-scale wavelet decomposition should be appropriate. For the SS-CUSUM scheme in our proposed method, we pick $k = 0.2$ in order to achieve robustness to non-normal distribution. The control chart is shown in Figure 3.10. From the figure, we can see that our proposed method has three alarms, the Runger Chart has two alarms, and the Rank CUSUM chart has numerous alarms. After checking the general trend, mean and spread of both the reference sample and the test sample, we could not see any anomaly presented in the test data. Therefore, our proposed method and the Runger chart has appropriate control on ARL_0. The Rank CUSUM chart, on the other hand has too many false alarms. This result is consistent with our simulation result, where Rank CUSUM chart tends to have more frequent false alarms than the other two methods.

To demonstrate the anomaly detection power of our method, we add a persistent location shift of $\delta = 15$ to the second week of the test sample. This choice of value for δ is due to the fact that the overall standard deviation of the data is 10. The control chart are reconstructed based on the new data, and shown in Figure 3.11. Since Rank CUSUM chart

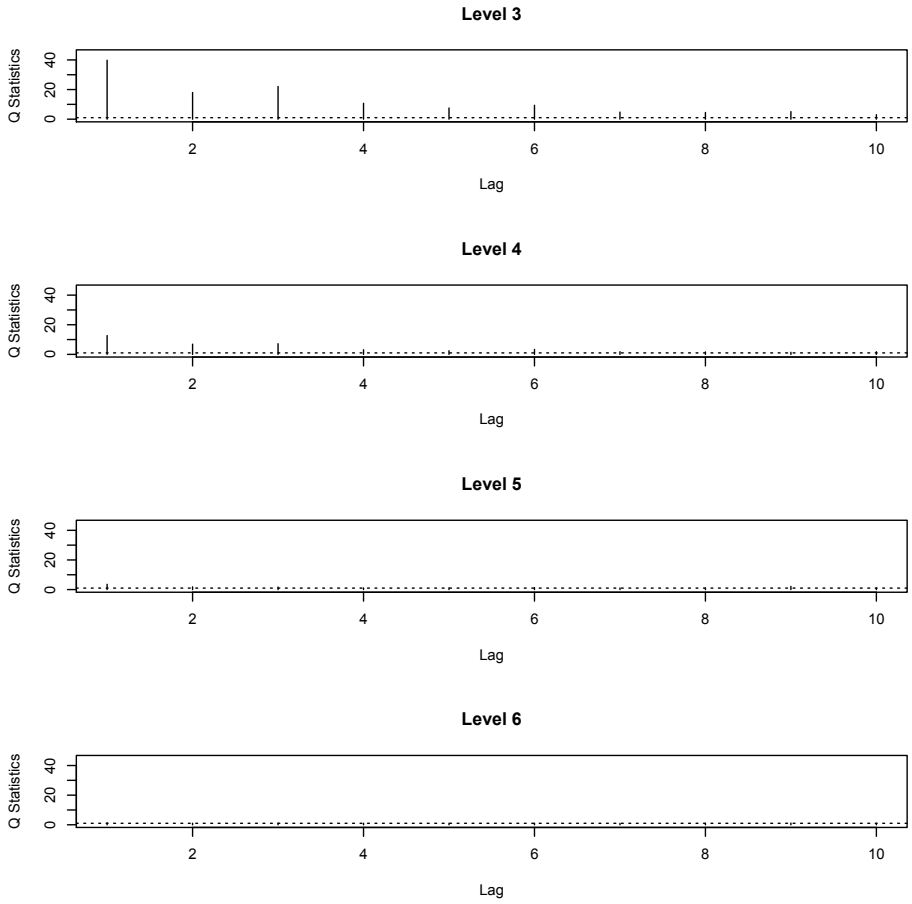


Figure 3.9: Multivariate Run Test for Decomposition Scale Determination on Real Data.

fails in the ARL_0 control, we do not include it in this power comparison. For the remaining two control charts, we terminate the monitoring process as soon as an alarm is triggered in the second week. We can see from the figure that the location shift was picked up by our control chart after 129 observations in the second week, compared to the original control chart, which has a false alarm at 1505 observation. This shows that our procedure could identify the abnormal activity in the network data within a short time. On the other hand, the Runger Chart has no power in detecting this location shift.

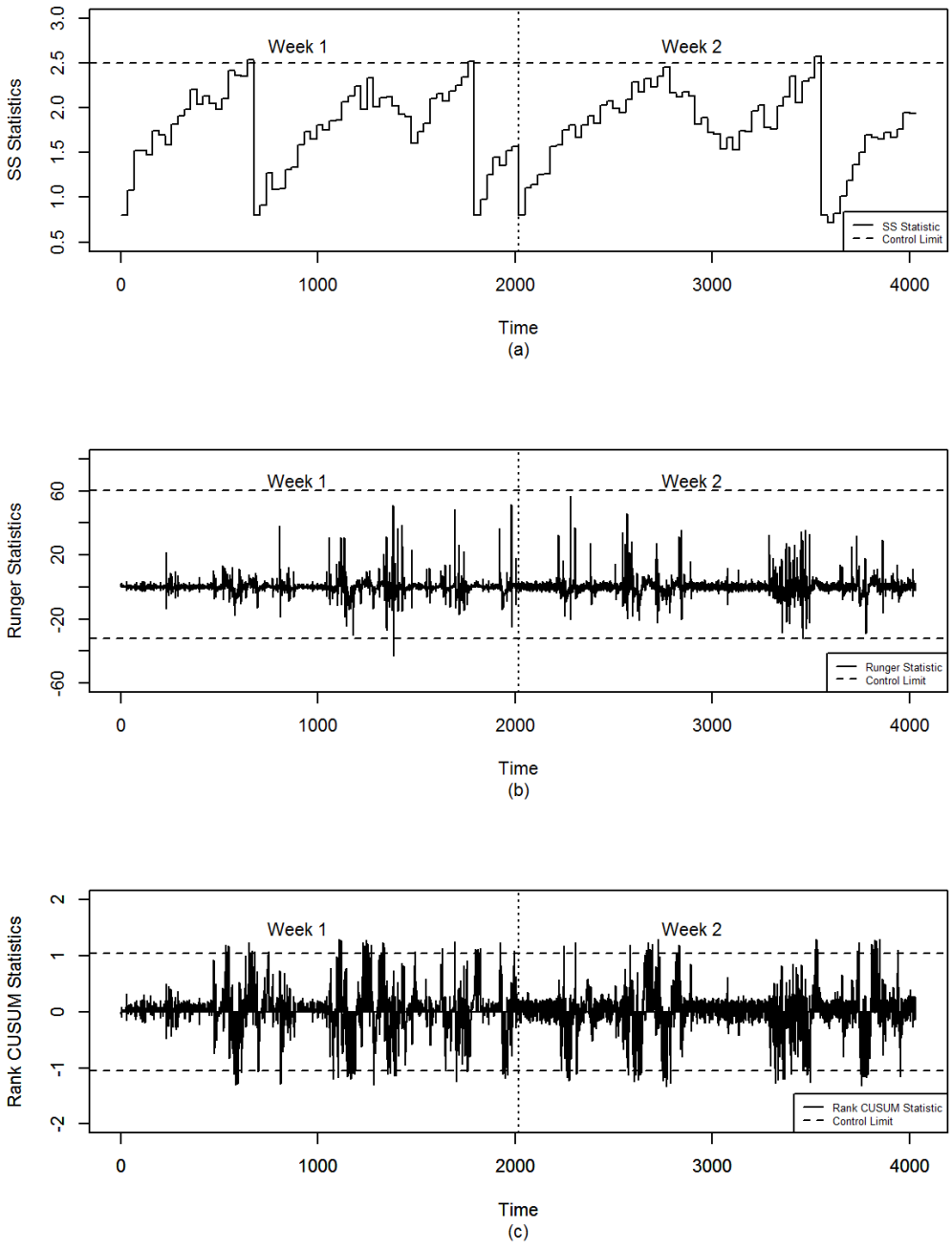


Figure 3.10: Real CPU Usage Monitoring with (a) Wavelet-Based SS-CUSUM control chart, (b) Runger Control Chart, and (c) Rank CUSUM Control Chart.

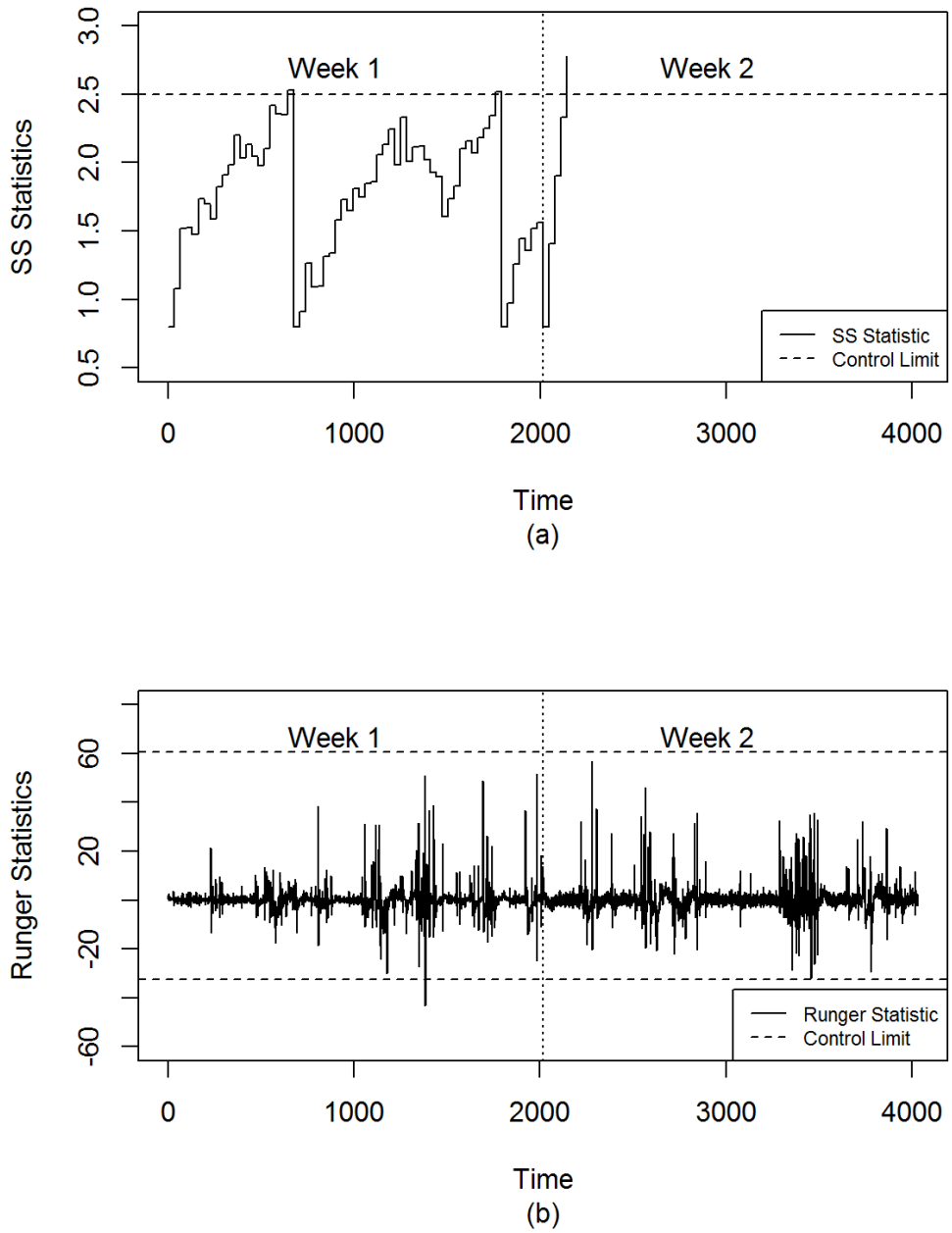


Figure 3.11: Real CPU Usage Monitoring with location shift δ, with (a) Wavelet-Based SS-CUSUM control chart and (b) Runger Control Chart

Table 3.1: ARL_1 Comparison for Fractional ARIMA (d=0.2)

Fractional ARIMA (1, 0.2, 1) with Normal Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	577.54	237.46	135.97	106.18	92.40
Wavelet l=4, k=0.3	594.40	226.45	113.43	86.21	74.62
Wavelet l=5, k=0.2	658.81	355.31	232.02	184.47	162.51
Wavelet l=5, k=0.3	645.68	314.53	190.16	149.11	132.29
Runger	996.01	947.94	848.94	692.65	495.00
Rank CUSUM	668.81	466.83	251.57	112.63	37.16
Fractional ARIMA (3, 0.2, 2) with Normal Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	923.18	757.52	568.57	410.96	298.89
Wavelet l=4, k=0.3	948.19	797.54	609.86	437.32	306.00
Wavelet l=5, k=0.2	876.77	747.74	600.79	472.50	377.13
Wavelet l=5, k=0.3	874.60	743.47	587.84	446.87	341.10
Runger	876.38	829.11	752.61	651.93	535.51
Rank CUSUM	609.25	494.92	375.69	306.37	236.72
Fractional ARIMA (1, 0.2, 1) with Poisson Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	593.45	235.65	133.40	104.41	91.16
Wavelet l=4, k=0.3	608.92	223.96	110.29	84.52	73.43
Wavelet l=5, k=0.2	654.63	347.14	227.15	181.56	161.05
Wavelet l=5, k=0.3	640.57	304.70	185.09	146.46	130.89
Fractional ARIMA (3, 0.2, 2) with Poisson Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	932.28	776.00	586.71	421.24	302.59
Wavelet l=4, k=0.3	951.55	812.16	625.84	447.78	309.67
Wavelet l=5, k=0.2	882.57	762.30	619.38	491.40	392.77
Wavelet l=5, k=0.3	878.52	756.93	606.65	466.80	358.04

Table 3.2: ARL_1 Comparison for Fractional ARIMA (d=0.3)

Fractional ARIMA (1, 0.3, 1) with Normal Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=5, k=0.2	822.73	647.71	464.09	331.49	252.66
Wavelet l=5, k=0.3	815.16	630.57	432.38	289.81	209.45
Wavelet l=6, k=0.2	859.00	736.29	601.88	492.18	413.69
Wavelet l=6, k=0.3	840.37	701.29	550.89	433.44	353.47
Runger	908.31	858.30	772.00	648.47	499.26
Fractional ARIMA (3, 0.3, 2) with Normal Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=5, k=0.2	927.50	899.11	851.17	789.20	718.94
Wavelet l=5, k=0.3	934.31	905.93	857.86	794.62	720.79
Wavelet l=6, k=0.2	912.41	894.13	862.05	819.11	768.97
Wavelet l=6, k=0.3	885.51	865.51	829.87	782.15	726.39
Fractional ARIMA (1, 0.3, 1) with Poisson Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=5, k=0.2	851.85	673.82	473.40	329.89	250.34
Wavelet l=5, k=0.3	844.12	658.90	442.47	286.39	205.50
Wavelet l=6, k=0.2	870.95	742.81	601.61	487.81	408.11
Wavelet l=6, k=0.3	839.57	697.38	540.81	420.50	342.01
Fractional ARIMA (3, 0.3, 2) with Poisson Innovation					
	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=5, k=0.2	921.69	894.23	847.83	787.68	718.86
Wavelet l=5, k=0.3	929.18	901.66	854.50	792.21	719.86
Wavelet l=6, k=0.2	904.04	880.18	843.60	797.15	745.55
Wavelet l=6, k=0.3	876.88	850.57	810.18	759.16	701.36

Table 3.3: ARL_1 Comparison for GLMM

Normal with $\rho = 0.9$

	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	915.47	780.12	619.93	476.16	364.79
Wavelet l=4, k=0.3	931.20	811.65	659.32	509.00	382.21
Wavelet l=5, k=0.2	965.86	899.93	820.14	746.38	686.24
Wavelet l=5, k=0.3	971.10	907.93	826.20	746.53	679.39
Runger	922.39	874.99	790.91	677.05	544.25

Normal with $\rho = 0.5$

	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	770.66	445.77	263.60	182.99	144.73
Wavelet l=4, k=0.3	801.77	470.99	257.34	161.99	121.64
Wavelet l=5, k=0.2	767.34	502.91	345.88	265.06	220.49
Wavelet l=5, k=0.3	766.31	478.39	304.45	222.37	181.43
Runger	956.95	921.20	857.73	767.37	653.97

Poisson with $\rho = 0.9$

	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	886.43	713.60	490.34	297.35	178.23
Wavelet l=4, k=0.3	849.65	672.37	442.85	252.28	141.63
Wavelet l=5, k=0.2	803.70	584.41	398.94	277.15	211.56
Wavelet l=5, k=0.3	789.93	552.14	348.98	224.66	167.58
Runger	922.64	918.94	913.34	905.55	893.82

Poisson with $\rho = 0.5$

	Location Shift δ				
Procedure	0.6	1.2	1.8	2.4	3
Wavelet l=4, k=0.2	546.42	218.08	137.73	111.79	98.60
Wavelet l=4, k=0.3	571.87	202.70	116.87	93.41	81.87
Wavelet l=5, k=0.2	629.84	342.72	226.81	180.80	161.61
Wavelet l=5, k=0.3	619.01	294.78	183.22	148.44	132.06
Runger	971.57	968.09	962.77	955.43	944.71

Chapter 4

Sequential Classifier for Longitudinal Data

4.1 Introduction

Statistical classification is the problem of assigning an observation to one of a set of populations, based on a training set of observations whose population membership is known. One applicable area of statistical classification is disease diagnosis. Many diseases can be detected based on a patients' change in levels of certain clinical characteristics (e.g. biomarkers). Commonly, these characteristics would be repeatedly measured throughout hospitalization, so that disease diagnosis can be made according to patients' profile change. Consider, for example, the data illustrated in Figure 4.1, which consists of a subset of biomarker profiles in a severe sepsis diagnosis study that includes 990 patients. Within a 72 hour window, as many as 7 blood draws were taken from each patient at time points

$t = 0, 3, 6, 12, 24, 48,$ and 72 hours, and the biomarker level at each time point was recorded. The goal of the study is to use the evolving biomarker profiles of the patients to build a classifier which can separate sepsis patients from severe sepsis patients.

Classifying longitudinal data presents more challenges than classifying regular multivariate data. First, there are usually many missing values in the longitudinal data. Second, the time points at which the repeated measures are observed may vary between subjects. To overcome these difficulties several procedures have been proposed for classifying longitudinal data (see Verbeke and Lesaffre (1996), Marshall and Barón (2000), James and Hastie (2001), Luts et al. (2012) for example). These procedures were developed to classify the longitudinal data based on the complete profile of the data. In the example of our sepsis diagnosis study, those procedures will classify the subject at the end of 72 hours. However, in the sepsis diagnosis study, it is extremely desirable to classify the subject as early as possible, since severe sepsis is a life-threatening condition and an earlier detection would result in a lower mortality rate. If we directly apply the existing longitudinal data classifiers at earlier time points, the performance of the classifiers may be compromised, since some subjects may not have enough information to be correctly classified at earlier time points. On the other hand, some subjects may indeed show signs of belonging to one of the class groups based on data from early time points, and therefore could be accurately classified sooner.

This observation motivates us to consider a sequential classifier, which will sequentially evaluate the subject and decide whether to classify at each time point. To develop such a sequential classifier, we adopt the neutral zone classifier framework. Different from

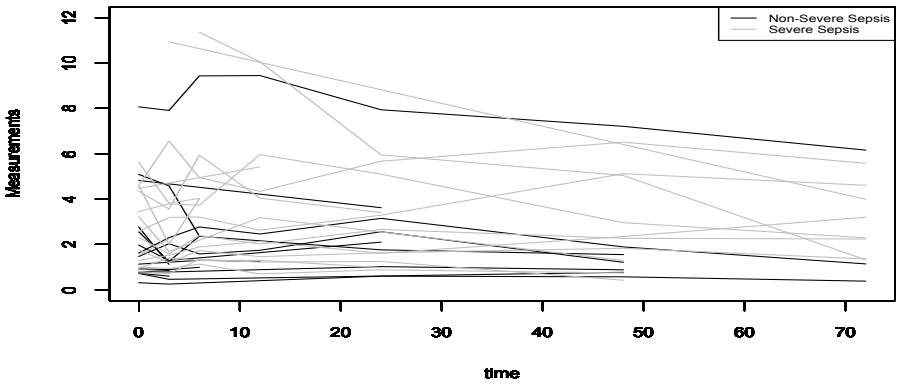


Figure 4.1: Biomarker measurements of selected 40 patients with Severe Sepsis symptoms. The black lines represent non-severe sepsis group and the grey lines represent severe sepsis group

the traditional classifiers which always assign the subject to one of the class groups, a neutral zone classifier allows to assign a neutral classification (not belonging to any of the class groups) when there is not enough confidence to classify to any of the class groups. Therefore, for our sequential classifier, at each time point (starting from the first time point), we evaluate the confidence in classifying the subject to each of the class groups. If we do not have enough confidence in making a classification, we assign a neutral classification, and continue to collect another measurement at the next time point and re-evaluate how confident we are given the new measurement. This process continues until there is enough confidence of making a classification or the last time point where data can be collected is reached. Using this sequential procedure allows early decisions on subjects which are easier to classify, and also delays decisions on subjects that are difficult to classify. As a result,

the proposed sequential classifier reduces the overall cost when the cost of time is taken into account.

The rest of the chapter is organized as follows. In Section 4.2, we present our sequential classifier. Since our sequential classifier is developed based on the neutral zone classifier, we first discuss the neutral zone classifier in Section 4.2.1. To implement the neutral zone classifier and make it a sequential procedure for classifying longitudinal data, calculating the posterior probability of the subject belonging to one of the class groups at any given time point is a key step. In Section 4.2.2, we describe calculating such probabilities based on a combined logistic regression model and mixed effects model. In Section 4.2.3, we detail the overall sequential classification procedure. A simulation study for evaluating the performance of our sequential classifier is given in Section 4.3. In Section 4.4 we revisit the sepsis data to demonstrate an application of our proposed sequential classifier.

4.2 Sequential Classifier for Longitudinal Data

4.2.1 Neutral Zone Classifier

Different from traditional classifiers, neutral zone classifiers can assign a neutral classification to a subject when there is not enough confidence in assigning it to any of the class groups. Since most of the disease diagnosis applications concern two class groups, we focus on two-class classification problem throughout the chapter. For the two-class classification problem, there are three possible classification outcomes for a neutral zone classifier: group 0 or group 1 or neutral (labeled by N). Jeske et al. (2007), Yu et al. (2010)

and Benecke et al. (2013) derived a neutral zone classifier based on the minimum cost criterion. Denote the posterior probabilities of the new subject belonging to group 1 and group 0 by $p_1(\boldsymbol{y})$ and $p_0(\boldsymbol{y})$, respectively. Assume that a cost structure is given in Table 4.1, where C_{ij} is the cost of assigning the subject in group j to group i, $i \neq j$ and $i, j \in \{0, 1\}$, and C_N is the cost of assigning the subject the neutral classification. Define $\rho_{ij} = C_{ij}/C_N$. Benecke et al. (2013) show that, if $\frac{1}{\rho_{01}} + \frac{1}{\rho_{10}} < 1$, the minimum cost classifier is

$$\begin{cases} \text{classify to group 0} & \text{if} \quad p_1(\boldsymbol{y}) \leq \frac{1}{\rho_{01}} \\ \text{classify to group 1} & \text{if} \quad p_1(\boldsymbol{y}) \geq 1 - \frac{1}{\rho_{10}} \\ \text{classify as N} & \text{if} \quad \frac{1}{\rho_{01}} \leq p_1(\boldsymbol{y}) \leq 1 - \frac{1}{\rho_{10}} \end{cases} \tag{4.1}$$

and if $\frac{1}{\rho_{01}} + \frac{1}{\rho_{10}} \geq 1$, the minimum cost classifier is

$$\begin{cases} \text{classify to group 0} & \text{if} \quad p_1(\boldsymbol{y}) \leq \frac{\rho_{10}}{\rho_{01}+\rho_{10}} \\ \text{classify to group 1} & \text{if} \quad p_1(\boldsymbol{y}) \geq \frac{\rho_{10}}{\rho_{01}+\rho_{10}}. \end{cases} \tag{4.2}$$

The classifier given in (4.2) is the traditional minimum cost classifier. In real applications, C_{ij} is usually much larger than C_N, therefore we assume that $\frac{1}{\rho_{01}} + \frac{1}{\rho_{10}} < 1$ holds throughout the chapter.

Table 4.1: Cost Structure of Two Class Neutral Zone Classifier

True Label	Predicted Label		
	0	1	N
0	0	C_{10}	C_N
1	C_{01}	0	C_N

Note that $p_0(\boldsymbol{y}) = 1 - p_1(\boldsymbol{y})$, and $p_0(\boldsymbol{y})$ and $p_1(\boldsymbol{y})$ can be used to measure the confidence in assigning the subject to group 0 and 1, respectively. From (4.1), we can see that, only if $p_0(\boldsymbol{y})$ or $p_1(\boldsymbol{y})$ is large enough (i.e. only if we have enough confidence classifying) will the subject be assigned to one of the two class groups. Otherwise, the subject will be given a neutral classification.

The above neutral zone classifier provides a framework for classifying subjects with longitudinal data sequentially. Given the cost structure in Table 4.1, at each time point (starting from the first time point), we can classify the subject to group 0 or group 1 if $p_1(\mathbf{y}) \leq \frac{1}{\rho_{01}}$ or $p_1(\mathbf{y}) \geq 1 - \frac{1}{\rho_{10}}$. If $\frac{1}{\rho_{01}} \leq p_1(\boldsymbol{y}) \leq 1 - \frac{1}{\rho_{10}}$, we will assign the subject to N and wait until the next time point to collect another measurement for this subject. With this new measurement, we will evaluate the subject again based on the updated $p_1(\boldsymbol{y})$ and a decision of classifying into group 0 or 1 or N will be made accordingly. We note that it would be easy, and perhaps useful, to modify the procedure to allow the cost structure in Table 4.1 to vary with time.

To implement the above sequential procedure, it is necessary to evaluate $p_1(\boldsymbol{y})$, the posterior probability of the subject belonging to group 1 given the repeated measurements $\boldsymbol{y}$ observed up to the current time point. Since we have two class groups in the training sample, a logistic regression (LR) model can be used to fit the data and the prediction from the LR model can then be considered as the posterior probability of the subject belonging to one of the class groups. However, in longitudinal data, the time points at which the repeated measures are observed usually vary between subjects. Even when the subjects are measured synchronously, the number of available measurements for each subject will usually

vary. Therefore, it is difficult to directly use the repeated measurements $\boldsymbol{y}$ as features in the LR model. To overcome this difficulty, we propose fitting a mixed effects model first to the longitudinal data, and extract the subject-specific random effects from the fitted mixed effects model and use them as features in the LR model. In the following we describe this method, which we refer to as mixed effects model based logistic regression method.

4.2.2 Mixed Effects Model Based Logistic Regression

Mixed effects models are widely used for modeling longitudinal data in the literature, since they are capable of handling asynchronous measurements and providing robust inference under missing at random (MAR) data patterns which are common in longitudinal studies. In the literature, there are two popular mixed effects models: linear mixed effects(LME) models (Harville(1976, 1977)) and nonparametric mixed effects (NME) models (see Wu and Zhang (2006) for examples). Our classification procedure can be built with either type of mixed effects model. For ease of exposition, we first use LME model to demonstrate how the method works. At the end of this section, we will extend it from LME model to NME model.

Suppose that there are m subjects in a randomly sampled training data set. We denote the training data set by (y_{ij}, t_{ij}, U_i), $i = 1, ..., m$, $j = 1, ..., n_i$, where y_{ij} is the jth measurement for the ith subject, t_{ij} is the time when y_{ij} is measured, and U_i is the group label for the ith subject, $U_i = 0$ if it is from group 0 and $U_i = 1$ if it is from group 1. Assuming a linear trend over time for both groups, we can have the following LME model

for our training sample (y_{ij}, t_{ij}, U_i),

$$y_{ij} = \beta_0 + \delta_0 U_i + b_{0i} + (\beta_1 + \delta_1 U_i + b_{1i}) t_{ij} + \epsilon_{ij} \tag{4.3}$$

where (β_0, β_1) and $(\beta_0+\delta_0, \beta_1+\delta_1)$ are the population intercept and slope of the linear trends (fixed effects) for group 0 and 1, respectively, and (b_{0i}, b_{1i}) are the unobservable subject-specific random effects for the intercept and slope. Together $(\beta_0 + \delta_0 U_i + b_{0i}, \beta_1 + \delta_1 U_i + b_{1i})$ represent the subject-specific intercept and slope. The standard assumptions are that (b_{0i}, b_{1i}) are independent and identically distributed (iid) bivariate normal random vector with mean zero and covariance matrix $\mathbf{D}$, and the random errors ϵ_{ij} are iid univariate normal random variables with mean zero and variance σ_e^2. It is also assumed that the random effects are independent of the random errors.

The Fitting Phase

As mentioned above, it is difficult to directly use y_{ij} as features in the LR model. Instead, we can represent each subject by its subject-specific intercept and slope $(\beta_0+\delta_0 U_i + b_{0i}, \beta_1+\delta_1 U_i+b_{1i})$ and use them as features in the LR model. To this end, we first fit the LME model (4.3) to our training sample and obtain the estimates of $(\beta_0+\delta_0 U_i+b_{0i}, \beta_1+\delta_1 U_i+b_{1i})$. The estimates for $(\beta_0 + \delta_0 U_i + b_{0i}, \beta_1 + \delta_1 U_i + b_{1i})$ we use are the empirical Best Linear Unbiased Predictors (eBLUPs). In the following, we briefly describe how to obtain those eBLUPs.

We first introduce a few notations. Denote all the measurements for the ith subject by $\boldsymbol{y}_i = (y_{i1}, ..., y_{in_i})'$, and then $\boldsymbol{y} = (\boldsymbol{y}_1', ..., \boldsymbol{y}_m')'$ is the n-dimensional vector of all the measurements in the training sample, where $n = \sum_{i=1}^m n_i$. Similarly, denote the n-dimensional

vector of all random errors by $\boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1', ..., \boldsymbol{\epsilon}_m')'$, where $\boldsymbol{\epsilon}_i = (\epsilon_{i1}, ..., \epsilon_{in_i})'$. Further denote the vector of the fixed-effects by $\boldsymbol{\beta} = (\beta_0, \beta_1)'$ and $\boldsymbol{\delta} = (\delta_0, \delta_1)'$, and write $\boldsymbol{\gamma} = (\boldsymbol{\beta}', \boldsymbol{\delta}')'$, and denote the vector of random-effects by $\boldsymbol{b} = (b_{01}, b_{11}, ..., b_{0m}, b_{1m})'$. Then we can represent the model in (4.3) in the following matrix form,

$$\boldsymbol{y} = \mathbf{X}\boldsymbol{\gamma} + \mathbf{Z}\boldsymbol{b} + \boldsymbol{\epsilon}, \tag{4.4}$$

where $\mathbf{X}$ (of dimension $n \times 4$) and $\mathbf{Z}$ (of dimension $n \times 2m$)) are the design matrices for the fixed and random effects. The Best Linear Unbiased Predictors (BLUPs) of $\boldsymbol{b}$ and $\boldsymbol{\gamma}$ (denoted by $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{\gamma}}$, respectively) can be obtained by solving the following Mixed Model Equations (MME, see Henderson (1950)):

$$\begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \boldsymbol{\Sigma}_s^{-1} \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\gamma}} \\ \tilde{\boldsymbol{b}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\boldsymbol{y} \\ \mathbf{Z}'\mathbf{R}^{-1}\boldsymbol{y} \end{bmatrix} \tag{4.5}$$

where $\boldsymbol{\Sigma}_s = \mathbf{I}_m \otimes \mathbf{D}$ is the covariance matrix of $\boldsymbol{b}$, $\mathbf{R} = \sigma_e^2 \mathbf{I}_n$ is the covariance matrix of $\boldsymbol{\epsilon}$, $\mathbf{I}_p$ is the p-dimensional identity matrix and $\otimes$ denotes the Kronecker product. By plugging in estimates of $\mathbf{D}$ and σ_e^2 into $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{\gamma}}$, one can obtain the eBLUPs of $\boldsymbol{b}$ and $\boldsymbol{\gamma}$, denoted by $\hat{\boldsymbol{b}}$ and $\hat{\boldsymbol{\gamma}}$, respectively. The eBLUPs of the subject-specific intercept and slope $(\beta_0 + \delta_0 U_i + b_{0i}, \beta_1 + \delta_1 U_i + b_{1i})$ then can be obtained by replacing $\beta_0, \delta_0, b_{0i}, \beta_1, \delta_1$, and b_{1i} by their respective eBLUPs. We denote those eBLUPs by $(\hat{\alpha}_{0i}, \hat{\alpha}_{1i})$.

Once we obtain $(\hat{\alpha}_{0i}, \hat{\alpha}_{1i})$, $i = 1, ..., m$, our training data can be represented by $\{(\hat{\alpha}_{0i}, \hat{\alpha}_{1i}, U_i)\}_{i=1}^m$. A fitted LR model is given by

$$\log(\hat{p}_i/(1 - \hat{p}_i)) = \hat{c}_0 + \hat{c}_1\hat{\alpha}_{0i} + \hat{c}_2\hat{\alpha}_{1i}, \tag{4.6}$$

where $\hat{p}_i$ is the predicted probability of the ith subject belonging to group 1, and the $\hat{c}_i$ $(i = 0, 1, 2)$ are the fitted coefficients in the LR model based on the training data.

The Prediction Phase

Since we use eBLUPs of subject-specific intercepts and slopes as the features in the above LR model, we need to find eBLUPs of the subject-specific intercept and slope for a new subject. To this end, we first need to modify the LME model in (4.3) to account for the fact that we do not know the group label of a new subject. Assume that the new subject has s measurements, denoted by $\boldsymbol{y}^* = (y_1^*, ..., y_s^*)'$. The time points when the s measurements are taken are $(t_1, ..., t_s)$. Then the LME model for $\boldsymbol{y}^* = (y_1^*, ..., y_s^*)'$ is

$$y_j^* = \beta_0 + \delta_0 W + b_0 + (\beta_1 + \delta_1 W + b_1)t_j + \epsilon_j, \quad j = 1, ..., s,$$

where W now is a Bernoulli random variable with probability π being the prevalence proportion of group 1. Throughout this chapter, we assume that π is known *a priori* or can be estimated based on the random sample of training data. The subject-specific intercept and slope for this new subject is then $(\beta_0 + b_0 + \delta_0 W, \beta_1 + b_1 + \delta_1 W)$. Define $\mathbf{X}^* = \begin{bmatrix} \mathbf{1}_s & \boldsymbol{t} \end{bmatrix}$ and $\mathbf{Z}^* = \begin{bmatrix} \mathbf{X}^* & \mathbf{X}^*\boldsymbol{\delta} \end{bmatrix}$, where $\mathbf{1}_s$ is a vector of s ones and $\boldsymbol{t} = (t_1, ..., t_s)'$. The eBLUP of $(\beta_0 + \delta_0 W + b_0, \beta_1 + \delta_1 W + b_1)'$, denoted by $(\hat{\alpha}_0^*, \hat{\alpha}_1^*)'$, can be shown to be equal to

$$\hat{\boldsymbol{\beta}} + \pi\hat{\boldsymbol{\delta}} + \left\{\mathbf{X}^*\hat{\mathbf{D}} + \pi(1-\pi)\mathbf{X}^*\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}'\right\}' \left(\mathbf{Z}^*\hat{\boldsymbol{\Lambda}}\mathbf{Z}^{*\prime} + \hat{\sigma}_e^2\mathbf{I}_s\right)^{-1} \left\{\boldsymbol{y}^* - \mathbf{X}^*\hat{\boldsymbol{\beta}} - \mathbf{Z}^*\boldsymbol{\pi}\right\}, \qquad (4.7)$$

where $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1)'$, and $\hat{\boldsymbol{\delta}} = (\hat{\delta}_0, \hat{\delta}_1)'$ are elements of the eBLUPs of $(\beta_0, \beta_1, \delta_0, \delta_1)'$ obtained from the training sample, $\hat{\mathbf{D}}$ and $\hat{\sigma}_\epsilon^2$ are estimates of $\mathbf{D}$ and σ_ϵ^2, respectively (e.g. maximum likelihood estimates), $\hat{\boldsymbol{\Lambda}} = \begin{bmatrix} \hat{\mathbf{D}} & \mathbf{0} \\ \mathbf{0} & \pi(1-\pi) \end{bmatrix}$, and $\boldsymbol{\pi} = (0, 0, \pi)'$. A detailed derivation for

this formula can be found in the Appendix C. Once we obtain $(\hat{\alpha}_0^*, \hat{\alpha}_1^*)$, we can put them into the fitted LR model in (4.6) and obtain the predicted probability of this new subject belonging to group 1, namely,

$$p^* = \frac{\exp(\hat{c}_0 + \hat{c}_1\hat{\alpha}_0^* + \hat{c}_2\hat{\alpha}_1^*)}{1 + \exp(\hat{c}_0 + \hat{c}_1\hat{\alpha}_0^* + \hat{c}_2\hat{\alpha}_1^*)}. \tag{4.8}$$

Extension to Nonparametric Mixed Effects Model

The LME model in the previous section is suitable for the situation where one is willing to assume a parametric form for the data trend. However, it might be difficult to make such an assumption for many real applications. While other specific parametric trends can be incorporated as alternatives, a Nonparametric Mixed Effects (NME) model has been developed as a more robust approach. This NME model does not require the specification of any parametric form for the data trend. Considering our training data (y_{ij}, t_{ij}, U_i), the corresponding NME model is as follows:

$$y_{ij} = g_0(t_{ij})I_{\{U_i=0\}} + g_1(t_{ij})I_{\{U_i=1\}} + v_i(t_{ij}) + \epsilon_{ij}$$

where $g_0(\cdot)$, $g_1(\cdot)$ and $v_i(\cdot)$ are all smooth functions, $g_0(\cdot)$ and $g_1(\cdot)$ model the population mean curves (fixed effects) for group 0 and 1, respectively, $v_i(\cdot)$ models the individual curve variation (random effects) from its population mean curve, and $I_{\{A\}}$ takes the value of 1 if the condition A is true and 0 otherwise. Using basis functions to approximate the above smooth functions, we can rewrite the model as

$$y_{ij} = \sum_{k=0}^{q}(\beta_k + \delta_k U_i + b_{ki})\Psi_k(t_{ij}) + \epsilon_{ij} \tag{4.9}$$

where $\Psi_k(\cdot)$, $k = 0, ..., q$, are the basis functions, $(\beta_0, \beta_1, \cdots, \beta_q)$ are the coefficients of the basis functions in the approximation of $g_0(\cdot)$ and $(\beta_0 + \delta_0, \beta_1 + \delta_1, \cdots, \beta_q + \delta_q)$ are the coefficients of the basis functions in the approximation of $g_1(\cdot)$, and $\boldsymbol{b}_i = (b_{0i}, b_{1i}, \cdots, b_{qi})'$ are the coefficients of the basis functions in the approximation of $v_i(\cdot)$. Similar to the LME model, it is usually assumed that the $\boldsymbol{b}_i$ are iid $(q+1)$-dimensional multivariate normal random vectors with mean zero and covariance $\mathbf{D}$ and the ϵ_{ij} are iid univariate normal random variables with mean zero and variance σ_e^2, and that the $\boldsymbol{b}_i$ are independent of the ϵ_{ij}.

The basis functions $\Psi_k(\cdot)$ we use in our simulation study and real data analysis are the B-spline basis functions. There are many other choices of basis functions. For a complete list of basis functions, one can refer to Wu and Zhang (2006). After we choose appropriate basis functions, the model can be seen as a linear mixed effects model with respect to the basis functions. Then, similar to the previous section, we can use the eBLUPs of $(\beta_0 + \delta_0 U_i + b_{0i}, ..., \beta_q + \delta_q U_i + b_{qi})$ as the features to represent each subject in the training sample. The model fitting technique we discuss in the previous section can be used to find those eBLUPs after we redefine $\mathbf{X}$, $\boldsymbol{\beta}$, $\mathbf{Z}$, $\boldsymbol{b}$ according to model (4.9). An LR model is then fitted with those eBLUPs as features. The feature vector of a new subject can be similarly obtained by the eBLUP of $(\beta_1 + \delta_1 W + b_{1i}, ..., \beta_q + \delta_q W + b_{qi})$, where W is a Bernoulli random variable with probability π being the prevalence proportion of group 1 (see details in Appendix C). Based on this feature vector and the fitted LR model from the training sample, we can obtain p^*, the predicted probability of this new subject belonging to group 1.

4.2.3 Sequential Classification Procedure

In this section, we detail our sequential classification procedure. At the training phase, a mixed effects model is fitted to the training data. The eBLUPs of the subject-specific random effects from the fitted mixed effects model are then used as features to fit a LR model. For a new subject, starting from the first time point, we find the eBLUPs of its subject-specific random effects using the measurements available, say $(y_1^*, ..., y_\ell^*)$, from the prediction phase mixed effects model. We then insert the eBLUPs into the fitted LR model to find p^*, the predicted probability of belonging to group 1. Given the cost structure in Table 4.1, we will classify the new subject according to the rule

$$\begin{cases} \text{classify to group 0} & \text{if} \quad p^* \leq \frac{1}{\rho_{01}} \\ \text{classify to group 1} & \text{if} \quad p^* \geq 1 - \frac{1}{\rho_{10}} \\ \text{classify as N} & \text{if} \quad \frac{1}{\rho_{01}} \leq p^* \leq 1 - \frac{1}{\rho_{10}} \end{cases}$$

If we classify the new subject as N, the subject is evaluated again after collecting another measurement $y_{\ell+1}^*$ using the same procedure as above with the updated measurements $(y_1^*, ..., y_{\ell+1}^*)$. This process continues until the subject is classified to group 0 or 1, or the measurement process reaches the last time point. At the last time point, if we have to classify the subject to group 0 or 1, we will use the classifier in (4.2) with $p_1(\boldsymbol{y})$ being replaced by the most recent p^*.

Our sequential classifier depends on the cost structure specified in Table 4.1. In our sequential procedure, C_N can be interpreted as the cost of time. The costs of making misclassification (C_{01} and C_{10}) as well as the cost of time C_N can be determined based

on the specific application, according to expert opinion. Different cost structures lead to different sequential classifiers, and as mentioned earlier, one could easily incorporate time-varying cost structures. In general, a higher ratio between the cost of misclassification and the cost of time (i.e. a higher ρ_{01} or ρ_{10}) would result in a more cautious classifier which allows for waiting longer and collecting more measurements before a classification can be made. On the other hand, a lower cost ratio would drive the classifier to make a classification earlier in order to reduce the cost of time. The effect of different cost structures on the performance of our sequential classifier will be investigated further in the next section. There are two extreme cases worth discussing. If one does not care about the cost of time (i.e. $C_N = 0$), our sequential classifier becomes the conventional non-sequential classifier and makes decisions at the last time point where data can be collected using (4.2). However, if the cost of time is large compared to the misclassification cost such that there is no neutral zone based on (4.1), our proposed classifier would force classification at the first time point to prevent introducing more cost of time. In either case, our sequential classifier would result in a cost as low as, if not lower than, the conventional non-sequential method.

4.3 Performance Evaluation

In this section, we report on a simulation study that demonstrates the performance of our proposed sequential classifier. First, we generate training data from the LME model in (4.3). In order to match prevalence proportion in our real sepsis example, we use 200 subjects in group 0 and 800 subjects in group 1 for the training data set. For each subject the measurements are taken at up to 7 time points that are equally spaced at $0/7, 1/7, \cdots, 6/7$.

The covariance matrix of the random effects (b_{0i}, b_{1i}) is $\mathbf{D} = \begin{pmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{pmatrix}$. Values for σ_0^2, σ_1^2, and σ_e^2 are chosen from 0.2, 0.4, and 0.6. Three scenarios are considered for the linear trend over time from the two groups: V-shape, reverse V-shape, and X-shape. For the V-shape situation, we use $\beta_0 = 0$, $\delta_0 = 0$, $\beta_1 = 0.5 \times \sqrt{\sigma_0^2 + \sigma_1^2 + \sigma_e^2}$, and $\delta_1 = -2 \times \beta_1$. For the reverse V-shape scenario, we use $\beta_0 = 0.5 \times \sqrt{\sigma_0^2 + \sigma_1^2 + \sigma_e^2}$, $\delta_0 = -2 \times \beta_0$, $\beta_1 = -0.5 \times \sqrt{\sigma_0^2 + \sigma_1^2 + \sigma_e^2}$, and $\delta_1 = -2 \times \beta_1$. Finally for the X-shape case, we use $\beta_0 = \frac{1}{2} \times 0.5 \times \sqrt{\sigma_0^2 + \sigma_1^2 + \sigma_e^2}$, $\delta_0 = -2 \times \beta_0$, $\beta_1 = -0.5 \times \sqrt{\sigma_0^2 + \sigma_1^2 + \sigma_e^2}$, and $\delta_1 = -2 \times \beta_1$. The above fixed-effects parameters are chosen so that the two populations in each scenario have only moderate separation. Typical simulated sample paths for the V-shape are shown in Figure 4.2. The black lines represent simulated trajectories for subjects in group 0, while gray lines represent trajectories for subjects in group 1. The bold lines are population curves.

In our simulation, we incorporate missing values via a missing at random (MAR) mechanism. MAR is a more realistic mechanism in longitudinal data than missing completely at random (MCAR) (see Fitzmaurice et al. (2004) for example). We assume that the MAR mechanism follows a drop-out model, which implies that as soon as a measurement is missing at time point t for one subject, there will not be any further measurements observed for that subject. To generate our missing data, we utilize a model in Fitzmaurice et al. (2004). More specifically, given that j measurements have been observed for the ith subject, the probability that the $j+1$th measurement for the subject will be missing, $P_{i,j+1}$,

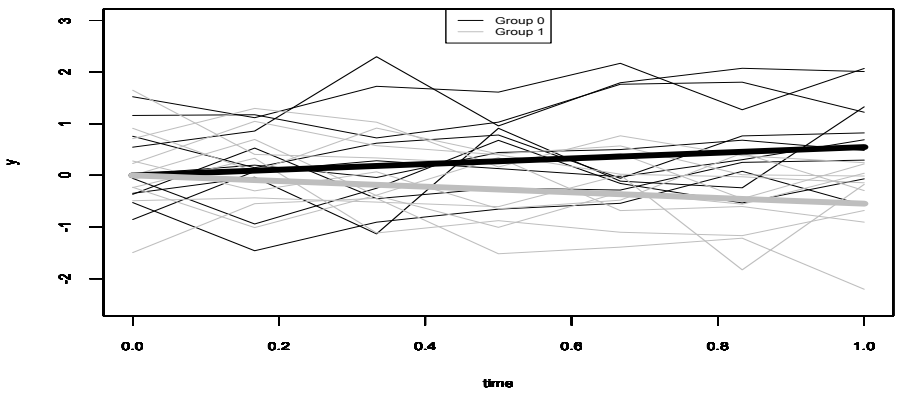


Figure 4.2: Typical Sample Path for LME Simulation: V-shape

is determined by:

$$\log \frac{P_{i,j+1}}{1 - P_{i,j+1}} = \theta_0 + \theta_1 y_{ij} + \theta_2 y_{i,j+1} \tag{4.10}$$

where y_{ij} is the jth measurement for subject i. In (4.10), if $\theta_1 \neq 0$ but $\theta_2 = 0$, we have MAR. In our simulation, we choose $\theta_0 = -1$, $\theta_1 = 1$, and $\theta_2 = 0$. Based on those choices, the overall missing proportion is approximately 50%.

Table 4.2: Cost Structures for Simulations

setting	C_N	C_{10}	C_{01}
1	1	4	4
2	1	4	16
3	1	100	100

As mentioned in Section 4.2.3, the performance of our sequential classifier depends on the two cost ratios ρ_{01} and ρ_{10}. In our simulation, we consider three cost structures listed in Table 4.2. In the first setting, the costs for the two types of misclassification are equal.

In the second setting, we have unequal costs for the two types of misclassification, which is quite common in many disease diagnosis studies where misclassifying a subject into one of the two groups has a relatively more severe consequence. The last setting is an extreme case where both types of misclassification would cost a lot more than the cost of time. According to the comments in Section 4.2.3, we can expect that this cost structure would lead to a very cautious classifier, with classification being made mostly at the last time point.

To evaluate the performance of our sequential classifier, a test set is simulated using the same LME model that generated the training data. There are 1000 subjects in the test set, with 200 in group 0 and 800 in group 1. To simplify our comparison, we do not incorporate missing data in the test set. We apply our sequential classifier to this test set and the average cost of our sequential classifier is calculated as follows. At each time point (starting from the first time point) we use our sequential procedure to predict the group label for each subject in the test set. When the predicted group label is N, we add C_N to the cost of this subject. When the predicted group label is 0 or 1, we add the corresponding misclassification cost to the cost of this subject if a misclassification is made. This way we obtain the total cost for each subject in the test set. The average cost of our sequential classifier is then obtained by averaging the total cost of the 1000 subjects in the test set.

To compare the performance of our sequential classifier to the non-sequential classifier, we also classify each subject in the test set one time using all 7 measurements. That is, we calculate the predicted probability of each subject in the test set belonging to group 1 using all 7 measurements. We then use those predicted probabilities as their posterior probabilities $p_1(\boldsymbol{y})$, and classify the subject to group 0 or 1 according to the classification

rule in (4.2). The cost for each subject from this non-sequential classifier is then the sum of the time cost ($C_N \times 6$) and the misclassification cost (C_{01} or C_{10}) if a misclassification is made. The average cost for this non-sequential classifier is then the average of the cost of the 1000 subjects in the test set.

We repeat the above simulations 100 times and the boxplots of the average costs for the sequential classifier and non-sequential classifier for the V-shape model under the three cost structures are shown in Figure 4.3 (a) - (c). From the figure, we can see that, when the cost ratio between the cost of misclassification and the cost of time (i.e. ρ_{01} or ρ_{10}) is low, our sequential classifier benefits from being able to make an early decision, which results in significant reduction in the overall cost. When the cost ratio becomes larger, which implies that the cost of time is less significant compared to the cost of misclassification, our sequential classifier becomes more cautious and more strict in terms of making early decisions. In the extreme case as in our third cost structure setting, our sequential classifier tends to wait until a later time point to make the decision. This leads to the similar average cost performance between the two methods. The results for the reverse V-shape and X-shape models are shown in Figure 4.3 (d) - (f) and Figure 4.4 (a) - (c). They are very similar to the V-shape case.

We also carried out the above simulation using an NME model. The model we use to generate data is the following trigonometric model, which was discussed in Wu and Zhang (2002):

$$y_{ij} = (\alpha_0 + \gamma_0 U_i + a_{0i}) + (\alpha_1 + \gamma_1 U_i + a_{1i}) \cos(2\pi t_{ij}) + (\alpha_2 + \gamma_2 U_i + a_{2i}) \sin(2\pi t_{ij}) + \epsilon_{ij}$$

where $\mathbf{a}_i = (a_{0i}, a_{1i}, a_{2i})'$ are random-effects, with a_{0i} following a normal distribution with

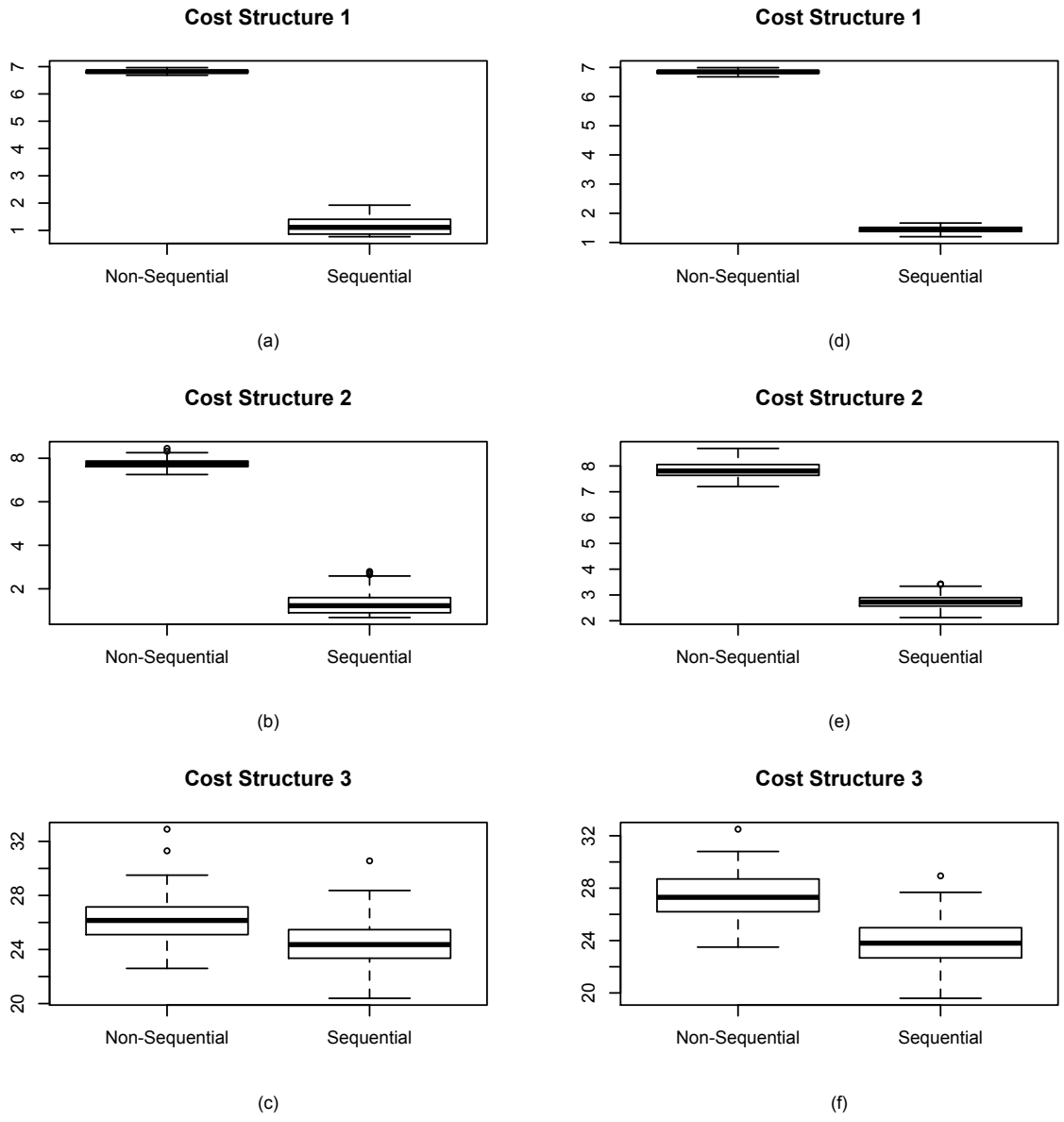


Figure 4.3: Cost Comparisons 1: (a) - (c) represents cost comparisons for V-shape profiles; (d) - (f) represents cost comparisons for reverse V-shape profiles.

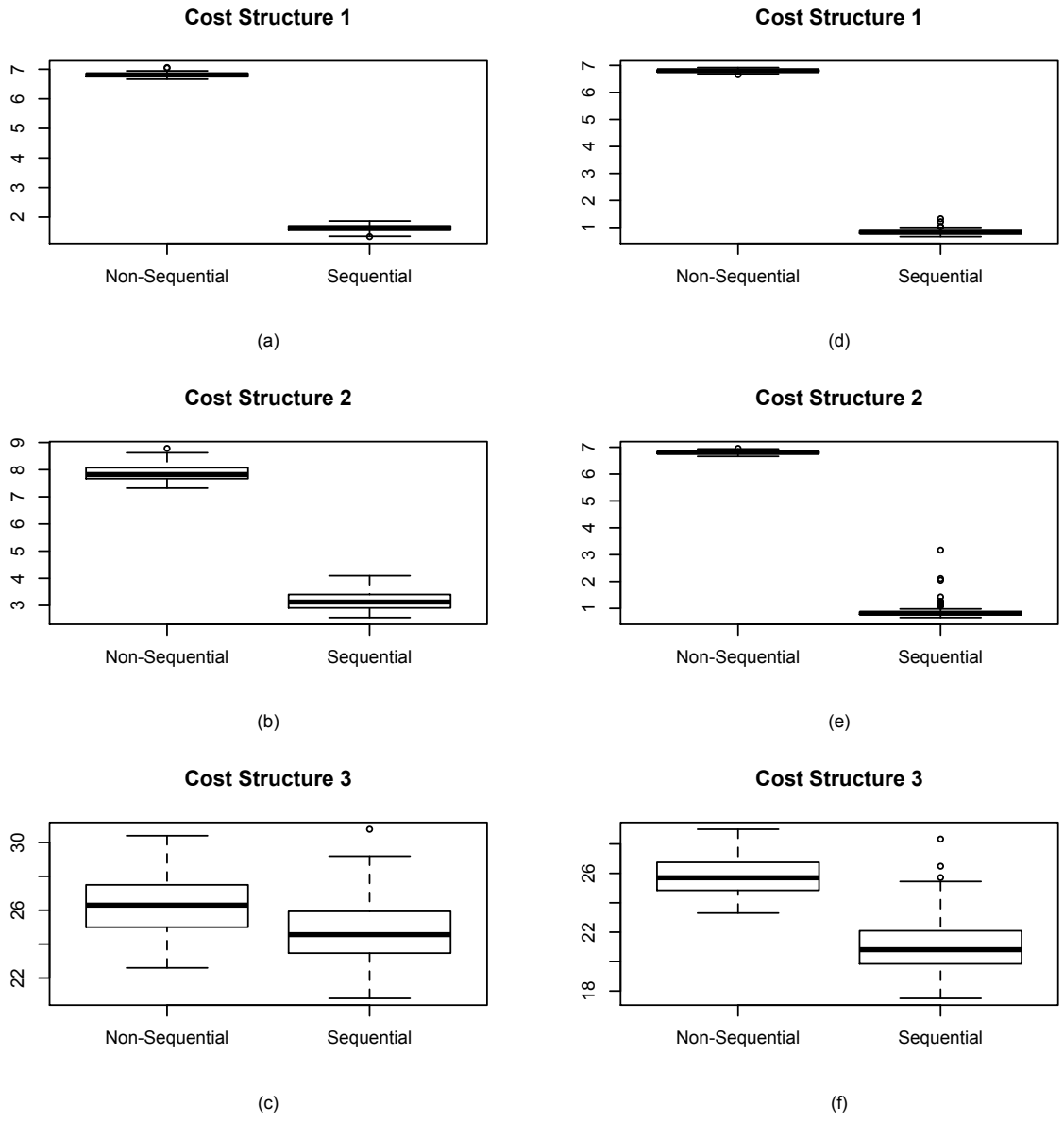


Figure 4.4: Cost Comparisons 2: (a) - (c) represents cost comparisons for X-shape profiles; (d) - (f) represents cost comparisons for trigonometric model.

zero mean and standard deviation of 2, and a_{1i} and a_{2i} both following a standard normal distribution. In order for the two groups to have moderate seperation, we use $\alpha_0 = 1, \gamma_0 = 0, \alpha_1 = 1.5, \gamma_1 = -1, \alpha_2 = 2,$ and $\gamma_2 = -1.5$. The model contains sin-cos functions that would be difficult to capture with an LME model. Therefore we use the NME model described in Section 4.2.2. In the NME model, we use the B-spline basis functions with $q = 3$. The boxplots of the average costs for the sequential classifier and the non-sequential classifier over 100 simulations under the three cost structures are shown in Figure 4.4 (d) - (f). The results are very similar to what we observe in the LME setting. A lower cost ratio between the cost of misclassification and the cost of time results in much lower average cost for our sequential classifier. This indicates that, when the cost of time is not negligible compared to the cost of misclassification, the sequential classifier is a more suitable choice.

4.4 Application to Sepsis Study

In this section, we use the sepsis data introduced in Section 4.1 to illustrate the application of our proposed sequential classifier. Sepsis is a potentially lethal medical condition characterized by a whole-body inflammatory state that is triggered by an infection. Early detection of sepsis can significantly reduce the chance of death. Therefore, diagnosis of sepsis as early as possible is extremely helpful for managing the disease. The sepsis diagnosis study includes 990 individuals, with 798 confirmed severe sepsis patients and 192 confirmed non-severe sepsis patients. Within a 72 hour window, up to 7 blood draws were taken from each subject at time points $t = 0, 3, 6, 12, 24, 48,$ and 72 hours, and the biomarker level at each time point was measured. A subset of profiles is displayed in Figure 4.1.

Since we do not have any information about the form of the data trend, we use the NME model in our sequential classifier as described in 4.2.2. In the NME model, we use the B-spline basis functions with $q = 2$. To assess the performance of our sequential classifier, we use a 10-fold cross validation to estimate the expected cost. More specifically, the sepsis data is randomly partitioned into 10 disjoint sets, preserving the proportions of the two groups in the original data set. Each of the 10 sets then takes a turn to serve as the test set and the remaining nine sets serve as the training data. For each of the test sets, we calculate the average costs of our sequential classifier and the non-sequential classifier as in our simulation study. Table 4.3 reports the average costs of the two classifiers over the 10 test sets. The cost structures we use are the same as in Table 4.2. The cost of time C_N is assumed to be the cost per hour. We can see from the table that the sequential classifier has a much lower average cost than the non-sequential classifier. Even for the extreme case, where the cost ratio between the cost of misclassification and the cost of time is as large as 100, the sequential classifier still has a significant advantage.

Table 4.3: Average Cost: Sepsis Data

Classifier	Cost Structure		
	1	2	3
Sequential	9.25	31.71	66.21
Non-Sequential	72.91	74.02	94.74

We also use our 10-fold cross validation analysis to compare the average waiting time for the sequential and the non-sequential classifiers. For the non-sequential classifier, the classification for each subject is made at the last time point. Therefore, the average waiting time is 72 hours. For our sequential classifier, if a subject is assigned to group 0 or

1 at time point t, then the waiting time is t. By averaging all the subjects' waiting time, we can obtain the average waiting time for our sequential classifier. Table 4.4 shows the average waiting time for the two classifiers under the three cost structures. From the table we can see that the sequential classifier has a much lower average waiting time than its non-sequential counterpart for cost structures 1 and 2. Even for cost structure 3, where our sequential classifier does not encourage to make early decisions, a smaller average waiting time is observed. From the table, we can see significant time can be saved when we apply the sequential classifier. This is often very important in the disease diagnosis process.

Table 4.4: Average Waiting Time: Sepsis Data

Classifier	Cost Structure		
	1	2	3
Sequential	8.47	29.80	44.30
Non-Sequential	72	72	72

Chapter 5

Concluding Remarks

In this dissertation, we propose sequential procedures for statistical process control and longitudinal data classification. This includes the development of two nonparametric multivariate CUSUM control charts for location and scale change detection, a wavelet-based nonparametric CUSUM control chart for autocorrelated processes, and the construction of a sequential classifier for longitudinal data.

Our proposed nonparametric multivariate control charts can be viewed as the nonparametric counterparts of the two parametric multivariate CUSUM procedures developed by Crosier (1988). The first one is based on the spatial sign, which is particularly powerful for detecting location shifts. The second one is based on the spatial depth, which is particularly suitable for detecting scale increases. Computation of the control limit for each of these CUSUM procedures is particularly easy due to their distribution-free properties. We recommend using both procedures in practice, since it is rarely known in advance what type of distributional changes the process will have. Similar to multiple comparison problem,

when using both procedures in parallel, the nominal ARL_0 for each individual procedure needs to be adjusted so that the overall ARL_0 is still controlled at the desired level. One possible adjustment based on Bonferroni inequality is doubling the nominal ARL_0 for each individual procedure. Similar to all other CUSUM procedures in the literature, the two proposed CUSUM procedures depend on the choice of k. In general, for both procedures, smaller k is more powerful for detecting smaller changes, while larger k is more powerful for detecting larger changes. In practice, if some location shift or scale increase is particularly of interest, we can always choose an optimal k by comparing the performance of different choices of k under this particular distributional change.

On the foundation of the above spatial sign based nonparametric multivariate CUSUM control chart, we develop a nonparametric CUSUM control chart for autocorrelated processes. The procedure utilizes the approximate decorrelation property of the wavelet coefficients, which makes the procedure robust to serially correlated processes, including the ones with long-memory. The method is also approximately distribution-free under a variety of distributions if a smaller k for the CUSUM scheme is chosen. We conducted extensive simulation studies and compared our proposed method with residual-based nonparametric control charts. Our proposed method illustrates a superior ARL_0 control, while at the same time, has a similar, if not better, ARL_1 performances. We also used network CPU usage data to demonstrate a real application of our method. The procedure outperforms its competitors both in ARL_0 control and detection power. Implementation issues such as determination of decomposition scale, reference sample depth, and missing value treatment are also discussed. Our proposed wavelet-based method is easy to implemente and therefore

will be widely useable to many applications that feature serially correlated data patterns. In this dissertation we only considered location shift detection. Some further study may be conducted by applying similar procedure but incorporating the DD-CUSUM control chart discussed in Chapter 2 to develop a SPC procedure for scale change detection. Ultimately, our goal would be combining the two procedures together to form a control chart for both location shift and scale increase detection.

In this dissertation, we also consider another sequential procedure for classification problems. The proposed sequential classifier for longitudinal data is based on a neutral zone classifier. Simulation results show that the proposed sequential classifier can significantly reduce the average cost compared to the non-sequential classifier which waits until the last time point to classify when the cost of time is taken into account. The proposed sequential classifier is applied to real data obtained from a severe sepsis diagnosis study and is shown to outperform the non-sequential classifier. Our proposed sequential classifier is especially valuable for its reduction in the time required to make a classification, which is very important in disease diagnosis and other applications where the cost of time is significant. There are several interesting topics for future study. For example, the cost of time C_N considered in this dissertation is fixed. However, in real life, applications may require a changing cost over time. For instance, the cost of time may increase for patients waiting for a diagnosis result, especially in the context of acute disease diagnosis. Also we mainly focus on the two-class classification problem. Another research topic we plan to pursue is how to extend our sequential classifier to the multiple-class setting.

References

[1] Aradhye, H. B., Bakshi, B. R., Strauss, R. A., and Davis, J. F. Multiscale SPC using wavelets - theoretical analysis and properties. *AIChE Journal 49*, 4 (2003), 939–958.

[2] Bakir, S. T., and Reynolds, M. R. A nonparametric procedure for process control based on within-group ranking. *Technometrics 21*, 2 (1979), 175–183.

[3] Benecke, S., Jeske, D. R., Reugger, P., and Borneman, J. Bayes neutral zone classifiers with applications to nonparametric unsupervised settings. *Journal of Agricultural, Biological, and Environmental Statistics 18* (2013), 39–52.

[4] Beran, J. *Statistics for Long-Memory Processes*. Chapman and Hall/CRC, 1994.

[5] Bersimis, S., Psarakis, S., and J.Panaretos. Multivariate statistical process control charts: An overview. *Quality and Reliability Engineering International 23*, 5 (2007), 517–543.

[6] Beylkin, G., Coifman, R., and Rokhlin, V. Fast wavelet transforms and numerical algorithms. *Communications on Pure and Applied Mathematics 44* (1991), 141–183.

[7] Black, G., Smith, J., and Wells, S. The impact of weibull data and autocorrelation on the performance of the Shewhart and Exponentially Weighted Moving Average control charts. *International Journal of Industrial Engineering Computations 2* (2011), 575–582.

[8] Chakraborty, B., Chaudhuri, P., and Oja, H. Operating transformation retransformation on spatial median and angle test. *Statistica Sinica 8* (1998), 767–784.

[9] Chan, L. K., and Zhang, J. Cumulative sum control charts for the covariance matrix. *Statistica Sinica 11* (2001), 767–790.

[10] Chaudhuri, P. On a geometric notion of quantiles for multivariate data. *Journal of the American Statistical Association 91* (1996), 862–872.

[11] Cheng, S. W., and Thaga, K. Max-CUSUM chart for autocorrelation processes. *Statistica Sinica 15* (2005), 527–546.

[12] Craigmile, P. F., and Percival, D. B. Asymptotic decorrelation of between-scale wavelet coefficients. *IEEE Transactions on Information Theory 51*, 3 (2005), 1039–1048.

[13] Crosier, R. B. Multivariate generalization for cumulative sum quality-control schemes. *Technometrics 30* (1988), 291–303.

[14] Daubechies, I. Orthonormal bases of compactly supported wavelets. *Communications of Pure and Applied Mathematics XLI* (1988), 909–996.

[15] Donoho, D. L. Breakdown properties of multivariate location estimators. Ph.D. qualifying paper, Harvard University, 1982.

[16] Donoho, D. L., and Gasko, M. Breakdown properties of location estimates based on half-space depth and projected outlyingness. *Annals of Statistics 20* (1992), 1803–1827.

[17] Fitzmaurice, G. M., Laird, N. M., and Ware, J. H. *Applied Longitudinal Analysis*. John Wiley and Sons, New York, 2004.

[18] Ganesan, R., Das, T. K., and Venkataraman, V. Wavelet-based multiscale statistical process monitoring: A literature review. *IIE Transactions 36*, 9 (2010), 787–806.

[19] Granger, C., and Joyeux, R. An introduction to long-range time series models and fractional differencing. *Journal of Time Series Analysis 1* (1980), 15–30.

[20] Harris, T. J., and Ross, W. H. Statistical process control prodedures for correlated observations. *The Canadian Journal of Chemical Engineering 69* (1991), 48–57.

[21] Harville, D. A. Extension of the Gauss-Markov theorem to include the estimation of random effects. *The Annals of Statistics 4* (1976), 384–395.

[22] Harville, D. A. Maximum likelihood approaches to variance component estimation and to related problems. *Journal of the American Statistical Association 72* (1977), 320–340.

[23] Haslett, J., and Raftery, A. E. Space-time modelling with long-memory dependence: Assessing Ireland's wind power resource (with discussion). *Applied Statistics 38* (1989), 1–50.

[24] Hawkins, D. M., and Olwell, D. H. *Cumulative Sum Charts and Charting for Quality Improvement*. New York: Springer-Verlag, 1998.

[25] Hawkins, D. M., and Tchao, E. M. Multivariate exponentially weighted moving covariance matrix. *Technometrics 50* (2008), 155–166.

[26] Henderson, C. R. Estimation of genetic parameters. *Annals of Mathematical Statistics 21* (1950), 309–310.

[27] Hettmansperger, T. P., and Randles, R. H. A practical affine equivariant multivariate median. *Biometrika 89* (2002), 851–860.

[28] Hosking, J. Fractional differencing. *Biometrika 68* (1981), 165–176.

[29] Hyndman, R. J., and Khandakar, Y. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software 26*, 3 (2008).

[30] James, G. M., and Hastie, T. J. Functional linear discriminant analysis for irregularly sampled curves. *Journal of the Royal Statistical Society: Series B 63* (2001), 533–550.

[31] Janacek, G. J., and Meikle, S. E. Control charts based on medians. *Journal of the Royal Statistical Society: Series D (The Statistician) 46*, 1 (1997), 19–31.

[32] Jeske, D. R., Liu, Z., Bent, E., and Borneman, J. Classification rules that include neutral zones and their application to microbial community profiling. *Communication in Statistics - Theory and Methods 36* (2007), 1965–1980.

[33] Jeske, D. R., Oca, V. M. D., Bischoff, W., and Marvasti, M. CUSUM technique for timeslot sequence with application to network surveillance. *Computational Statistics and Data Analysis 53* (2009), 4332–4344.

[34] Johnson, R. A., and Bagshaw, M. The effect of serial correlation on the performance of CUSUM tests. *Technometrics 16* (1974), 103–112.

[35] Jr., M. R. R., and Cho, G. Y. Multivariate control charts for monitoring the mean vector and covariance matrix. *Journal of Quality Technology 38* (2006), 230–253.

[36] Jr., M. R. R., and Stoumbos, Z. G. Combinations of multivariate shewhart and mewma control charts for monitoring the mean vector and covariance matrix. *Journal of Quality Technology 40* (2008), 381–393.

[37] Koltchinskii, V. I. M-estimation, convexity and quantiles. *Annals of Statistics 25* (1997), 435–477.

[38] Li, J., Zhang, X., and Jeske, D. R. Nonparametric multivariate CUSUM control charts for location and scale changes. *Journal of Nonparametric Statistics 25* (2013), 1–20.

[39] Liu, R. On a notion of data depth based on random simplices. *Annals of Statistics 18* (1990), 405–414.

[40] Liu, R., and Singh, K. A quality index based on data depth and multivariate rank tests. *Journal of the American Statistical Association 88* (1993), 252–260.

[41] Liu, R. Y. Control chart for multivariate processes. *Journal of the American Statistical Association 90* (1995), 1380–1387.

[42] Liu, R. Y., Parelius, J. M., and Singh, K. Multivariate analysis by data depth: Descriptive statistics, graphics and inference. *The Annals of Statistics 27* (1999), 783–858.

[43] Luts, J., Molenberghs, G., Verbeke, G., Van Huffel, S., and Suykens, J. A. K. A mixed effects least squares support vector machine model for classification of longitudinal data. *Computational Statistics and Data Analysis 56* (2012), 611–628.

[44] Mallat, S. G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions and Pattern Analysis and Machine Intelligence 11*, 7 (1989), 674–693.

[45] Marshall, G., and Barón, A. E. Linear discriminant models for unbalanced longitudinal data. *Statistics in Medicine 19* (2000), 1969–1981.

[46] McDonald, D. A CUSUM procedure based on sequential ranks. *Naval Research Logistics 37* (1990), 627–646.

[47] Oca, V. M. D., Jeske, D. R., Zhang, Q., Rendon, C., and Marvasti, M. A CUSUM change-point detection algorithm for non-stationary sequences with application to data network surveillance. *The Journal of Systems and Software 83* (2010), 1288–1297.

[48] Park, C., Hernández-Campos, F., Le, L., Marron, J. S., Park, J., Pipiras, C., Smith, F. D., Smith, R. L., Trovero, M., and Zhu, Z. Long-range dependence analysis of internet traffic. *Journal of Applied Statistics 38* (2011), 1407–1433.

[49] Peiris, M. S., and Perera, B. J. C. On prediction with fractionally differenced ARIMA models. *Journal of Time Series Analysis 9*, 3 (1987), 215–220.

[50] Qiu, P., and Hawkins, D. A rank based multivariate cusum procedure. *Technometrics 43* (2001), 120–132.

[51] Qiu, P., and Hawkins, D. A nonparametric multivariate cusum procedure for detecting shifts in all directions. *Journal of the Royal Statistical Society, Series D 52* (2003), 151–164.

[52] Randles, R. H. A distribution-free multivariate sign test based on interdirections. *Journal of the American Statistical Association 84*, 1045-1050 (1989).

[53] Runger, G. C., and Willemain, T. R. Model-based and model-free control of autocorrelated processes. *Journal of Quality Technology 27*, 4 (1995), 283–292.

[54] Serfling, R. A depth function and a scale curve based on spatial quantiles. In *In Statistics and Data Analysis based on L_1-Norm and Related Methods (Y. Dodge ed.), Birkhaeuser* (2002), pp. 25–38.

[55] Shewhart, W. A. *Economic Control of Manufactured Products.* Van Nostrand Reinhold, New York, 1931.

[56] Soule, A., Salamatian, K., and Taft, N. Combining filtering and statistical methods for anomaly detection. *Internet Measurement Conference* (2005).

[57] Stahel, W. Robust schaetzungen: Infinitesmale optimalitaet und schaetzungen von kovarianzmatrizen. Ph.D. thesis, ETH Zurich, 1981.

[58] Stoumbos, Z. G., and Sullivan, J. H. Robustness to non-normality of the multivariate ewma control chart. *Journal of Quality Technology 34* (2002), 260–276.

[59] Tukey, J. Mathematics and the picturing of data. In *Proceedings of the 1975 International Congress of Mathematics* (1975), vol. 2, pp. 523–531.

[60] Tyler, D. E. A distribution-free m-estimator of multivariate scatter. *The Annals of Statistics 15*, 1 (1987), 234–251.

[61] Vardi, Y., and Zhang, C. H. The multivariate l_1-median and associated data depth. In *Proceedings of the National Academy of Sciences* (2000), vol. 97, pp. 1423–1426.

[62] Verbeke, G., and Lesaffre, E. A linear mixed-effects model with heterogeneity in the random-effects population. *Journal of the American Statistical Association 91*, 433 (1996), 217–221.

[63] Wald, A. Sequential tests of statistical hypotheses. *Annals of Mathematical Statistics 16* (1945), 117–186.

[64] Willemain, T. R., and Runger, G. C. Designing control charts using an empirical reference distribution. *Journal of Quality Technology 28*, 1 (1996), 31–38.

[65] Wu, H., and Zhang, J. Local polynomial mixed-effects models for longitudinal data. *Journal of the American Statistical Association 97* (2002), 883–897.

[66] Wu, H., and Zhang, J. *Nonparametric Regression Methods for Longitudinal Data Analysis: Mixed Effects Modeling Approaches.* Wiley-interscience, 2006.

[67] Yen, C., and Shiau, J. H. A multivariate control chart for detecting increases in process dispersion. *Statistica Sinica 20* (2010), 1683–1707.

[68] Yu, H., Jeske, D. R., Ruegger, P., and Borneman, J. A three-class neutral zone classifier using a decision - theoretic approach with application to dan array analyses. *Journal of Agricultural, Biological, and Environmental Statistics 15* (2010), 474–490.

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

[70] Zuo, Y. J. Projection-based depth functions and associated medians. *Annals of Statistics 31* (2003), 1460–1490.

[71] Zuo, Y. J., and Serfling, R. General notions of statistical depth function. *Annals of Statistics 28* (2000), 461–482.

Appendix A

Proofs

Proof of Proposition 1. Let $(\boldsymbol{Y}_1, \cdots, \boldsymbol{Y}_m)$ be the reference sample, and $(\boldsymbol{X}_1, \boldsymbol{X}_2, \cdots)$ be the future sample. Let $\boldsymbol{Z}_i = D\boldsymbol{Y}_i + \boldsymbol{v}$ and $\boldsymbol{W}_i = D\boldsymbol{X}_i + \boldsymbol{v}$, where D is an arbitrary $p \times p$ nonsigular matrix and $\boldsymbol{v}$ is an arbitrary p-dimensional vector. Denote $\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}}$ and $\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}}$ as the solution of $\boldsymbol{\theta}$ in (2.2) computed on $\boldsymbol{Y}_i$ and $\boldsymbol{Z}_i$, respectively. Based on the proposition in Hettmansperger and Randles (2002), we have $\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}} = D\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}} + \boldsymbol{v}$. Denote $\hat{A}_{\boldsymbol{Y}}$ and $\hat{A}_{\boldsymbol{Z}}$ as nonsingular $p \times p$ matrice computed on $\boldsymbol{Y}_i$ and $\boldsymbol{Z}_i$, respectively, that satisfy (2.3). The SS-CUSUM is affine-invariant if the statistic L_n calculated from $\boldsymbol{X}_i$ using $(\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}}, \hat{A}_{\boldsymbol{Y}})$ is the same as the one calculated from $\boldsymbol{W}_i$ using $(\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}}, \hat{A}_{\boldsymbol{Z}})$.

From (2.3), we have

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{A}_{\boldsymbol{Y}}(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}})(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}})'\hat{A}'_{\boldsymbol{Y}}}{\|\hat{A}_{\boldsymbol{Y}}(\boldsymbol{Y}_i - \hat{\boldsymbol{\theta}}_{\boldsymbol{Y}})\|^2}\right) = \frac{1}{p}I_p. \tag{A.1}$$

Define

$$\hat{A}^{*\prime}_{\boldsymbol{Z}}\hat{A}^{*}_{\boldsymbol{Z}} = (D')^{-1}\hat{A}'_{\boldsymbol{Y}}\hat{A}_{\boldsymbol{Y}}D^{-1}$$

where $\hat{A}_{\boldsymbol{Z}^*}$ is a nonsingular $p * p$ matrix. Hence, we have

$$\hat{A}'_{\boldsymbol{Y}}\hat{A}_{\boldsymbol{Y}} = D'\hat{A}^{*\prime}_{\boldsymbol{Z}}\hat{A}^*_{\boldsymbol{Z}}D \tag{A.2}$$

Left and right multiplying both sides of (A.1) with $\hat{A}'_{\boldsymbol{Y}}$ and $\hat{A}_{\boldsymbol{Y}}$, respectively, and plugging (A.2) into (A.1), we get

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{D'\hat{A}^{*\prime}_{\boldsymbol{Z}}\hat{A}^*_{\boldsymbol{Z}}(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})'\hat{A}^{*\prime}_{\boldsymbol{Z}}\hat{A}^*_{\boldsymbol{Z}}D}{\|\hat{A}^*_{\boldsymbol{Z}}(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})\|^2}\right) = \frac{1}{p}D'\hat{A}^{*\prime}_{\boldsymbol{Z}}\hat{A}^*_{\boldsymbol{Z}}D$$

Since D and $\hat{A}^*_{\boldsymbol{Z}}$ are both nonsigular $p \times p$ matrices, we can easily get

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{A}^*_{\boldsymbol{Z}}(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})'\hat{A}'^*_{\boldsymbol{Z}}}{\|\hat{A}^*_{\boldsymbol{Z}}(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})\|^2}\right) = \frac{1}{p}I_p.$$

Therefore, $\hat{A}^*_{\boldsymbol{Z}}$ is also a solution to

$$\frac{1}{m}\sum_{i=1}^{m}\left(\frac{A(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})'A'}{\|A(\boldsymbol{Z}_i-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})\|^2}\right) = \frac{1}{p}I_p. \tag{A.3}$$

According to Tyler (1987), the solution to (A.3) is unique up to a multiplication by some positive constant, hence $\hat{A}_{\boldsymbol{Z}} = k\hat{A}^*_{\boldsymbol{Z}}$. Therefore, we have

$$\hat{A}'_{\boldsymbol{Z}}\hat{A}_{\boldsymbol{Z}} = k^2(D')^{-1}\hat{A}'_{\boldsymbol{Y}}\hat{A}_{\boldsymbol{Y}}D^{-1}$$

Define $\Theta = k^{-1}\hat{A}_{\boldsymbol{Z}}D\hat{A}^{-1}_{\boldsymbol{Y}}$. It is easy to verify that Θ is orthogonal, and $\hat{A}_{\boldsymbol{Z}}D = k\Theta\hat{A}_{\boldsymbol{Y}}$. Then we have,

$$\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}}) = \hat{A}_{\boldsymbol{Z}}D(\boldsymbol{X}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}}) = k\Theta\hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}}), \quad n=1,2,...$$

Therefore,

$$\boldsymbol{U}_{\boldsymbol{W}^*_n} = \frac{\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})}{\|\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Z}})\|} = \Theta\frac{\hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}})}{\|\hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_n-\hat{\boldsymbol{\theta}}_{\boldsymbol{Y}})\|} = \Theta\boldsymbol{U}_{\boldsymbol{X}^*_n}$$

where $\boldsymbol{U}_{\boldsymbol{W}_n^*}$ and $\boldsymbol{U}_{\boldsymbol{X}_n^*}$ denote the spatial signs of $\boldsymbol{W}_n^*$ and $\boldsymbol{X}_n^*$, the transformed data of $\boldsymbol{W}_n$ and $\boldsymbol{X}_n$, respectively. Thus any orthogonally invariant statistic computed on $\boldsymbol{U}_{\boldsymbol{X}_n^*}$ will be affine-invariant. Recall that, in our SS-CUSUM procedure, $\boldsymbol{U}_n \equiv \boldsymbol{U}_{\boldsymbol{X}_n^*}$. Therefore, it is not difficult to see that our statistic L_n in SS-CUSUM is orthogonally invariant statistic computed on $\boldsymbol{U}_{\boldsymbol{X}_n^*}$. Hence, our SS-CUSUM procedure is affine-invariant. ■

Proof of Proposition 2. Since the in-control distribution F_0 belongs to the elliptical directions family, when the process is in-control, we have $\boldsymbol{X}_i = r_i D\boldsymbol{u}_i + \boldsymbol{\mu}$, where D is a fixed $p \times p$ nonsingular matrix, the $\boldsymbol{u}_i$ are independent and identically distributed uniformly distributed on the unit p sphere, and the r_i are positive scalars. Based on the transformation in (2.1), $\boldsymbol{X}_i^* = r_i \hat{A}_m D\boldsymbol{u}_i + \hat{A}_m(\boldsymbol{\mu} - \hat{\boldsymbol{\theta}}_m)$. Following the result in Hettmansperger and Randles (2002), $\hat{\boldsymbol{\theta}}_m \overset{a.s.}{\to} \boldsymbol{\mu}$, as $m \to \infty$. Therefore, $\boldsymbol{X}_i^*$ is asymptotically equivalent to $r_i \hat{A}_m D\boldsymbol{u}_i$. Since the statistic in our SS-CUSUM is only related to the direction vector of $\boldsymbol{X}_i^*$, the r_i are irrelevant. In other words, the SS-CUSUM from $\boldsymbol{X}_i$ is asymptotically equivalent to the SS-CUSUM from $D\boldsymbol{u}_i + \boldsymbol{\mu}$. Based on Proposition 1 that the SS-CUSUM is affine-invariant, the SS-CUSUM from $\boldsymbol{X}_i$ is asymptotically equivalent to the SS-CUSUM from $\boldsymbol{u}_i$. Therefore, the SS-CUSUM is asymptotically distribution-free under the elliptical directions family. ■

Proof of Proposition 3. The DD-CUSUM procedure is affine-invariant if the spatial depth of the transformed data we use to generate R statistic is affine-invariant. Using the same notation as in the proof of Proposition 1, it suffices to show that $SPD_{F_m^*}(\boldsymbol{X}_j^*) = SPD_{G_m^*}(\boldsymbol{W}_j^*)$, where F_m^* and G_m^* are the empirical distributions of the reference samples $(\boldsymbol{Y}_1^*, \cdots, \boldsymbol{Y}_m^*)$ and $(\boldsymbol{Z}_1^*, \cdots, \boldsymbol{Z}_m^*)$, respectively. From the proof of Proposition 1, we have

$\hat{A}_{\boldsymbol{Z}} D = k\Theta \hat{A}_{\boldsymbol{Y}}$. Therefore

$$\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_j - \boldsymbol{Z}_i) = \hat{A}_{\boldsymbol{Z}} D(\boldsymbol{X}_j - \boldsymbol{Y}_i) = k\Theta \hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_j - \boldsymbol{Y}_i).$$

This implies that

$$\begin{aligned} SPD_{G_m^*}(\boldsymbol{W}_j^*) &= 1 - \left\| \frac{1}{m} \sum_{i=1}^{m} \frac{\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_j - \boldsymbol{Z}_i)}{\|\hat{A}_{\boldsymbol{Z}}(\boldsymbol{W}_j - \boldsymbol{Z}_i)\|} \right\| \\ &= 1 - \left\| \frac{1}{m} \sum_{i=1}^{m} \frac{\Theta \hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_j - \boldsymbol{Y}_i)}{\|\hat{A}_{\boldsymbol{Y}}(\boldsymbol{X}_j - \boldsymbol{Y}_i)\|} \right\| = SPD_{F_m^*}(\boldsymbol{X}_j^*). \end{aligned}$$

The result follows. ■

Proof of Proposition 4. Define $\boldsymbol{\theta}_0$ and A_0 as the solutions to the following equations:

$$\begin{aligned} E_{F_0}\left(\frac{A(\boldsymbol{Y} - \boldsymbol{\theta})}{\|A(\boldsymbol{Y} - \boldsymbol{\theta})\|} \right) &= \mathbf{0}, \\ E_{F_0}\left(\frac{A(\boldsymbol{Y} - \boldsymbol{\theta})(\boldsymbol{Y} - \boldsymbol{\theta})' A'}{\|A(\boldsymbol{Y} - \boldsymbol{\theta})\|^2} \right) &= \frac{1}{p} I_p. \end{aligned}$$

It is easy to see that our $(\hat{\boldsymbol{\theta}}_m, \hat{A}_m)$ used in the transformation is the sample version of $(\boldsymbol{\theta}_0, A_0)$. Following the proof in Hettmansperger and Randles (2002), we have $\hat{\boldsymbol{\theta}}_m \overset{a.s.}{\to} \boldsymbol{\theta}_0$, $\hat{A}_m \overset{a.s.}{\to} A_0$, as $m \to \infty$.

Define $\boldsymbol{Z}_i = A_0(\boldsymbol{Y}_i - \boldsymbol{\theta}_0)$ and denote the empirical distribution of $\boldsymbol{Z}_i$ $(i = 1, ..., m)$

by G_m and its population version as G. Therefore, for any $\boldsymbol{y} \in \Re^p$, we have

$$
\begin{aligned}
&\left|SPD_{F_m^*}(\hat{A}_m(\boldsymbol{y}-\hat{\boldsymbol{\theta}}_m)) - SPD_{G_m}(A_0(\boldsymbol{y}-\boldsymbol{\theta}_0))\right| \\
=&\left|\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)\|}\right\| - \left\|\frac{1}{m}\sum_{i=1}^{m}\frac{A_0(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|A_0(\boldsymbol{y}-\boldsymbol{Y}_i)\|}\right\|\right| \\
\leq&\left\|\frac{1}{m}\sum_{i=1}^{m}\frac{\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)\|} - \frac{1}{m}\sum_{i=1}^{m}\frac{A_0(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|A_0(\boldsymbol{y}-\boldsymbol{Y}_i)\|}\right\| \\
\leq&\frac{1}{m}\sum_{i=1}^{m}\left\|\frac{\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|\hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i)\|} - \frac{A_0(\boldsymbol{y}-\boldsymbol{Y}_i)}{\|A_0(\boldsymbol{y}-\boldsymbol{Y}_i)\|}\right\| \\
=&\frac{1}{m}\sum_{i=1}^{m}2\sin\alpha_j/2 \quad \text{(where } \alpha_j \text{ is the angle between } \hat{A}_m(\boldsymbol{y}-\boldsymbol{Y}_i) \text{ and } A_0(\boldsymbol{y}-\boldsymbol{Y}_i)\text{)} \\
\leq&\frac{1}{m}\sum_{i=1}^{m}2\sin\alpha_j \quad \text{(provided that } \alpha_j \leq \pi/2, \text{ which can be guaranteed with sufficiently large } m\text{)} \\
\leq&\frac{1}{m}\sum_{i=1}^{m}2\frac{\|(\hat{A}_m - A_0)(\boldsymbol{y}-\boldsymbol{Y}_i)\|}{\|A_0(\boldsymbol{y}-\boldsymbol{Y}_i)\|} \\
=&\frac{1}{m}\sum_{i=1}^{m}2\sqrt{\frac{(\boldsymbol{y}-\boldsymbol{Y}_i)'(\hat{A}_m - A_0)'(\hat{A}_m - A_0)(\boldsymbol{y}-\boldsymbol{Y}_i)}{(\boldsymbol{y}-\boldsymbol{Y}_i)'A_0'A_0(\boldsymbol{y}-\boldsymbol{Y}_i)}} \\
\leq&2\sqrt{\lambda_1\left((A_0'A_0)^{-1}(\hat{A}_m - A_0)'(\hat{A}_m - A_0)\right)} \quad \text{(where } \lambda_1(B) \text{ is the largest eigenvalue of } B\text{)}
\end{aligned}
$$

Since $\lambda_1(B)$ is a continuous function with respect to B and $\hat{A}_m \overset{a.s.}{\to} A_0$ as $m \to \infty$, for any given $\epsilon > 0$, there exists some m_1 such that, for all $m \geq m_1$,

$$\sup_{\boldsymbol{y}\in\Re^P}\left|SPD_{F_m^*}(\hat{A}_m(\boldsymbol{y}-\hat{\boldsymbol{\theta}}_m)) - SPD_{G_m}(A_0(\boldsymbol{y}-\boldsymbol{\theta}_0))\right| \leq \epsilon/4,$$

along almost all $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ sequences. Based on the uniform convergence of sample spatial depth, there exists some m_2 such that, for all $m \geq m_2$,

$$\sup_{\boldsymbol{y}\in\Re^P}|SPD_{G_m}(A_0(\boldsymbol{y}-\boldsymbol{\theta}_0)) - SPD_G(A_0(\boldsymbol{y}-\boldsymbol{\theta}_0))| \leq \epsilon/4,$$

along almost all $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ sequences. Therefore, for all $m \geq \max(m_1, m_2)$ and any given

$\boldsymbol{x} \in \Re^p$, we have

$$\begin{aligned}
&\{\boldsymbol{Y} : SPD_G(A_0(\boldsymbol{Y}-\boldsymbol{\theta}_0)) \le SPD_G(A_0(\boldsymbol{x}-\boldsymbol{\theta}_0)) - \epsilon\} \\
\subseteq &\{\boldsymbol{Y} : SPD_{F_m^*}(\hat{A}_m(\boldsymbol{Y}-\hat{\boldsymbol{\theta}}_m)) \le SPD_{F_m^*}(\hat{A}_m(\boldsymbol{x}-\hat{\boldsymbol{\theta}}_m))\} \\
\subseteq &\{\boldsymbol{Y} : SPD_G(A_0(\boldsymbol{Y}-\boldsymbol{\theta}_0)) \le SPD_G(A_0(\boldsymbol{x}-\boldsymbol{\theta}_0)) + \epsilon\}. \qquad \text{(A.4)}
\end{aligned}$$

Define

$$R_{F_m^*}(\hat{A}_m(\boldsymbol{x}-\hat{\boldsymbol{\theta}}_m)) = \#\{\boldsymbol{Y}_j \mid SPD_{F_m^*}(\hat{A}_m(\boldsymbol{Y}_j-\hat{\boldsymbol{\theta}}_m)) \le SPD_{F_m^*}(\hat{A}_m(\boldsymbol{x}-\hat{\boldsymbol{\theta}}_m))\}/m,$$

$$R_G(A_0(\boldsymbol{x}-\boldsymbol{\theta}_0)) = P\{SPD_G(A_0(\boldsymbol{Y}-\boldsymbol{\theta}_0)) \le SPD_G(A_0(\boldsymbol{x}-\boldsymbol{\theta}_0)) | \, \boldsymbol{Y} \sim F_0\}.$$

By letting ϵ in (A.4) tend to 0, it is not difficult to see that $R_{F_m^*}(\hat{A}_m(\boldsymbol{x}-\hat{\boldsymbol{\theta}}_m))$ converges to $R_G(A_0(\boldsymbol{x}-\boldsymbol{\theta}_0))$ for almost all fixed $\boldsymbol{x}$ along almost all $\{\boldsymbol{Y}_1, ..., \boldsymbol{Y}_m\}$ sequences. Following Liu and Singh (1993), when the process is in control, $R_G(A_0(\boldsymbol{X}_n - \boldsymbol{\theta}_0))$ follows Uniform$[0,1]$. Therefore, $R_{F_m^*}(\boldsymbol{X}_n^*) = R_{F_m^*}(\hat{A}_m(\boldsymbol{X}_n - \hat{\boldsymbol{\theta}}_m))$ asymptotically follows Uniform$[0,1]$, and our DD-CUSUM is asymptotically distribution-free. ■

Appendix B

Additional Power Comparison Results for SS-CUSUM

Table B.1: Power Comparison for One Direction Location Shift: Cauchy_5

p=5	SS-CUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	203(1.64)	200(1.82)	202(1.94)	197(1.94)	200(1.95)
0.5	83.8(0.68)	100(0.93)	120(1.15)	133(1.31)	150(1.50)
1.0	42.6(0.26)	44.5(0.34)	54.0(0.46)	64.5(0.58)	78.1(0.74)
1.5	28.7(0.15)	26.4(0.17)	29.3(0.22)	34.2(0.29)	41.1(0.37)
2.0	22.6(0.11)	19.4(0.11)	20.0(0.13)	21.7(0.16)	25.1(0.21)
2.5	19.0(0.09)	15.8(0.08)	15.2(0.09)	16.1(0.10)	17.6(0.13)
3.0	16.8(0.07)	13.6(0.06)	12.6(0.07)	12.8(0.08)	13.7(0.08)
p=5	MCUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	2440(24.57)	1847(18.77)	1477(14.86)	1281(12.82)	1077(10.90)
0.5	2410(23.83)	1865(18.38)	1509(15.10)	1249(12.55)	1093(10.89)
1.0	2444(24.50)	1833(18.49)	1484(14.58)	1284(12.72)	1095(10.95)
1.5	2420(24.28)	1857(18.23)	1509(15.16)	1264(12.84)	1083(10.87)
2.0	2394(24.42)	1827(18.39)	1501(15.14)	1253(12.56)	1112(11.03)
2.5	2418(24.34)	1869(19.15)	1500(14.83)	1266(12.87)	1082(10.83)
3.0	2389(23.66)	1856(18.46)	1518(15.20)	1257(12.38)	1096(11.08)
p=5	ARCUSUM First				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	201(2.68)	197(2.33)	199(2.23)	199(2.09)	202(2.17)
0.5	172(1.71)	160(1.57)	153(1.53)	147(1.46)	146(1.46)
1.0	119(1.13)	125(1.24)	122(1.20)	116(1.15)	120(1.19)
1.5	89.9(0.80)	105(1.04)	107(1.07)	103(1.02)	105(1.05)
2.0	75.1(0.62)	92.9(0.91)	99.6(0.98)	97.5(0.98)	97.1(0.97)
2.5	66.3(0.52)	84.5(0.81)	94.0(0.93)	92.7(0.93)	92.2(0.93)
3.0	60.5(0.46)	80.1(0.77)	90.9(0.91)	89.0(0.89)	88.9(0.90)
p=5	ARCUSUM Last				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	199(2.72)	201(2.40)	205(2.29)	196(2.08)	201(2.10)
0.5	192(1.87)	220(2.18)	245(2.43)	258(2.58)	286(2.85)
1.0	70.9(0.55)	88.2(0.77)	109(1.01)	129(1.23)	159(1.53)
1.5	38.3(0.25)	42.8(0.32)	49.2(0.39)	57.4(0.48)	69.3(0.61)
2.0	26.5(0.15)	27.2(0.17)	29.7(0.21)	32.9(0.24)	37.4(0.29)
2.5	20.6(0.11)	20.4(0.11)	21.2(0.13)	23.0(0.15)	25.2(0.18)
3.0	17.6(0.09)	16.8(0.09)	16.9(0.09)	17.9(0.11)	19.3(0.12)

Table B.2: Power Comparison for One Direction Location Shift: $t_{5,3}$

p=5	SS-CUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	203(1.68)	199(1.82)	201(1.88)	198(1.88)	202(1.94)
0.5	36.8(0.22)	36.6(0.27)	43.9(0.37)	53.6(0.48)	65.8(0.61)
1.0	18.9(0.09)	16.0(0.08)	15.7(0.09)	16.8(0.11)	19(0.14)
1.5	13.9(0.05)	11.0(0.04)	9.9(0.04)	9.7(0.05)	10.1(0.06)
2.0	11.4(0.04)	8.9(0.03)	7.8(0.03)	7.4(0.03)	7.3(0.03)
2.5	10.2(0.04)	7.8(0.03)	6.8(0.02)	6.2(0.02)	6.0(0.02)
3.0	9.3(0.03)	7.2(0.02)	6.1(0.02)	5.6(0.02)	5.3(0.02)
p=5	MCUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	239(1.82)	237(2.15)	212(2.00)	178(1.70)	144(1.40)
0.5	65.5(0.37)	64.3(0.43)	68.9(0.53)	78.4(0.70)	84.0(0.79)
1.0	34.0(0.15)	29.8(0.14)	27.9(0.14)	28.3(0.16)	30.3(0.20)
1.5	23.0(0.09)	19.2(0.08)	17.2(0.07)	16.5(0.07)	16.0(0.08)
2.0	17.6(0.06)	14.1(0.05)	12.4(0.05)	11.4(0.05)	10.8(0.04)
2.5	14.1(0.05)	11.3(0.04)	9.7(0.04)	8.8(0.03)	8.2(0.03)
3.0	11.9(0.04)	9.4(0.03)	8.0(0.03)	7.2(0.02)	6.7(0.02)
p=5	ARCUSUM First				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	205(2.75)	198(2.34)	202(2.25)	197(2.10)	197(2.12)
0.5	121(1.15)	140(1.39)	141(1.43)	136(1.35)	137(1.36)
1.0	61.2(0.45)	87.3(0.83)	104(1.05)	106(1.10)	109(1.12)
1.5	46.1(0.28)	62.7(0.51)	87.4(0.87)	93.3(0.95)	90.6(0.96)
2.0	40.0(0.23)	52.2(0.38)	77.1(0.75)	87.2(0.89)	90.6(0.92)
2.5	37.7(0.20)	47.2(0.33)	70.0(0.68)	82.3(0.85)	86.3(0.87)
3.0	36.4(0.19)	45.8(0.30)	67.1(0.63)	80.2(0.83)	85.1(0.86)
p=5	ARCUSUM Last				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	205(2.74)	204(2.45)	203(2.31)	195(2.11)	199(2.11)
0.5	80.6(0.66)	99.9(0.90)	121(1.16)	137(1.34)	170(1.65)
1.0	27.2(0.16)	28.1(0.18)	30.5(0.21)	34.0(0.25)	38.6(0.31)
1.5	16.3(0.08)	15.5(0.08)	15.5(0.08)	16.1(0.09)	17.3(0.10)
2.0	12.4(0.05)	11.4(0.05)	11.1(0.05)	11.1(0.05)	11.5(0.06)
2.5	10.5(0.04)	9.5(0.04)	9.1(0.04)	9.0(0.04)	9.2(0.04)
3.0	9.4(0.04)	8.5(0.03)	8.2(0.03)	8.0(0.03)	8.1(0.03)

Table B.3: Power Comparison for One Direction Location Shift: $\chi^2_{5,1}$

p=5	SS-CUSUM				
b	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$
0	199(1.65)	202(1.87)	196(1.87)	184(1.75)	172(1.67)
0.5	29.4(0.15)	28.1(0.17)	33.0(0.25)	42.2(0.36)	52.3(0.48)
1.0	15.2(0.06)	12.1(0.05)	11.3(0.05)	11.6(0.05)	12.8(0.07)
1.5	11.6(0.04)	8.9(0.03)	7.9(0.02)	7.3(0.02)	7.3(0.03)
2.0	10.1(0.03)	7.6(0.02)	6.6(0.02)	6.0(0.02)	5.8(0.02)
2.5	9.2(0.03)	7.0(0.02)	5.9(0.02)	5.4(0.01)	5.1(0.01)
3.0	8.7(0.03)	6.6(0.02)	5.6(0.01)	5.0(0.01)	4.7(0.01)
p=5	MCUSUM				
b	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$
0	207(1.48)	203(1.68)	189(1.69)	166(1.55)	133(1.27)
0.5	52.9(0.29)	48.1(0.31)	47.6(0.35)	50.9(0.42)	51.7(0.46)
1.0	27.3(0.12)	22.9(0.11)	20.8(0.11)	20.3(0.11)	19.9(0.12)
1.5	18.6(0.07)	15.0(0.06)	13.0(0.06)	12.3(0.06)	11.6(0.05)
2.0	14.0(0.05)	11.2(0.04)	9.7(0.04)	8.8(0.04)	8.1(0.03)
2.5	11.4(0.04)	9.0(0.03)	7.6(0.03)	6.8(0.03)	6.3(0.02)
3.0	9.6(0.03)	7.5(0.03)	6.3(0.02)	5.6(0.02)	5.1(0.02)
p=5	ARCUSUM First				
b	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$
0	203(2.25)	202(2.18)	201(2.03)	198(2.00)	203(2.03)
0.5	47.6(0.26)	66.2(0.47)	96.7(0.92)	116(1.14)	117(1.16)
1.0	36.3(0.20)	44.3(0.29)	58.7(0.53)	68.0(0.68)	71.1(0.70)
1.5	36.1(0.20)	45.0(0.29)	59.9(0.54)	69.7(0.71)	71.3(0.71)
2.0	36.5(0.20)	44.8(0.29)	60.3(0.53)	68.8(0.69)	71.3(0.70)
2.5	36.0(0.20)	45.0(0.30)	59.2(0.53)	68.1(0.68)	70.4(0.70)
3.0	36.2(0.20)	44.8(0.29)	59.8(0.54)	67.4(0.67)	71.3(0.72)
p=5	ARCUSUM Last				
b	$k=0.1$	$k=0.2$	$k=0.3$	$k=0.4$	$k=0.5$
0	198(2.86)	197(2.32)	205(2.30)	201(2.14)	200(2.07)
0.5	128(1.17)	128(1.20)	133(1.29)	139(1.39)	142(1.41)
1.0	22.8(0.12)	21.3(0.12)	21.8(0.14)	22.9(0.16)	23.5(0.18)
1.5	15.6(0.07)	13.9(0.07)	13.3(0.07)	13.2(0.08)	13.4(0.08)
2.0	12.2(0.05)	10.6(0.05)	9.9(0.05)	9.5(0.05)	9.5(0.03)
2.5	10.4(0.04)	8.9(0.04)	8.2(0.04)	7.8(0.03)	7.7(0.03)
3.0	9.2(0.04)	7.9(0.03)	7.3(0.03)	6.8(0.03)	6.7(0.03)

Table B.4: Power Comparison for One Direction Location Shift: $\text{Gamma}_{5,1}$

p=5	SS-CUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	201(1.63)	198(1.81)	197(1.89)	187(1.80)	183(1.77)
0.5	25.4(0.12)	22.8(0.12)	24.9(0.17)	30.3(0.24)	38.1(0.34)
1.0	13.5(0.05)	10.6(0.04)	9.5(0.03)	9.2(0.04)	9.7(0.05)
1.5	10.5(0.03)	8.0(0.02)	6.9(0.02)	6.3(0.02)	6.0(0.02)
2.0	9.2(0.03)	7.0(0.02)	6.0(0.02)	5.4(0.01)	5.0(0.01)
2.5	8.6(0.03)	6.5(0.02)	5.5(0.01)	4.9(0.01)	4.5(0.01)
3.0	8.2(0.03)	6.2(0.02)	5.3(0.01)	4.6(0.01)	4.3(0.01)
p=5	MCUSUM				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	200(1.40)	196(1.64)	182(1.61)	169(1.55)	141(1.34)
0.5	37.6(0.18)	31.9(0.18)	30.0(0.18)	30.4(0.21)	31.1(0.23)
1.0	19.3(0.08)	15.6(0.07)	13.5(0.06)	12.7(0.06)	11.9(0.06)
1.5	13.0(0.05)	10.4(0.04)	8.8(0.03)	8.0(0.03)	7.3(0.03)
2.0	9.9(0.03)	7.8(0.03)	6.6(0.02)	5.8(0.02)	5.3(0.02)
2.5	8.0(0.03)	6.3(0.02)	5.3(0.02)	4.7(0.02)	4.2(0.01)
3.0	6.7(0.02)	5.3(0.02)	4.4(0.01)	3.9(0.01)	3.5(0.01)
p=5	ARCUSUM First				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	203(2.65)	203(2.38)	201(2.18)	204(2.16)	200(2.13)
0.5	45.8(0.26)	68.9(0.54)	127(1.29)	178(1.87)	204(2.10)
1.0	33.3(0.17)	41.3(0.25)	68.2(0.59)	109(1.14)	128(1.35)
1.5	33.5(0.17)	41.3(0.26)	67.6(0.59)	107(1.14)	128(1.32)
2.0	33.3(0.17)	42.2(0.26)	67.9(0.58)	108(1.15)	128(1.33)
2.5	33.2(0.17)	41.3(0.25)	68.0(0.59)	107(1.12)	129(1.35)
3.0	33.5(0.17)	41.3(0.25)	68.1(0.59)	107(1.13)	129(1.33)
p=5	ARCUSUM Last				
b	$k = 0.1$	$k = 0.2$	$k = 0.3$	$k = 0.4$	$k = 0.5$
0	201(2.77)	204(2.48)	197(2.17)	201(2.19)	198(2.10)
0.5	65.0(0.51)	70.6(0.61)	73.4(0.66)	81.5(0.76)	87.1(0.81)
1.0	17.1(0.08)	15.7(0.08)	15.4(0.09)	15.1(0.09)	15.6(0.10)
1.5	11.7(0.05)	10.3(0.05)	9.6(0.04)	9.3(0.04)	9.2(0.04)
2.0	9.5(0.04)	8.2(0.03)	7.5(0.03)	7.2(0.03)	7.0(0.03)
2.5	8.4(0.03)	7.2(0.03)	6.6(0.02)	6.3(0.02)	6.1(0.03)
3.0	7.8(0.03)	6.7(0.02)	6.1(0.02)	5.8(0.02)	5.6(0.03)

Appendix C

Derivations of eBLUPs for New Subjects

We denote a new subject with s measurements observed at time points $\mathbf{t} = (t_1, \cdots, t_s)'$ by $\boldsymbol{y}^* = (y_1^*, \cdots, y_s^*)'$. Since we do not know his group label, we will modify the model (4.9) to

$$y_j^* = \sum_{k=0}^{q} (\beta_k + \delta_k W + b_k)\Psi_k(t_j) + \epsilon_j \qquad j = 1, \cdots, s,$$

where W is a Bernoulli random variable with probability π being the prevalence proportion of group 1. Define $\boldsymbol{\beta} = (\beta_0, \cdots, \beta_q)'$ and $\boldsymbol{\delta} = (\delta_0, \cdots, \delta_q)'$ as the fixed effects. Also Define $\mathbf{X}^* = (\boldsymbol{\Psi}_1, \cdots, \boldsymbol{\Psi}_s)'$ and $\mathbf{Z}^* = \begin{bmatrix} \mathbf{X}^* & \mathbf{X}^*\boldsymbol{\delta} \end{bmatrix}$, where $\boldsymbol{\Psi}_j = (\Psi_0(t_j), \Psi_1(t_j), \cdots, \Psi_q(t_j))'$ is the basis functions for the new patient at time j. Further define $\mathbf{b}^* = (b_0, b_1, \cdots, b_q, W)'$ as the vector of random effects and $\boldsymbol{\epsilon}^* = (\epsilon_1, \cdots, \epsilon_s)'$ as the random errors. Then the NME model

for the new patient can be written as

$$\boldsymbol{y}^* = \mathbf{X}^*\boldsymbol{\beta} + \mathbf{Z}^*\mathbf{b}^* + \boldsymbol{\epsilon}^*$$

If we write $\tilde{\boldsymbol{y}} = (\boldsymbol{y}', \boldsymbol{y}^{*'})'$, then the BLUP of $\boldsymbol{\omega} = (\beta_0+\delta_0 W+b_0, \cdots, \beta_q+\delta_q W+b_q)'$ can be expressed as

$$\boldsymbol{\mu}_\omega + \mathbf{V}'_{\tilde{\boldsymbol{y}}\boldsymbol{\omega}}\mathbf{V}^{-1}_{\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}}(\tilde{\boldsymbol{y}} - \boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}),$$

where $\boldsymbol{\mu}_{\boldsymbol{\omega}}$ is the mean of $\boldsymbol{\omega}$, $\mathbf{V}_{\tilde{\boldsymbol{y}}\boldsymbol{\omega}}$ is the covariance matrix of $(\tilde{\boldsymbol{y}}, \boldsymbol{\omega})$, $\mathbf{V}_{\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}}$ is the variance of $\tilde{\boldsymbol{y}}$, and $\boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}$ is the mean of $\tilde{\boldsymbol{y}}$. It is not difficult to find $\boldsymbol{\mu}_{\boldsymbol{\omega}}$, $\mathbf{V}_{\tilde{\boldsymbol{y}}\boldsymbol{\omega}}$, $\mathbf{V}_{\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}}$, and $\boldsymbol{\mu}_{\tilde{\boldsymbol{y}}}$, and plug them into the above equation to obtain BLUP of $\boldsymbol{\omega}$. Then by substituting elements of $\boldsymbol{\beta}$ and $\boldsymbol{\delta}$ by their eBLUPs obtained in the training sample (which are elements of $\hat{\boldsymbol{\gamma}}$ and are denoted by $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\delta}}$, respectively) and replacing $\mathbf{D}$ and σ_e^2 by their maximum likelihood estimates (denoted by $\hat{\mathbf{D}}$ and $\hat{\sigma}_e^2$, respectively), one can obtain the eBLUPs of $\boldsymbol{\omega}$ as

$$\hat{\boldsymbol{\beta}} + \pi\hat{\boldsymbol{\delta}} + \left\{\mathbf{X}^*\hat{\mathbf{D}} + \pi(1-\pi)\mathbf{X}^*\hat{\boldsymbol{\delta}}\hat{\boldsymbol{\delta}}'\right\}' \left(\mathbf{Z}^*\hat{\boldsymbol{\Lambda}}\mathbf{Z}^{*'} + \hat{\sigma}_e^2\mathbf{I}_s\right)^{-1} \left\{\boldsymbol{y}^* - \mathbf{X}^*\hat{\boldsymbol{\beta}} - \mathbf{Z}^*\boldsymbol{\pi}\right\},$$

where $\hat{\boldsymbol{\Lambda}} = \begin{bmatrix} \hat{\mathbf{D}} & \mathbf{0} \\ \mathbf{0} & \pi(1-\pi) \end{bmatrix}$, and $\boldsymbol{\pi}$ is a $(q+2)$-demensional vector with the first $q+1$ elements as 0 and the last element as π.

The eBLUPs under the LME model can be derived similarly as under the NME model. We only need to redefine $\mathbf{X}^* = \begin{bmatrix} \mathbf{1}_s & \mathbf{t} \end{bmatrix}$, where $\mathbf{1}_s$ is a vector of s ones and $\mathbf{t} = (t_1, \cdots, t_s)'$, and set $q = 1$. By following the same steps as above, one can easily show that the eBLUPs of $(\beta_0 + \delta_0 W + b_0, \beta_1 + \delta_q W + b_1)'$ (denoted by $(\hat{\alpha}_0^*, \hat{\alpha}_1^*)'$) is equal to (4.7).

Appendix D

Mixed Effects Model Based Logistic Regression Classifier Performance Evaluation

As we discussed in Section 4.2.1, the mixed effects model could overcome the difficulties that are often encountered in longitudinal data analysis, such as missing values and irregularly sampled data. Therefore, for longitudinal data classification, we propose to fit a mixed effects model to the longitudinal data and extract the subject-specific random effects which are used as features in the LR model. In this Appendix, we first use a simple example to theoretically demonstrate the advantage of the mixed effects model based LR procedure. Then simulation results are shown to compare the proposed method to the traditional observation based LR classifier. The comparisons are based on misclassification error rate (MER).

D.1 Motivation

In this section, we discuss the motivation of utilizing the subject-specific random effects extracted from LME model for classification instead of using the original observations directly. Assume that the data come from an LME model as we introduced in (4.3). Also assume that we know the true values of fixed and random effects. If we use the subject-specific intercept and slope as a feature vector for classification, we might expect a lower Misclassification Error Rate (MER) than using the original observations. This intuition is based on the observation that the variation in the original observations comes from both random effects and random error, and therefore is larger than the variation in random effects themselves. In the following we use a simple example to show this conjecture is true.

Suppose the data come from (4.3). For simplicity, we further assume that the measurements for all subjects are taken at the same n time points, as in the sepsis example. Denote the design matrix as $\mathbf{X}_i = (\mathbf{1}_n, \mathbf{t}_n)$, the common fixed effects for both groups as $\boldsymbol{\beta} = (\beta_0, \beta_1)'$, the offsets of group 1 from group 0 as $\boldsymbol{\delta} = (\delta_0, \delta_1)'$, and random effects as $\mathbf{b}_i = (b_{0i}, b_{1i})'$. Also assume that the random effects follow a bivariate normal distribution, $\mathbf{b}_i \sim N(\mathbf{0}, \mathbf{D})$, where $\mathbf{D} = \begin{bmatrix} \sigma_0^2 & 0 \\ 0 & \sigma_1^2 \end{bmatrix}$, and $\boldsymbol{\epsilon}_i \sim N(\mathbf{0}, \sigma_e^2 \mathbf{I})$. Then the optimal MER using true subject-specific intercept and slope as features can be calculated as

$$MER_1 = \boldsymbol{\Phi}(-\frac{\Delta}{2})$$

where $\boldsymbol{\Phi}(\cdot)$ is the cumulative distribution function of a standard normal distribution, and $\Delta^2 = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)' \mathbf{D}^{-1} (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_0)$ with $\boldsymbol{\mu}_0 = \boldsymbol{\beta}$ and $\boldsymbol{\mu}_1 = \boldsymbol{\beta} + \boldsymbol{\delta}$. After some simple algebra, one obtains that $\Delta^2 = \frac{\delta_0^2}{\sigma_0^2} + \frac{\delta_1^2}{\sigma_1^2}$.

On the other hand, if we use the original observations $\mathbf{Y}_i$ for classification, the optimal MER can be calculated as

$$MER_2 = \mathbf{\Phi}(-\frac{\Delta^*}{2})$$

where

$$\Delta^{*2} = (\boldsymbol{\mu}_1^* - \boldsymbol{\mu}_0^*)'\mathbf{W}^{-1}(\boldsymbol{\mu}_1^* - \boldsymbol{\mu}_0^*) \tag{D.1}$$

with $\boldsymbol{\mu}_0^* = \mathbf{X}_i\boldsymbol{\beta}$ and $\boldsymbol{\mu}_1^* = \mathbf{X}_i(\boldsymbol{\beta} + \boldsymbol{\delta})$ as the population means for group 0 and group 1, respectively, and $\mathbf{W} = \mathbf{X}_i\mathbf{D}\mathbf{X}_i' + \sigma_e^2\mathbf{I}$ as the population covariance matrix. We write $\mathbf{R} = \sigma_e^2\mathbf{I}$, and the inverse of $\mathbf{W}$ can be expressed as

$$\mathbf{W}^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{X}\mathbf{D}(\mathbf{D} + \mathbf{D}\mathbf{X}^{'}\mathbf{R}^{-1}\mathbf{X}\mathbf{D})^{-1}\mathbf{D}\mathbf{X}^{'}\mathbf{R}^{-1}$$

After some algebra, we are able to find that

$$\Delta^{*2} = \frac{1}{C}((n\sigma_1^2\sum_{i=1}^n t_i^2 + n\sigma_e^2 - \sigma_1^2(\sum_{i=1}^n t_i)^2)\delta_0^2 + 2\sigma_e^2\sum_{i=1}^n t_i\delta_0\delta_1 + (n\sigma_0^2\sum_{i=1}^n t_i^2 + \sigma_e^2\sum_{i=1}^n t_i^2 - \sigma_0^2(\sum_{i=1}^n t_i)^2)\delta_1^2)$$

where $C = (n\sigma_0^2 + \sigma_e^2)(\sigma_1^2\sum_{i=1}^n t_i^2 + \sigma_e^2) - \sigma_0^2\sigma_1^2(\sum_{i=1}^n t_i)^2$. Therefore, to compare MER_1 and MER_2, we need to compare Δ^2 and Δ^{*2}. Since $\mathbf{\Phi}(\cdot)$ is a monotonically increasing function, to prove that $MER_1 < MER_2$, it suffices to show $\Delta^2 > \Delta^{*2}$.

$$\begin{aligned}
&\Delta^2 > \Delta^{*2} \\
\iff &\sigma_1^4\sum_{i=1}^n t_i^2\delta_0^2 + \sigma_1^2\sigma_e^2\delta_0^2 + n\sigma_0^4\delta_1^2 + \sigma_0^2\sigma_e^2\delta_1^2 - 2\sigma_0^2\sigma_1^2(\sum_{i=1}^n t_i)\delta_0\delta_1 > 0 \\
\iff &\sigma_1^4\delta_0^2\mathbf{t}_n^{'}\mathbf{t}_n + \sigma_0^4\delta_1^2\mathbf{1}_n^{'}\mathbf{1}_n - 2\sigma_0^2\sigma_1^2\delta_0\delta_1\mathbf{1}_n^{'}\mathbf{t}_n + \sigma_e^2(\sigma_0^2\delta_1^2 + \sigma_1^2\delta_0^2) > 0 \\
\iff &(\sigma_1^2\delta_0\mathbf{t}_n - \sigma_0\delta_1\mathbf{1})^{'}(\sigma_1^2\delta_0\mathbf{t}_n - \sigma_0\delta_1\mathbf{1}) + a > 0 \\
\iff &\boldsymbol{\lambda}^{'}\boldsymbol{\lambda} + a > 0
\end{aligned}$$

Since $a = \sigma_e^2(\sigma_0^2\delta_1^2 + \sigma_1^2\delta_0^2) > 0$, it follows that $\Delta^2 > \Delta^{*2}$, and thereby $MER_1 < MER_2$. Consequently, we can see that if the subject-specific intercept and slope are available, using them as the feature vector in a classifier would lead to a better performance than using the original measurements.

D.2 MER Comparison

D.2.1 LME Model Based Classification

In this section, we compare the performance of the LME model based LR classifier with the original observation based LR classifier with respect to MER. The data we use here are the same as the ones we generated in Section 4.3 for LME model performance evaluation.

We implement the LME based LR classifier as in Sections 4.2.2. For the original observation based method, we use mean imputation to fill in the missing values. We find the group sample mean for each time point, and substitute the missing value at that time point with the corresponding sample mean. Then for time point t, we can use all the observations up to time t in the training sample to build an LR classifier. For the testing sample, we use the original observations as the feature vector and predict on the group label based on the built classifier. The performance of the two classifiers are compared based on MER, which indicates how many mistakes we make on predicting group labels for test set based on the classifiers we built. We run the simulation 100 times, and get 100 MERs for each of the classifiers.

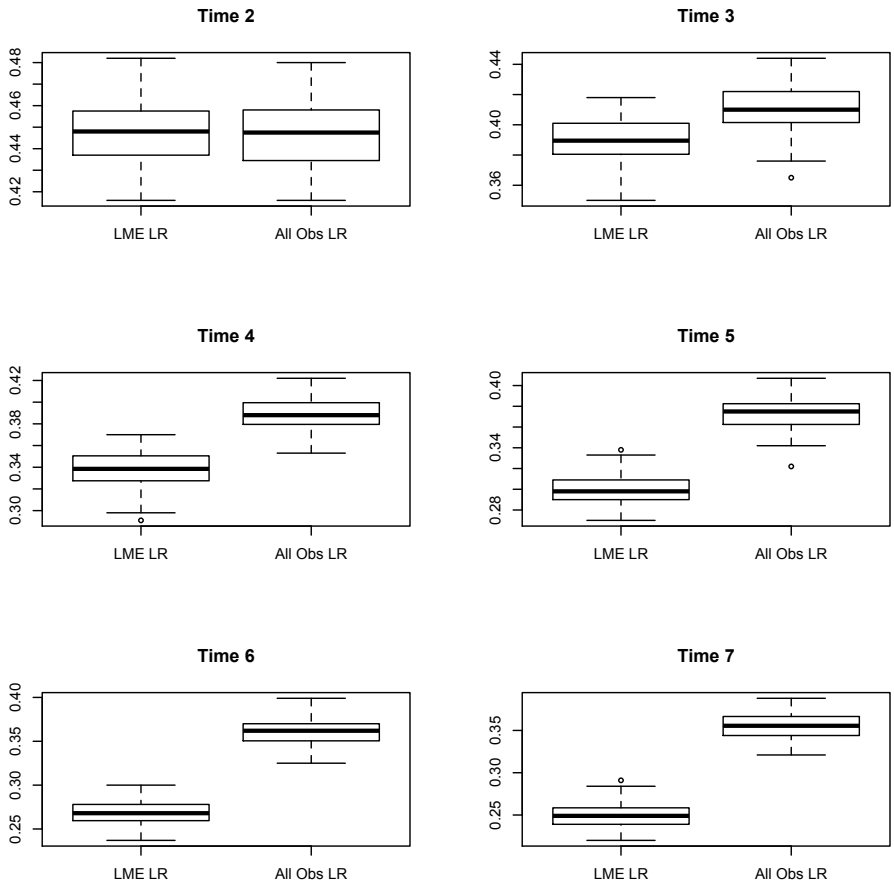


Figure D.1: MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for V-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$

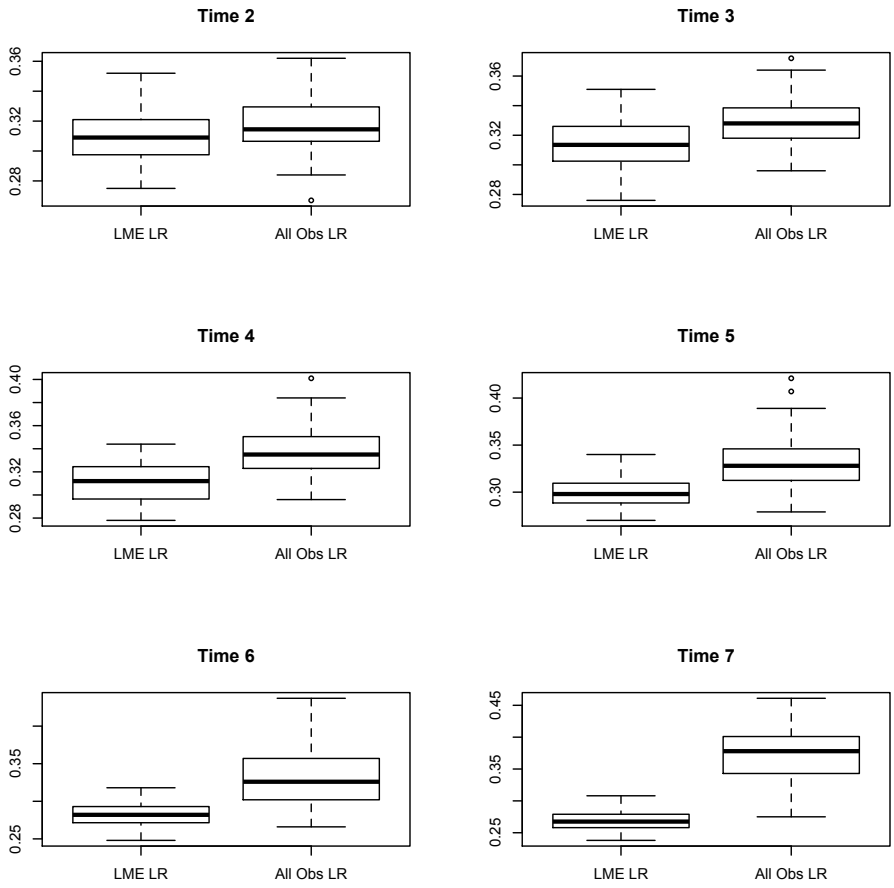


Figure D.2: MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for reverse V-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$

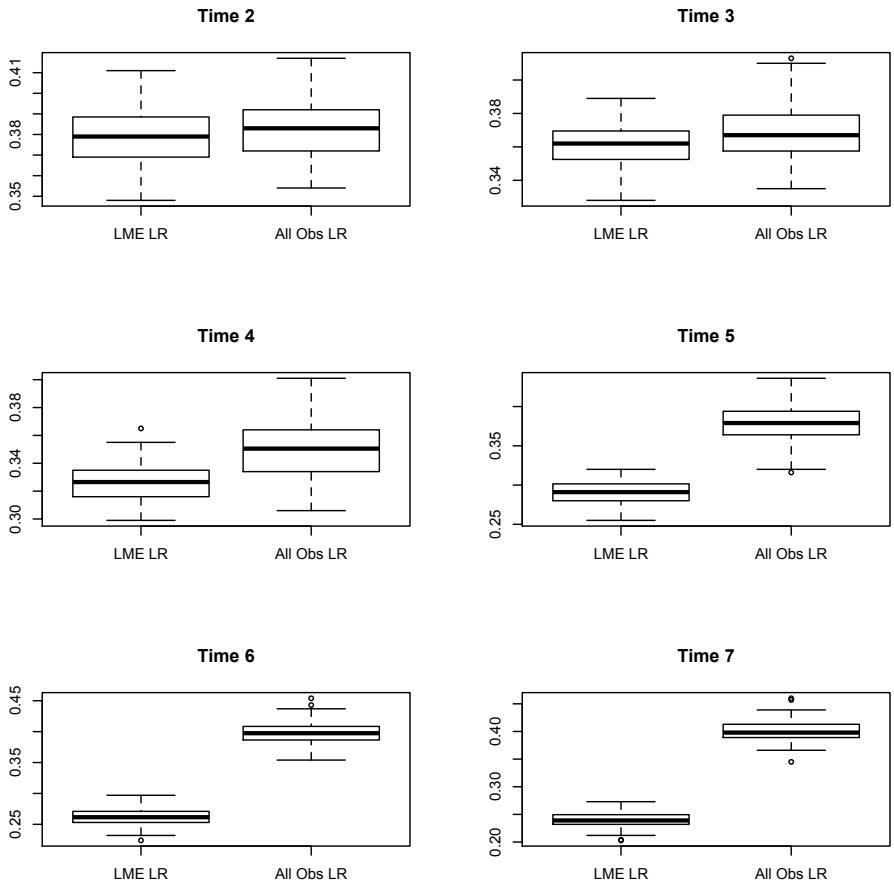


Figure D.3: MER Comparison for the LME based Classifiers and the Original Observation based Classifiers for X-shape. $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$

Figures D.1, D.2, and D.3 show the results on comparisons of MERs for the two classifiers. In each of the figures, there are six plots, corresponding to the six time points except for time point 1. And for each of the plot, there are two boxplots, corresponding to LME model based LR ("LME LR") and original observation based LR ("All Obs LR"), respectively. The results shown in the figures are all based on one simulation setting where $\sigma_0^2 = 0.6$, $\sigma_1^2 = 0.4$, and $\sigma_e^2 = 0.2$. The results from other simulation settings are similar to the results shown. From the figures we can see that MERs for LME based method get smaller rapidly as time goes by. This is intuitive since the classifiers are able to use more information at a later time point. However, the MERs for observation based method do not decrease as fast. As a result, the LME based LR classifier has a much lower MER than original observation based LR classifier at later time points. And the difference margin gets larger as more time points are considered. This is due to the fact that under MAR missing mechanism, LME based method is robust while method based on observation with mean imputation is not. At a later time point, more missing values are introduced. Hence mean imputation introduces more biases to observation based method, and ultimately distorts its performance.

D.2.2 NME Model Based Classification

In this section, we compare the performance of the NME model based LR classifier and the original observation based LR model based on MER. We use the data simulated in Section 4.3 for NME model performance evaluation.

The implementation of the NME model based LR classifier is introduced in Section

4.2.2. We use B-spline basis functions with $q = 3$ to fit the NME model. For the observation based method, the implementation is exactly the same as in section D.2.1. Figure D.4 shows the comparison between the NME model based LR and observation based LR using $\sigma^2 = 4$. Other simulation settings would give similar results. From the figure we can see that the result is quite like what we've observed in Section D.2.1. The NME model based LR performs much better than observation based method. The advantage gets larger as time points increases, since more missing values are included.

Through our simulation study we were able to show that the mixed effects model based LR classifier is compared favorably to the conventional method which uses the original observations as feature vectors.

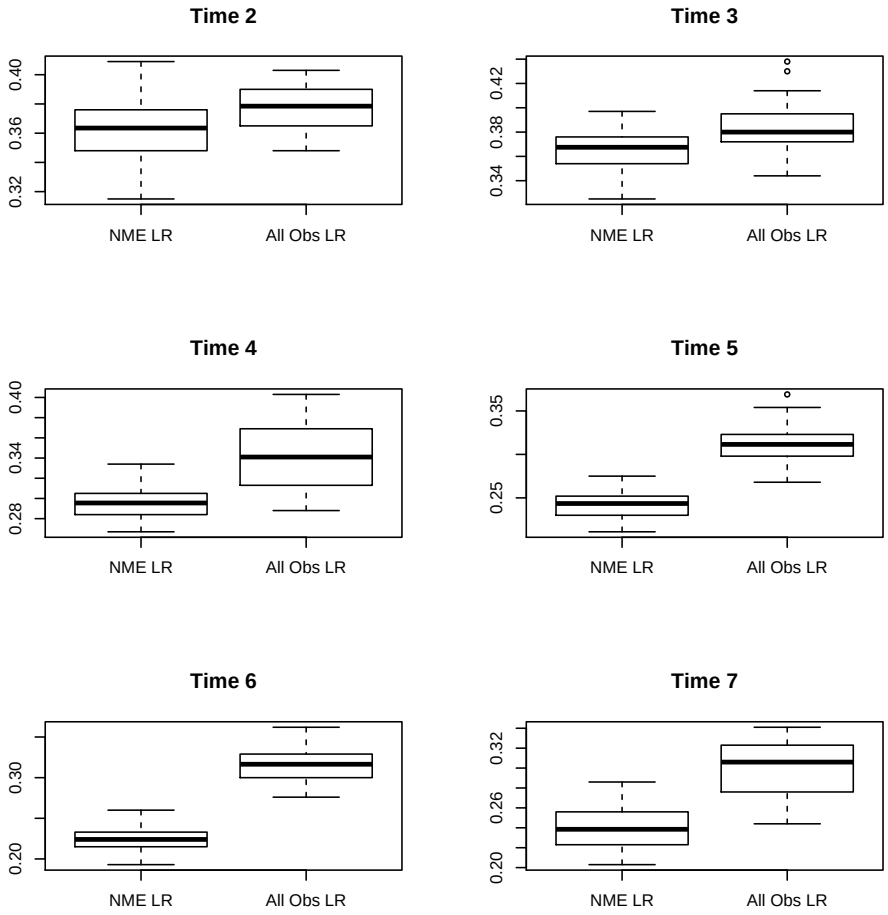


Figure D.4: MER Comparison for the NME based Classifier and the Original Observation based Classifier for $\sigma^2 = 4$