Change Detection and Analysis of Data with Heterogeneous Structures

Shuyu Chu

Dissertation submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

in

Statistics

Xinwei Deng, Co-chair

Achla Marathe, Co-chair

Marion R. Reynolds

Hongxiao Zhu

July 12, 2017

Blacksburg, Virginia

Keywords: Adaptive network lasso, Gaussian process, generalized likelihood ratio, logistic regression, mixed-type observation, particle filter, robustness, spectral mixture kernels, State space model, thermal image data

Change Detection and Analysis of Data with Heterogeneous Structures

Shuyu Chu

(ABSTRACT)

Heterogeneous data with different characteristics are ubiquitous in the modern digital world. For example, the observations collected from a process may change on its mean or variance. In numerous applications, data are often of mixed types including both discrete and continuous variables. Heterogeneity also commonly arises in data when underlying models vary across different segments. Besides, the underlying pattern of data may change in different dimensions, such as in time and space. The diversity of heterogeneous data structures makes statistical modeling and analysis challenging.

Detection of change-points in heterogeneous data has attracted great attention from a variety of application areas, such as quality control in manufacturing, protest event detection in social science, purchase likelihood prediction in business analytics, and organ state change in the biomedical engineering. However, due to the extraordinary diversity of the heterogeneous data structures and complexity of the underlying dynamic patterns, the change-detection and analysis of such data is quite challenging.

This dissertation aims to develop novel statistical modeling methodologies to analyze four types of heterogeneous data and to find change-points efficiently. The proposed approaches have been applied to solve real-world problems and can be potentially applied to a broad range of areas.

Change Detection and Analysis of Data with Heterogeneous Structures

Shuyu Chu

(General Audience Abstract)

Heterogeneous data with different characteristics are ubiquitous in the modern digital world. Detection of change-points in heterogeneous data has attracted great attention from a variety of application areas, such as quality control in manufacturing, protest event detection in social science, purchase likelihood prediction in business analytics, and organ state change in the biomedical engineering. However, due to the extraordinary diversity of the heterogeneous data structures and complexity of the underlying dynamic patterns, the change-detection and analysis of such data is quite challenging.

This dissertation focuses on modeling and analysis of data with heterogeneous structures. Particularly, four types of heterogeneous data are analyzed and different techniques are proposed in order to find change-points efficiently. The proposed approaches have been applied to solve real-world problems and can be potentially applied to a broad range of areas.

Acknowledgments

I would like to express my sincere gratitude to my PhD advisors, Drs. Xinwei Deng and Achla Marathe, who have offered invaluable guidance, encouragement and support throughout my graduate career. I also thank Dr. Deng for his selfless help on editing and finishing this dissertation and Dr. Marathe's generous support throughout my graduate study.

I would also like to extend my gratitude to Dr. Marion R. Reynolds and Dr. Hongxiao Zhu, who were gracious enough to serve on my committee. Their instructive advices and suggestions are greatly appreciated. I give my special thank to Professor Emeritus Reynolds, who introduced me to the interesting world of statistics, and keeps inspiring and encouraging me .

I also would like to thank all faulty, staff and students in NDSSL (Network Dynamics and Simulation Science Laboratory) for their support and help. I have been fortunate enough to join NDSSL since 2013. It was a very pleasant working experience with all the group members, where I learned a lot.

Finally, I want to thank my family, who provided constant support during many personal challenges.

Contents

List of Figures

List of Tables

Chapter 1 Introduction

Heterogeneous data is ubiquitous in modern statistical studies. One type of commonly encountered heterogeneity is that observations collected from a process do not follow a specified distribution with a constant mean, which often implies that there is a mean shift. Another type of heterogeneity comes from the mixed-type observations, where discrete observations are often collected together with continuous observations. Heterogeneity also commonly occurs where data points under different segments have heterogeneous model structures. Besides, the underlying pattern of data changes frequently in time, space and other dimensions.

Change-point detection is an important task for monitoring the performance of systems with heterogeneous data. The statistical analysis of change-point detection has been developed in decades in a wide range of disciplines including manufacturing processing (Basseville and Nikiforov, 1993), nuclear power plant control (Hwang et al., 2008; Ge and Smyth, 2000), bioinformatics (Lio and Vannucci, 2000; Erdman and Emerson, 2008), malware detection (Young and Kuo, 2001), economics (Saniga, 1989; Spokoiny, 2009), remote sensing (Singh, 1989), climate science (Hansen et al., 2012), and business analytics (Xue et al., 2015).

The change-point detection problems considered in this dissertation are the change points dividing dataset into distinct homogeneous segments. When the data is time-related, the change-points are time locations. Otherwise, the "change-points" can be generalized as underlying boundaries that clustering data into different segments. One type of changes occurs in a parameter from the original value to some other value in a process, which is called a shift. Another type can be the change in model structures or coefficients. Generally, change-point analysis can be categorized by the parametric versus non-parametric, offline versus online, and frequentist versus Bayesian approaches. This dissertation restricts its attention to the parametric methods: an online frequentist quality control chart in Chapter 2; a Bayesian approach through latent process in Chapter 3; an adaptive network lasso approach for logistic

regression clustering in Chapter 4; and a Gaussian process based modeling for spatio-temporal observations in Chapter 5.

The control chart technique in the statistical process control (SPC) is one widely-used frequentist method for process monitoring and change-point detection. A common problem in the process monitoring is the detection of changes in the process mean. While there have been many control charts developed to monitor the process mean, the generalized likelihood ratio (GLR) charts based on a sequential likelihood ratio test show great advantages in terms of its ability to detect a wide range of shift sizes (Lai, 2001; Hawkins et al., 2003; Reynolds Jr and Lou, 2010, 2012). However, the robustness of the GLR charts under the non-normal process has not been investigated thoroughly. In addition to the control chart technique, a variety of other statistical models have been developed to conduct change-point detection, see the review paper by Eckley et al. (2011).

Different from frequentist approaches based on the likelihood ratio or penalized likelihood, the Bayesian approaches rely on the specification of a prior for the number and position of change points (Eckley et al., 2011). Among various Bayesian methods for change detection, the Bayesian method using switching state-space models (SSSM) shows competitive performance (Cappé et al., 2009; Frühwirth-Schnatter, 2006; Eckley et al., 2011). However, most of these methods only consider single-type observations which are defined as data containing either continuous or discrete variable. Little work has been proposed for the change-point detection problem with mixed-type observations, where a process contains both continuous and discrete observations.

Another scenario of change detection problem comes from data under different segments sharing different model coefficients. Conventional methods often first conduct data segmentation and then build models for each segment, which can result in inefficient and inaccurate model estimations (Xue et al., 2015). Other widely used methods in the literature include mixture models (Jung and Wickrama, 2008; Muthen, 2001) and probabilistic graphical models (PGMs) (Nylund et al., 2007; Meila and Jordan, 2000). However, those methods have difficulties in finding the true number of the underlying structures as well as converging to

the global optimal solutions. Recently, the network lasso based approach proposed by Hallac et al. (2015) aims to conduct data clustering and model fitting simultaneously. However, a nature disadvantage of lasso approach is the serious shrinkage problem of model coefficients. In addition, the ADMM (Alternating Direction Method of Multipliers) based algorithm used in Hallac et al. (2015) can not converge efficiently for the non-linear logistic regression objective function.

Besides one-dimensional observations, data with spatio-temporal features is commonly collected in many applications, such as the organ status change detection in the biomedical monitoring process. The current evaluation methods are mostly based on doctors' visual inspection or pathologists' analysis of biopsy samples, which are either subjective or invasive. In the literature, image sequence data is widely used for change detection (Radke et al., 2005; Zhou et al., 2014). However, most of those models often are either hard to interpret or overlook the spatial dependency among observations. Thus, there is little work regarding the change detection problem of organ quality by fully understanding the underlying organ dynamic structures. The main challenge comes from a lack of appropriate statistical models and the potentially expansive computational burden caused by the high-resolution image data.

1.1 Statistical Process Control & Control Charts

Statistical process control (SPC) is often applied to conduct process monitoring and change-point detection in the industrial system. The quality of a product is referred as a group of characteristics for determining its desirability under certain specified requirements. It is always desirable to obtain high and stable quality in the process, but variations always exist. There are mainly two sources of variability. One source is called common causes, which is inevitable and inherent in the process. The other source is called special causes, which can lead to changes in the process parameters related to the quality characteristics. A process is defined to be in-control if common causes are the only source of variability. On the other hand, an out-of-control process is defined if the special causes are present.

The objective of SPC is to detect the special causes of variations as quickly as possible so that the quality can be maintained or improved. Among many techniques for SPC, the control charts are the most widely used techniques. When using a control chart, samples of $n \geqslant 1$ observations are usually taken at a certain time interval d. Without loss of generality, we assume that the sampling rate n/d in terms of the number of observations per unit time is fixed and is always one observation per unit time. Then a control chart statistic is computed from the process. This statistic differs with different types of control charts and will be compared with some predetermined control limits, which are specified according to the requirements of the controlled process. A signal is given when the statistic falls outside the control limits. When the process is out-of-control, this signal indicates a change in the process, suggesting actions needed to move the process back to the in-control state. While if the signal occurs in an in-control process, it is referred as a false alarm and no action is needed.

Usually, there are two phases involved in SPC. Phase I focuses on parameter estimation and control limits construction using historical in-control process data. Using these estimates and control limits, Phase II aims to detect any changes in future observations. Many quality characteristics used in the control charts can be expressed as a continuous random variable, which is often assumed to follow a normal distribution. A common problem in the univariate normal process monitoring is to detect the changes in the process mean. In the literature, several types of control chart methods have been developed for monitoring the process mean μ under the normality assumption. The first type of control charts is the Shewhart $\overline{X}$-chart proposed by Shewhart (1931) using the sample average $\overline{X}$ as the chart statistic. These methods are the most widely used control chart because of its simplicity in the chart construction. But they are only effective if the size of the shift in μ is relatively large. The second type of methods includes the CUSUM (the cumulative sum) (Page, 1954) and EWMA (the exponential weighted moving average) (Roberts, 2000) charts, which show good performance for detecting a specified shift size of interest based on selecting the value of some tuning parameters. However, they can not work well if the actual shift size is not close to the specified one. In practice, the actual size of the shift in μ is often unknown. Thus, it is desirable to use a control chart

that will be effective over a wide range of shift sizes. A good and simple alternative is the GLR (generalized likelihood ratio) control chart based on a sequential likelihood ratio test, where the size of the parameter shift does not need to be specified but can be estimated from the process data. Examples of the GLR charts can be found in Lai (2001), Hawkins et al. (2003), Reynolds Jr and Lou (2010), Peng et al. (2015), Reynolds Jr et al. (2013), and Peng and Reynolds Jr (2014).

The GLR control charts consider the situation in which the process variable X is assumed to have a $N(\mu, \sigma^2)$ distribution. When the process is in-control, the $\mu = \mu_0$ and $\sigma = \sigma_0$, where μ_0 and σ_0^2 are known or estimated accurately during a Phase I period. This dissertation mainly considers the problem of real-time monitoring in Phase II. The likelihood ratio test of the hypothesis is: $H_0 : \mu = \mu_0$ vs. $H_1 : \mu \neq \mu_0$. In practice, it is unknown when the shift in μ will occur or the size of the shift. Thus the objective is to detect any special cause that produces a shift in μ of any size. Suppose that an independent sample of size $n \geqslant 1$ are taken from the process using a sampling interval of length d between samples. Let $\mathbf{X}_k = (X_{k1}, X_{k2}, \cdots, X_{kn})$ represent the sample obtained at sampling point k. It is assumed that all observations within and between samples are independent. When $n > 1$, it is also assumed that the n observations in a sample are taken together so that there is a negligible probability of a shift in μ occurring within a sample. After k sampling point are taken, the log likelihood ratio statistic for testing a shift in μ is,

$$R_k = \ln \quad \max_{0 \leqslant \tau < k} \quad \max_{\mu_1} \frac{L(\tau, \mu_1 | \mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k)}{L(\infty, \mu_0 | \mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k)}, \tag{1.1}$$

where $L(\tau, \mu_1 | \mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k)$ represents the likelihood function when there is a shift in μ from μ_0 to μ_1 at some time between samples τ and $\tau + 1$, where $\tau < k$ and $\mu_1 \neq \mu_0$ (with no change in σ^2 from σ_0^2). And $L(\infty, \mu_0 | \mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k)$ represents the likelihood function when there is no change in μ. The GLR chart gives an signal at sample k, if $R_k > h$, where the control limit $h > 0$ can be chosen to achieve the desired in-control performance.

Note that both the traditional control charts and the GLR charts are using on the sample

means to monitor the process mean. Their construction often assumes that the sample means follow normal distributions. However, the distribution of the process observations may not follow normal distributions in many applications. For example, if the observations follow a distribution with a heavy tail, then there will more "outliers" than what would be expected from a normal distribution. These "extreme" observations do not correspond to special causes that should be detected. Thus, using a control chart with normality assumptions will give misleading results including more frequent false alarms. Although the central limit theorem can be used for providing certain justification of the sample mean following an approximated normal, such an approximation may not work in all situation. For instance, if the sample size $n = 1$, the central limit theorem surely can not be applied appropriately. The normality assumption is also invalid when n is relatively small while the true distribution of the observations is far from normal.

To relax the normality assumption in aforementioned control charts, one straightforward solution is to use a control chart designed for a specific distribution of the observations. However, the underlying distribution of the observations is usually unknown. Even if the underlying distribution can be estimated, there may not be a chart designed for a particular distribution. Hence it is not practical for practitioners to determine an appropriate distribution as well as to develop a suitable control chart. A more feasible solution is to consider a robust chart that work well for a variety of distributions. There are several approaches dealing with non-normal process variables in the literature. The CUSUM and EWMA charts can be robust if they are tuned to detect very small shifts (Montgomery, 2004; Borror et al., 1999; Stoumbos and Sullivan, 2002; Testik et al., 2003). Several researchers including Hawkins and Olwell (2012) and Reynolds and Stoumbos (2010) propose a robust CUSUM chart based on winsorization if the original observations contain many extreme values. With winsorization, the original observations that are larger than a specified value are replaced by the specified value. However, the robustness of these charts is quite sensitive to the choice of tuning parameters. Thus they would fail to detect shifts with all sizes. A third direction is to nonparametric control charts to deal with non-normal process variables, see the review paper

by Chakraborti et al. (2001). However, many nonparametric charts require n to be reasonably large and they usually take a long time to detect the shifts in the mean, which is not desirable in practice.

As the GLR charts show great advantages in terms of its ability to detect a wide range of shift sizes, it is worth investigating how to address the robustness of the GLR charts under non-normal process. The objective of Chapter 2 is to address the robustness of the GLR chart for monitoring the process mean. I also propose several modified GLR charts to improve their robustness while retaining most of the advantages of the GLR chart.

1.2 Switching State-Space Model

In contrast to the GLR control chart as a frequentist technique for the process monitoring, various Bayesian approaches are also developed to detect change-points relying on the specification of a prior for the number and position of change-points (Eckley et al., 2011). One commonly used Bayesian method, switching state-space model (SSSM), shows competitive performance, where the Markov chain Monte Carlo (MCMC) technique is used for efficient Bayesian inference (Cappé et al., 2009; Frühwirth-Schnatter, 2006; Eckley et al., 2011).

The switching state-space model is developed from the state-space model (SSM) that is also known as hidden Markov model or latent process model. The SSM refers to a class of probabilistic graphical model that describes the probabilistic dependency between the latent state variable and the observed measurement (Koller and Friedman, 2009; Chen and Brown, 2013). The SSM usually consists of two equations, a measurement equation to link the observed variables to unobserved latent variables, and a transition equation to model the latent dynamics. Although both equations can take on any forms, the linear Gaussian system is mostly used and has been widely applied due to the development of the efficient Kalman filter algorithm (Kalman, 1960). Consequently, the corresponding SSSM is known as the conditionally linear Gaussian state-space model or a jump linear system described as follows.

For notation convenience, the capital letters are used for random variables and lowercase

letters denote their values similar to the standard convention. To model the change-points of the observed data $\{Y_n, n = 1, \ldots, T\}$, the SSSM considers a discrete latent process $\{I_n\}_{n\geqslant 1}$, where the values of I_n belong to a finite set $\mathcal{I}$. For the SSSM, a first-order Markov process with initial distribution $I_1 \sim v_\theta(\cdot)$ is used, and the transition probabilities for $n \geqslant 1$ is

$$I_{n+1}|(I_n = i) \sim f^I_{\boldsymbol{\theta}}(\cdot|i), \tag{1.2}$$

where $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ denotes all static parameters used in the SSSM and may be multidimensional. In addition, f^I_θ is in fact, a stochastic transition matrix.

The discrete latent process $\{I_n\}_{n\geqslant 1}$ is used to indicate which state that each observation belongs to. That is, a change-point occurs when two consecutive sets of observations are in different states. Conditional on $\{I_n\}_{n\geqslant 1}$, a linear Gaussian state-space model is defined as,

$$X_{n+1} = \boldsymbol{A}_{\boldsymbol{\theta}}(I_{n+1})X_n + \boldsymbol{B}_{\boldsymbol{\theta}}V_n, \tag{1.3}$$

$$Y_{n+1} = \boldsymbol{C}_{\boldsymbol{\theta}}(I_{n+1})X_n + \boldsymbol{D}_{\boldsymbol{\theta}}W_n, \tag{1.4}$$

where $\{X_n\}_{n\geqslant 1}$ is another latent Markov process initialized by a normal distribution, $X_1 \sim N(m_0, \Sigma_0)$. The observed $\{Y_n\}_{n\geqslant 1}$ are connected through the latent $\{X_n\}_{n\geqslant 1}$. Here V_n and W_n are independent and identically distributed random variables with mean zero and variance one, and $\left\{\boldsymbol{A}_{\boldsymbol{\theta}}(i), \boldsymbol{B}_{\boldsymbol{\theta}}(i), \boldsymbol{C}_{\boldsymbol{\theta}}(i), \boldsymbol{D}_{\boldsymbol{\theta}}(i); i \in \mathcal{I}\right\}$ are matrices of appropriate dimension.

The SSSM contains two types of latent processes. The latent process $\{X_n\}_{n\geqslant 1}$ aims to control the latent dynamics of observations $\{Y_n\}_{n\geqslant 1}$, while the latent process $\{I_n\}_{n\geqslant 1}$ is to model the number of changes as well as change location(s). The objective is to conduct model estimation and inference for the SSSM using a proper Baysian approach. All the latent processes and the static parameter $\boldsymbol{\theta}$ are unknown. By assigning a prior $p(\boldsymbol{\theta})$ to $\boldsymbol{\theta}$, the inference mainly focuses on the joint density,

$$p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T}, \boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}) \propto p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:T}, \boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T})p(\boldsymbol{\theta}),$$

where

$$p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T}, \boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}) = p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{i}_{1:T}).$$

Conditional on the discrete latent process $\boldsymbol{i}_{1:T}$, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{i}_{1:T})$ is a conditional multivariate Gaussian density, which can be estimated using the Kalman filter algorithm efficiently (Kalman, 1960). However, it is difficult to make inference for the discrete-valued latent process $p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T})$. The computational burden of this distribution increases exponentially as T increases, making the exact sampling of the joint distribution impossible. To overcome this difficulty, one efficient approach is to use the particle filter techniques, which is an approximation technique based on a combination of importance sampling and resampling methods (Doucet et al., 2001a; Liu, 2008; Whiteley et al., 2010). Both empirical study and theoretical investigation show that the particle filter technique can effectively approximate the target distribution. In particular, the discrete particle filter (DPF) (Fearnhead, 1998) fully takes advantage of the discrete property in the latent process. Rather than using the standard particle filter technique of important sampling, the DPF uses a random pruning mechanism to select support points from the exponentially growing sequences of discrete latent state spaces (Whiteley et al., 2010). Thus, the model estimation and inferences can be conducted by using the proposal distributions estimated from the DPF within the MCMC scheme.

Note that in the SSSM discussed above, the observed data $\boldsymbol{Y}_{1:T}$ is in fact, a Gaussian process. However, the use of Gaussian distribution may not be applicable in many applications. For example, observations may be discrete following a Poisson or Bernoulli distribution. In addition, the model structures in SSSMs may be non-linear. In this context, the conditional distribution, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{i}_{1:T})$ is no longer a Gaussion density or even does not have a closed form. Thus exact estimation using Kalman Filter techniques does not work for this situation.

In the literature, a class of approximation methods for non-linear, non-Gaussian state-space model (SSM) is established under the MCMC procedures. Several sampling methods including Gibbs sampling and the Metropolis-Hasting independence sampling, are developed to approximate the joint posterior. However, sampling from the $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$ is typically

impractical due to its high dimensionality and implicit density form (Carter and Kohn, 1994; Liesenfeld and Richard, 2008). Thus, it is natural to resort to the particle filter algorithms. The particle MCMC (PMCMC) is recently developed in the SSM by Holenstein (2009) and Andrieu et al. (2010), which aims to build high-dimensional proposals through the use of particle filters. The sequential Monte Carlo (SMC) algorithm is also developed to design those efficient high-dimensional proposal distributions by approximating the sequence of posterior densities $\left\{p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}); n \geqslant 1\right\}$ sequentially with a set of weighted samples (particles). However, it is not clear how to extend the SMC algorithm to the SSSM with discrete observations, making its extension to mixed-type observation even difficult.

Mixed-type observations are widely present in various application areas, where both continuous and discrete observations are often collected together to evaluate the system. In order to monitor the performance of the system more effectively, it is desirable to analyze these multiple measurements simultaneously taking into account their hidden association. However, little work has been done on the change-point detection problem for the mixed-type observations.

Chen and Brown (2013) mention the possible extensions of the linear Gaussian state-space model to accommodate the discrete or mixed-type observations. The copula model offers a universal framework to model statistical dependencies among continuous, discrete, or mixed-type random variables (Chen, 2013). de Leon and Wu (2011) develop a copula-based regression model for a binary and continuous observations, where a latent variable formulation is adopted for the binary observations. However, the model is specially developed for binary and continuous observation, and it is difficult to be used for other types of discrete observations. Besides, it is also not clear how to extend the state-space model and the copula model for the change-point detection problem.

Several frequentist approaches have been proposed to address the change-point problem for the mixed-type observations. Ning and Tsung (2012) develop a density-based statistical process control scheme to detect process changes, where multi-dimensional observations are transformed into a one-dimensional measurement using a local outlier factor (LOF). Qiu (2008)

converts all the mixed-type data into binary or categorical variables and their distributions are estimated using the log-linear model. Thus the change-point is found based on changes in the estimated distributions. Both approaches require a reasonable amount of high-quality in-control data to give an accurate estimation of the in-control distribution in Phase I, which might be difficult to obtain in practice. To bypass the effort of collecting high-quality data that are in-control, Chapter 3 aims to generalize the SSSM with change-point detection for mixed-type observations in a Bayesian approach. The proposed method can automatically conduct model estimation and inference by taking advantage of the particle filter algorithms.

1.3 Network Lasso Model

Another type of change detection corresponds to the changes in model structure across different segments of data. Fitting one global model on the entire data may not be appropriate in many applications, such as the house price estimation based on spatial and other features; the purchase likelihood prediction with heterogeneous clients' and products' characteristics. The network lasso method introduced by Hallac et al. (2015) shows good performance for simultaneous clustering and regression model fitting in a graph setting. The network lasso penalty encourages data with similar regression coefficients to cluster together and thus share a common model.

Consider the problem as an undirected graph $\mathcal{G}$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. The objective function we want to minimize is,

$$\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i) + \sum_{(j,k) \in \mathcal{E}} g_{jk}\left(\boldsymbol{x}_j, \boldsymbol{x}_k\right). \tag{1.5}$$

The variables are, $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \in \mathbb{R}^p$, and $N = |\mathcal{V}|$ is the total number of nodes. Specially, $\boldsymbol{x}_i \in \mathbb{R}^p$ is the variable at node i. Let f_i denote the cost function at node i, and g_{jk} denotes the cost function associated with edge (j, k), Besides, denote $E = |\mathcal{E}|$ as the total number of edges.

The network lasso mainly focuses on the scenario when f_i are convex, and $g_{jk}(\boldsymbol{x}_j, \boldsymbol{x}_k) = \lambda w_{jk}||\boldsymbol{x}_j - \boldsymbol{x}_k||_2$, with $\lambda \geqslant 0$, and user-defined weights $w_{jk} \geqslant 0$,

$$\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i) + \lambda \sum_{(j,k) \in \mathcal{E}} g_{jk}(\boldsymbol{x}_j, \boldsymbol{x}_k). \tag{1.6}$$

The edge objectives penalize differences between the variables at adjacent nodes, where the edge between nodes i and j has weight λw_{ij}. In particular, w_{ij} can be considered as the relative weights of the edges in the network, while λ is considered as the global parameter that scales the edge objective relative to the node objectives. In the other word, λ defines a trade-off for the nodes between minimizing its own objective and agreeing with its neighbors.

For a relatively large graph problem where p, $N = |\mathcal{V}|$, $E = |\mathcal{E}|$ are potentially large, the general convex optimization methods can not work well. Thus, Hallac et al. (2015) developed an algorithm based on the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Parikh and Boyd, 2014) to solve the problem efficiently.

ADMM is a well-established method for solving distributed convex optimization problems, where each individual component solves its own private objective function and passes this solution to its neighbors and repeats until the entire network converges.

To solve via ADMM, a copy of $\boldsymbol{x}_i$ is introduced, called $\boldsymbol{z}_{ij}$, at each edge (i, j). Note that the same edge also has a $\boldsymbol{z}_{ji}$, a copy of $\boldsymbol{x}_j$. We can rewrite the problem in Equation 1.6 as an equivalent problem below,

$$\begin{aligned} &\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i) + \lambda \sum_{(j,k) \in \mathcal{E}} w_{jk}||\boldsymbol{z}_{jk} - \boldsymbol{z}_{kj}||_2, \\ &\text{subject to} \quad \boldsymbol{x}_i = \boldsymbol{z}_{ij}, \quad i = 1, \ldots, N, \quad j \in N(i), \end{aligned} \tag{1.7}$$

where $N(j)$ is the set of neighbors of node j. Deriving this problem's augmented Lagrangian (Hestenes, 1969), we get,

$$L_\rho(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{u}) = \sum_{i \in \mathcal{V}} f_i(\boldsymbol{x}_i) + \sum_{(j,k) \in \mathcal{E}} (\lambda w_{jk} ||\boldsymbol{z}_{jk} - \boldsymbol{z}_{kj}||_2 - (\rho/2)(||\boldsymbol{u}_{jk}||_2^2 + ||\boldsymbol{u}_{kj}||_2^2)$$
$$+ (\rho/2)(||\boldsymbol{x}_j - \boldsymbol{z}_{jk} + \boldsymbol{u}_{jk}||_2^2 + ||\boldsymbol{x}_k - \boldsymbol{z}_{kj} + \boldsymbol{u}_{kj}||_2^2)),$$

where u is the scaled dual variable at each edge and ρ is a scalar penalty parameter that determines the trade-off between primal and dual convergence. ADMM consists of the following steps, with t denoting the iteration number,

$$\begin{aligned}
\boldsymbol{x}^{t+1} &= \underset{\boldsymbol{x}}{\text{argmin}} \quad L_\rho(\boldsymbol{x}, \boldsymbol{z}^t, \boldsymbol{u}^t), \\
\boldsymbol{z}^{t+1} &= \underset{\boldsymbol{z}}{\text{argmin}} \quad L_\rho(\boldsymbol{x}^{t+1}, \boldsymbol{z}, \boldsymbol{u}^t), \\
\boldsymbol{u}^{t+2} &= \boldsymbol{u}^t + (\boldsymbol{x}^{t+1} - \boldsymbol{z}^{t+1}).
\end{aligned}$$

The details of these 3 steps are listed below.

$\boldsymbol{x}-$Update.

In the $\boldsymbol{x}-$update, we minimize a separable sum of functions, one per node, so it can be calculated independently at each node and solved in parallel. At node i, this is,

$$\boldsymbol{x}_i^{t+1} = \underset{\boldsymbol{x}_i}{\text{argmin}} \left(f_i(\boldsymbol{x}_i) + \sum_{j \in N(i)} \rho/2 ||\boldsymbol{x}_i - \boldsymbol{z}_{ij}^t + \boldsymbol{u}_{ij}^t||_2^2 \right).$$

$\boldsymbol{z}-$Update.

The $\boldsymbol{z}-$update is separable across the edges. Note that for edge ij, we need to jointly update $\boldsymbol{z}_{ij}$ and $\boldsymbol{z}_{ji}$. This becomes,

$$\boldsymbol{z}_{ij}^{t+1}, \boldsymbol{z}_{ji}^{t+1} = \underset{\boldsymbol{z}_{ij}, \boldsymbol{z}_{ji}}{\text{argmin}} \left(\lambda w_{ij} ||\boldsymbol{z}_{ij} - \boldsymbol{z}_{ji}||_2 + (\rho/2) \left(||\boldsymbol{x}_i^{t+1} - \boldsymbol{z}_{ij} + \boldsymbol{u}_{ij}^t||_2^2 + ||\boldsymbol{x}_j^{t+1} - \boldsymbol{z}_{ji} + \boldsymbol{u}_{ji}^t||_2^2 \right) \right).$$

This problem has a closed-form analytical solution,

$$\boldsymbol{z}_{ij}^{*} = \theta(\boldsymbol{x}_i + \boldsymbol{u}_{ij}) + (1-\theta)(\boldsymbol{x}_j + \boldsymbol{u}_{ji}),$$
$$\boldsymbol{z}_{ji}^{*} = (1-\theta)(\boldsymbol{x}_i + \boldsymbol{u}_{ij}) + \theta(\boldsymbol{x}_j + \boldsymbol{u}_{ji}),$$

where,

$$\theta = \max\left(1 - \frac{\lambda w_{ij}}{\rho||\boldsymbol{x}_i + \boldsymbol{u}_{ij} - (\boldsymbol{x}_j + \boldsymbol{u}_{ji})||_2},\ 0.5\right).$$

$\boldsymbol{u}-$Update.

The $\boldsymbol{u}-$update is also edge-separable. For each variable, it looks like,

$$\boldsymbol{u}_{ij}^{t+1} = \boldsymbol{u}_{ij}^{t} + (\boldsymbol{x}_j^{t+1} - \boldsymbol{z}_{ij}^{t+1}).$$

More detailed information about stopping criteria can be found in Boyd et al. (2011). While ADMM based algorithm works perfectly when the objective function is linear, it can stick on local optima when the objective function is non-linear. Besides, the shrinkage problems in network lasso lead to biased model coefficient estimates and thus it could be suboptimal in terms of estimation risk (Zou, 2006). The inaccurate coefficients estimation directly affects the data segmentation and model prediction. To overcome those issues, I proposed an IWLS (Iteratively Weighted Least Square) based adaptive network lasso algorithm particularly designed for the logistic objective functions in Chapter 4. The adaptive edge weights are introduced to alleviate the shrinkage problem effectively. Moreover, the key idea of IWLS is to linearize the objective function and update all parameters in a simultaneous fashion to ensure global convergence.

1.4 Gaussian Process Modeling of Spatio-temporal Data

Recent development of data collection techniques enables the formation of large spatio-temporal datasets in a wide variety of fields, such as the organ quality monitoring process. Organ transplantation provides a second chance at life for thousands of people each year. The demand

for organ transplantation has rapidly increased all over the world during the last decades (Abouna, 2008). However, poor preservation and evaluation methods cause many organs to be discarded. Current evaluation methods include doctors' visual inspection and pathologists' analysis of biopsy samples. However, the first method is often biased and subjective, while the second often leads to organ damages. It is thus very crucial to develop an accurate and non-invasive method for evaluating the quality of organs. The commonly used non-invasive methods are mainly based on the thermal image data.

One prominent tool to analyze the spatio-temporal image data is the Gaussian process (GP) based statistical method (Rasmussen, 2006; Hartikainen et al., 2011). A Gaussian process (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. Given data $(\mathbf{y}, \mathbf{x})$, where $\mathbf{y} = y_1, \ldots, y_N$, are responses or dependent variables, and $\mathbf{x} = x_1, \ldots, x_N$, $x_i \in R^D$, are covariates or independent variables, each of dimension P. Assume that the responses $\boldsymbol{y}$ are generated from the covariates by an underlying function through a Gaussian noise model $y = f(\boldsymbol{x}) + \epsilon$, where, $f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, $\epsilon \sim N(0, \sigma^2)$. The mean function $m(\boldsymbol{x})$ and covariance kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ are defined as,

$$m(x) = E[f(x)],$$

$$k(x, x') = cov(f(x), f(x')).$$

Thus any collection of function values has a joint Gaussian distribution,

$$[f(x_1), f(x_2), \ldots, f(x_N)]^T \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}),$$

where $\boldsymbol{\mu}_i = m(x_i)$, and K is the $N \times N$ covariance matrix with entries $K_{ij} = k(x_i, x_j)$, which characterizes correlations between different points in the process and must be positive semidefinite.

The smoothness and generalization properties of the GP are encoded by the covariance kernel function $k(x_i, x_j)$ and its hyperparameters $\boldsymbol{\theta}$. In order to learn hyperparameters, we

aim to optimize the marginal likelihood of the data, conditional on kernel hyperparamters $\boldsymbol{\theta}$, and inputs, $\mathbf{x}$,

$$p(y|\theta, \mathbf{x}) = \int p(y|f, \mathbf{x})p(f|\theta)df.$$

Since the responses $y(\mathbf{x})$ are modeled by a GP with additive Gaussian noise, $y(x)|f(x) \sim \mathcal{N}(y(x); f(x), \sigma^2)$, the corresponding log marginal likelihood can be expressed analytically as,

$$\log p(\mathbf{y}|\theta) \propto -log|K + \sigma^2 I| - \mathbf{y}^T(K + \sigma^2 I)^{-1}\mathbf{y}.$$

However, discovering the hidden dynamic from a sequence of high-resolution image data using Gaussian process is still challenging due to the lack of adequate kernel functions and potentially heavy computational cost in model estimation. Thus it is crucial to have an expressive kernel function such that it can approximate any stationary covariance kernel in order to fully discover the hidden dynamic structures of the organs. Moreover, the computational burden increases dramatically with the number of pixels on each dimension as well as the number of images taken along with time. Thus the exact calculation of the log marginal likelihood is impossible. In Chapter 5, I focus on detecting quality changes in organs under preservation by only using biomedical thermal image data. The results are mainly based on a scalable Gaussian process with expressive spectral mixture kernels. The spectral mixture kernels are closed-form kernels introduced by (Wilson and Adams, 2013), which are derived by modeling a spectral density - the Fourier transform of a kernel - with a Gaussian mixture. By applying the spatio-temporal Gaussian process model in a moving window of image sequences, I construct an instructive statistic related to the area under the semi-variogram curve (AUC), to reveal the underlying characteristics of each image automatically.

1.5 Outline of the Thesis

The rest of this dissertation is organized as follows. In Chapter 2, the robustness of GLR (generalized likelihood ratio) control charts is studied for non-normal observations. Several methods are proposed to improve the robustness of the GLR charts with respect to non-normal data. The proposed control charts are also compared with other robust control charts, elaborating the merits of the proposed methods in timely detecting changes across a wide range of shift sizes. Guidance for selecting the design parameters and control limits of the proposed charts is also provided for practitioners. In Chapter 3, we propose a latent process method, so-called mixed switching state-space model (mixed SSSM), to jointly model the mixed-type observations, and effectively detect the change-points. The proposed approach is applied to analyze the civil unrest data in three Latin American countries provided by ICEWS (Integrated Conflict Early Warning System) and GDELT (Global Database of Events, Language, and Tone), showing a superior performance of effectively detecting severe outbreaks of protests.

In Chapter 4, I develop an adaptive network lasso model to conduct segmentation and logistic regression modeling in a simultaneous fashion. This method ensures that the data sharing similar model behaviors is clustered into the same segment automatically. An iteratively weighted least squares (IWLS) algorithm is proposed to achieve a fast convergence rate in parameter estimation by linearizing the nonlinear objective functions. The proposed approach is applied to the IBM pricing data showing superior predictions regarding the purchase likelihood. In Chapter 5, I investigate the change detection problems in the quality of organs based on a sequence of thermal images. A spatio-temporal Gaussian process model with spectral mixture kernels is applied for model fitting and inference. Overlapping moving windows are introduced in the image sequence to maintain local stationarity. A V-statistic is constructed within each window and used directly for change detection.

Finally, discussions and conclusion of this dissertation are included in Chapter 6.

Chapter 2 Robust GLR Control Charts for Monitoring the Process Mean

2.1 Introduction

Many control charts have been developed for monitoring the process mean μ. The traditional Shewhart chart used in process monitoring is only effective if the size of the shift in μ is relatively large. CUSUM and EWMA charts have tuning parameters that allow these charts to be tuned to be particularly effective for detecting shifts of a specified size, but these charts may be much less effective when the actual shift that occurs is not close to the specified size. Thus, none of these charts work well over a wide range of shift sizes.

In practice, the actual size of the shift in μ that will occur is almost never known. Thus, it is desirable to use a control chart that will be effective regardless of the size of the shift that actually occurs. One approach to this problem is to use two or more charts in combination. A second option is adaptive CUSUM and EWMA charts. However, both of these options require multiple control chart parameters, resulting in increased complexity.

Another very good and simpler option is the GLR (generalized likelihood ratio) control chart, where the size of the parameter shift does not need to be specified but can be estimated from the process data. For examples of GLR charts, see Lai (2001), Hawkins et al. (2003), Reynolds Jr and Lou (2010), Peng et al. (2015), Reynolds Jr et al. (2013), and Peng and Reynolds Jr (2014). The overall performance of the standard GLR chart for normal observations is at least as good as other options, such as combinations of Shewhart and CUSUM charts and an adaptive CUSUM chart that has been proposed for detecting a wide range of shift sizes. Please refer to Reynolds Jr and Lou (2012) for more details.

The traditional control charts for monitoring the process mean are based on the sample means, and are constructed assuming that these sample means have a normal distribution. In

many applications, the distribution of the process observations themselves will not be normal, and then the central limit theorem will be used in an attempt to justify the approximate normality of the sample means. In many process monitoring applications, the sample size n is either one or a relatively small value such as 4 or 5. If $n = 1$ then, of course, the central limit theorem does not apply. If n is relatively small then it may not be large enough to produce approximate normality for the sample means if the true distribution of the observations is far away from normal, for example, highly skewed.

When the process observations are not normal, one option is to use a control chart designed for the specific distribution of the observations. However, this distribution may be unknown, or if it is known, there may be no existing chart designed for this particular distribution. Determining an appropriate distribution that fits the process data and then developing a suitable control chart for it may be beyond the time and resources available for many practitioners.

Thus, there is a need for robust charts that will work well for a variety of distributions as well as sample sizes. Some robust charts have been developed. CUSUM and EWMA charts can be made to be robust if they are tuned to detect very small shifts (Montgomery, 2004; Borror et al., 1999; Stoumbos and Sullivan, 2002; Testik et al., 2003; Hawkins and Olwell, 2012). The original observations can also be replaced with winsorized observations when the original observations contain too many extreme values (Hawkins and Olwell, 2012; Reynolds and Stoumbos, 2010).

Another approach to deal with non-normal process variables is to use nonparametric control charts, see the review paper by Chakraborti et al. (2001). However, many nonparametric charts require a relatively large value of n. In addition, nonparametric charts usually are not efficient in the sense that they require a relatively long time to detect shifts in the mean.

The GLR chart has been shown to work very well when the process observations are normal (which is the assumed distribution), but the robustness of the GLR to non-normal observations has not been investigated. The objective of this chapter is to investigate the robustness of the GLR chart for monitoring the process mean and provide several modified GLR charts that will give improved robustness while retaining most of the advantages of the GLR chart. In

particular, I consider the following strategies to improve robustness: (1) increase the sample size used in the GLR chart (and also increase the sampling interval to maintain the same sampling rate per unit time), (2) use a GLR chart with a second restriction on the window over which the GLR statistic is maximized, and (3) transform the process observations used in the GLR chart to reduce the effect of extreme observations. Two methods of transformation are considered, one is based on the χ^2 CDF, and the other is based on a linear transformation of the most extreme observations.

The performance of these proposed robust GLR charts is compared with the performance of a robust CUSUM chart tuned to detect a relatively small shift, and a CUSUM sign chart which is a nonparametric control chart based on the sign statistics. A robust EWMA chart has similar performance compared with the robust CUSUM chart when tuning parameters are adjusted so that they match in an appropriate way. Thus, only one of them is chosen to be compared with our robust GLR in this chapter. A robust GLR control chart based on strategy (3) above with the χ^2 CDF transformation is recommended for use in applications and guidelines are given for choosing the chart design parameters. This robust GLR control chart has good performance in detecting a wide range of shift sizes across a wide range of observational distributions. In addition, it contains few parameters. A prediction equation and a table of control limits are provided so that this GLR chart can be easily set up for use in applications.

2.2 Robustness of the Standard GLR Chart for μ

2.2.1 Sampling From the Process

Consider the situation in which the process variable μ being measured is assumed to have a $N(\mu, \sigma^2)$ distribution, and the distribution is $N(\mu_0, \sigma_0^2)$ when the process is in-control, where μ_0 and σ_0^2 are known or have been estimated accurately during a Phase I period when the process was in control. I consider the problem of real-time monitoring in Phase II. In practice, people never know in advance when the shift in μ will occur or the size of the shift. Thus

the objective is to detect any special cause that produces a shift in μ of any size. Detecting changes in σ^2 using a GLR chart has been evaluated by Reynolds Jr and Lou (2012). The robustness of this GLR chart can be evaluated in a similar manner to what I am going to show in this chapter, and this will be the future research work.

Suppose that independent samples of size $n \geqslant 1$ are taken from the process using a sampling interval of length d between samples. Let $\mathbf{X}_k = (X_{k1}, X_{k2}, \cdots, X_{kn})$ represent the sample obtained at sampling point k. It is assumed that all observations within and between samples are independent. When $n > 1$, we also assume that the n observations in a sample are taken together so that there is a negligible probability of a shift in μ occurring within a sample.

The primary focus of this chapter will be on the situation in which control charts are based on samples of size $n = 1$, because GLR and CUSUM charts have better overall performance when small samples are taken frequently compared to taking larger samples less frequently (Reynolds Jr and Stoumbos, 2004b; Reynolds and Stoumbos, 2010). However, larger values of n are also considered here in the context of investigating robustness.

When $n = 1$, we assume that the time unit is such that the sampling interval d is one time unit. When samples of $n > 1$ are taken from the process we assume $d = n$, so that $d/n = 1.0$ is always obtained. Thus, it is assumed in this chapter that the sampling rate in terms of the number of observations per unit time is fixed and the sampling rate is always one observation per unit time. So using $n > 1$ means that samples can be taken less frequently than when $n = 1$.

2.2.2 The GLR Control Chart

After sample k is obtained, we have the data, $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_k$. Consider the hypothesis that a shift in μ has occurred at some time τ^* between samples τ and $\tau + 1$, where $\tau < k$ and $\mu_1 \neq \mu_0$ (with no change in σ^2 from σ_0^2). Then the observations in samples $\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_\tau$ have a $N(\mu_0, \sigma_0^2)$ distribution, and the observations in samples $\mathbf{X}_{(\tau+1)}, \mathbf{X}_{(\tau+2)}, \cdots, \mathbf{X}_k$ have a $N(\mu_1, \sigma_0^2)$ distribution. Under the null hypothesis that there has been no shift in μ, the

observation in all k samples have a $N(\mu_0, \sigma_0^2)$ distribution, and a log likelihood ratio statistic for testing for a shift in μ is

$$\begin{aligned} R_k &= \log \quad \max_{0\leqslant\tau<k,\mu_0\neq\mu_1} \frac{(2\pi)^{-\frac{n(k-\tau)}{2}}(\sigma_0^2)^{-\frac{n(k-\tau)}{2}} exp\left(-\frac{1}{2\sigma_0^2}\sum_{i=\tau+1}^{k}\sum_{j=1}^{n}(X_{ij}-\mu_1)^2\right)}{(2\pi)^{-\frac{n(k-\tau)}{2}}(\sigma_0^2)^{-\frac{n(k-\tau)}{2}} exp\left(-\frac{1}{2\sigma_0^2}\sum_{i=\tau+1}^{k}\sum_{j=1}^{n}(X_{ij}-\mu_0)^2\right)} \\ &= \max_{0\leqslant\tau<k,\mu_0\neq\mu_1}\left(-\frac{1}{2\sigma_0^2}\sum_{i=\tau+1}^{k}\sum_{j=1}^{n}\left(\mu_1^2-\mu_0^2-2X_{ij}(\mu_1-\mu_0)\right)\right). \end{aligned} \tag{2.1}$$

In evaluating R_k we take the maximum of the log likelihood ratio over the possible values $0, 1, 2, \cdots, k-1$ for the change point τ using μ_1 estimated from observations collected at times $\tau+1, \tau+2, \cdots, k$. The requirement of taking the maximum over $0 \leqslant \tau < k$ means past observations need to be stored, but this should not be a significant problem with modern computing power. When k is large, the computational burden can be eased if necessary by using only the samples in a window of the past m_1 samples (Willsky and Jones, 1976). In order to improve the robustness of the standard GLR chart, I will introduce another restriction on this window later in Section 2.3 to avoid the estimate of τ getting too close to k. Let m_2 represent the size of this restriction, and let $R_(m_1, m_2, k)$ be the value of likelihood ratio when the maximum is taken over $0 \leqslant \tau < k - m_2$ when $k \leqslant m_1$ and over $k - m_1 \leqslant \tau < k - m_2$ when $k > m_1$, where $m_1 > m_2$. In particular, after simplification $R_{m_1,m_2,k}$ becomes

$$R_{m_1,m_2,k} = \begin{cases} \max_{0\leqslant\tau<k-m_2} \quad \frac{n(k-m_2-\tau)}{2\sigma_0^2}\left(\hat{\mu}_{1,\tau,k}-\mu_0\right)^2, & k = 1, 2, \cdots, m_1, \\ \max_{k-m_1\leqslant\tau<k-m_2} \quad \frac{n(k-m_2-\tau)}{2\sigma_0^2}\left(\hat{\mu}_{1,\tau,k}-\mu_0\right)^2, & k = m_1+1, m_1+2, \cdots, \end{cases} \tag{2.2}$$

where $\hat{\mu}_{1,\tau,k}$ is the MLE of μ_1 for a given value of τ, and is given by

$$\hat{\mu}_{1,\tau,k} = \frac{1}{n(k-\tau)}\sum_{i=\tau+1}^{k}\sum_{j=1}^{n} X_{ij}. \tag{2.3}$$

The GLR chart signals at sample k if $R_(m_1, m_2, k) > h$, where the control limit h can be chosen to give a specified in-control performance. The GLR control chart with $n = 1$

and $m_2 = 0$ is referred as the standard GLR chart. Larger values of both n and m_2 will be discussed in Section 2.3 as options for increasing the robustness of the GLR chart.

Lai (1998) has shown that the GLR chart with a window is asymptotically as effective as the GLR chart without a window if the window is large enough. The size of the shift can be expressed in terms of $\delta = |\mu - \mu_0|/\sigma_0$, which is the standardized shift. It appears that $m_1 = 400/n$ is large enough to effectively detect shifts as small as $\delta = 0.25$ (Reynolds Jr and Lou, 2010). Thus, $m_1 = 400/n$ was chosen in this chapter. The choice of m_2 to improve the robustness will be investigated in Section 2.3.2.

2.2.3 The Average Time to Signal

When the process is in control, control charts can be evaluated and compared in terms of the average time to signal (ATS), which is the expected time, measured from the start of monitoring, required by the procedure to signal that a shift in μ has occurred. As long as μ remains at μ_0, the in-control ATS is a measure of the rate of false alarms and should be large so that the frequency of false signals is low. However, if there is a shift in μ from μ_0, which is the out-of-control case, the ATS (calculated from the time of shift) is a measure of the effectiveness of the control chart and should be small so that the change is quickly detected. It is sometimes assumed for simplicity that the shift in μ is present at the very beginning of the monitoring process, and then the expected time to signal can be called the initial-state ATS. But it will usually be more realistic to assume that the shift in μ occurs some time after monitoring starts. If we assume that the shift in μ occurs after the control statistics have had time to reach their steady-state distributions, then the expected time from the shift to the signal is called the steady-state ATS (SSATS). In our simulations, a sequence of observations is discarded if a signal (a false alarm) occurs during the in-control period and the count of observations is reset to zero. This is what Crosier (1986) has called the conditional steady-state ATS.

The SSATS is also based on the assumption that, when the shift occurs at time τ^* in the interval between samples τ and $\tau + 1$, the distribution of τ^* is uniform on $[\tau d, (\tau + 1)d]$. If T

is the time that the control chart signals, then the amount of time required to detect the shift is $T - \tau^*$ and the SSATS is then $E(T - \tau^*)$.

Most of the evaluations of ATS and SSATS in this chapter were done using simulation with 1,000,000 runs, but 100,000 was used for the GLR charts with transformed observations. In the evaluations and comparisons conducted here, the control limits of the charts are chosen so that the in-control ATS is 1481.6 time units. This allows for comparisons with the results for the standard GLR chart given by Reynolds Jr and Lou (2010) and robust CUSUM charts given by Reynolds and Stoumbos (2010), who also used 1481.6. This value was first used in Reynolds Jr and Stoumbos (2004a), where the false alarm of a Shewhart $\overline{X}$ chart with three-sigma control limits occurs on average every 370.4 samples, and with 4 observations obtained every 4 time units, this corresponds to an in-control ATS of 1481.6 time units.

2.2.4 Robustness of the Standard GLR Chart

Now we consider the effect of non-normal distributions on control charts designed under the assumption of a normal distribution. Table 2.1 contains in-control ATS values for the standard GLR chart ($n = 1$) with a window size $m_1 = 400$ for normal and non-normal distributions. In addition to $n = 1$, for completeness I also look at $n = 4$, which is a traditional sample size used in Shewhart charts. Other values of $n > 1$ will be discussed later in subsection 2.3.1.

The column labeled [1] in Table 2.1 gives the in-control ATS of the standard GLR control chart for the normal, Laplace, $t(4)$, $t(10)$, gamma(1), gamma(2), gamma(4), and beta(4,4) distributions, where $t(\nu)$ represents a t distribution with ν degrees of freedom, gamma(α) represents a gamma distribution with shape parameter α, and beta(α, β) represents a beta distribution with parameters α and β. All distributions were scaled to have the same in-control standard deviation. The control limit $h = 7.3294$ was adjusted to give an in-control ATS value of 1481.6 for normal process observations (the actual values may vary slightly from 1481.6 due to simulation error). The gamma(1) distribution is, of course, the exponential distribution, and it is included here only to represent an extreme case of a distribution that looks nothing like the normal distribution. If it appears that the distribution of the process observations is

Table 2.1: In-control ATS values for non-normal observations for the GLR, Shewhart, and CUSUM (tuned to detect $\delta_1 = 1$) control charts, with control limits chosen to give 1481.6 for normal observations.

	$n = d = 1$			$n = d = 4$		
	Standard GLR	Shewhart	CUSUM	GLR	Shewhart	CUSUM
Distribution	[1]	[2]	[3]	[4]	[5]	[6]
Normal	1481.9	1481.7	1481.4	1481.7	1481.7	1481.6
Laplace	185.5	122.5	717.6	792.4	585.1	791.5
t(4)	161.5	116.4	526.1	601.8	449.7	610.9
t(10)	431.5	286.7	1082.9	1133.1	946.7	1130.5
Gamma(1)	108.5	81.5	375.9	508	386.6	511.6
Gamma(2)	159.8	115.9	543.9	691.9	537.3	701.2
Gamma(4)	247.3	175	761.9	900.6	739	917.3
Beta(4,4)	3079.2	$+\infty$	1814.3	1787.1	2269.1	1780.7
h	7.3294	3.3998	6.1465	5.7676	2.9998	1.2578

actually exponential, then it would be appropriate to use a control chart designed specifically for the exponential distribution, see, for example, Gan (1994).

Table 2.1 clearly shows that the standard GLR control chart is not robust to non-normal distributions. For the heavy-tailed Laplace and t distributions, or the skewed gamma distributions, the in-control ATS is far below the nominal value of 1481.6, corresponding to a false alarm rate much higher than expected when the process is operating properly. For example, when the observations have a $t(4)$ distribution and $n = 1$, the ATS will be only 161.5 when $\mu = \mu_0$ and, in the long run, there will be about 9 times as many false alarms as the ATS value of 1481.6 calculated under the normality would indicate. However, for the light-tailed beta distribution, the ATS was much larger than 1481.6.

In order to compare the robustness performance of the standard GLR chart with other charts, I also evaluate the robustness of the Shewhart chart and the CUSUM chart tuned to detect a shift of size $\delta_1 = \frac{|\mu_1 - \mu_0|}{\sigma_0} = 1$. These results are also shown in Table 2.1. $\frac{\delta_1}{2}$ is sometimes called the reference value of the CUSUM chart. Here δ_1 is used as a tuning parameter to tune the CUSUM chart to be particularly sensitive to a shift of this size. I choose $\delta_1 = 1$ here because this value is considered as a standard value for the CUSUM chart. For example, it is the default value in both Minitab and R (Scrucca, 2004). Note that the CUSUM chart usually used for monitoring μ is a two-sided chart that is based on two one-sided statistics.

Table 2.1 shows that the standard GLR chart is more robust than the Shewhart chart but less robust than the CUSUM chart with a "standard" value of δ_1. Thus, even though the GLR chart is not robust, it is still more robust than the widely used Shewhart chart. It has been seen that increasing n from 1 to 4 significantly increases the robustness of the GLR and Shewhart chart, but has relatively little effect on the robustness of the CUSUM chart. The robustness of the CUSUM chart depends on averaging across samples in addition to averaging within samples, with the result that the value of n has little effect. When $n = 4$, the GLR chart has almost the same robustness as the CUSUM chart. Although robustness is higher when $n = 4$, it is still concluded that none of these charts are robust even when $n = 4$.

2.3 Robust GLR Charts

The reason why the standard GLR chart is not robust is that, with $n = 1$ and $m_2 = 0$, only one extreme observation will cause the likelihood ratio statistic to be extremely large and then trigger a signal. Thus, now I will consider various options to make the GLR chart more robust. In addition, I will also consider some other robust or nonparametric charts that would be alternatives to any robust GLR charts that I develop.

2.3.1 GLR Charts with $n > 1$

Even though Table 2.1 indicates non-robustness of the GLR control charts, we can see that the in-control ATS values are improved as the sample size is increased from 1 to 4. Besides, for non-normal observations, the central limit theorem is used to justify the assumption that the distribution of the sample mean is approximately normal. Based on this information, I will investigate how large n needs to be to achieve a reasonable level of robustness.

The ATS for various values of n and d (with $n = d$) are shown in Table 2.2, with columns [2]-[6] corresponding to cases of $n \geqslant 1$. Column [1] represents the standard GLR chart shown in Table 2.1, and is included here for comparison. It is clear that the larger the value of n, the better the performance in terms of robustness. When $n = 10$ (column [4] in Table 2.2), all of the ATS values for non-normal observations are larger than 1,000 and are reasonably close to the nominal in-control ATS value of 1481.6 for normal process observations. (We assume that as long as the actual rate of false alarms is not more than about 1.5 times the nominal rate, this will be acceptable for most practical applications. This standard is much stricter than what is achieved by the standard Shewhart chart currently in widespread use.) It is not surprising to see that the performance is worse when the underlying distribution is far from normal. We also notice that h (given in the last row of Table 2.2) decreases as n increases. This occurs because increasing the sample size decreases the variability of the GLR statistic.

I conclude that using a sample size of about 10 will produce a relatively robust GLR control chart. Thus, this provides a method for obtaining robustness for situations in which

Table 2.2: In-control ATS values for non-normal observations for the GLR control chart for various values of n, with $m_2 = 0$, $d = n$, and control limits chosen to give 1481.6 for normal observations

$n =$	1	4	8	10	12	16
$d =$	1.0	4.0	8.0	10.0	12.0	16.0
Distribution	[1]	[2]	[3]	[4]	[5]	[6]
Normal	1481.9	1481.7	1481.6	1481.6	1481.7	1481.5
Laplace	185.5	792.4	1127.2	1210.5	1268.6	1339.9
t(4)	161.5	601.8	939.1	1040.5	1121.6	1232.5
t(10)	431.5	1133.1	1336.9	1375.1	1404.8	1482.6
Gamma(1)	108.5	508	888.9	1017.1	1118.9	1259.1
Gamma(2)	159.8	691.9	1081.1	1193	1272.6	1364.2
Gamma(4)	247.3	900.6	1240.9	1315.9	1368.3	1421.4
Beta(4,4)	3079.2	1787.1	1584.9	1551.6	1535.6	1512.4
h	7.3294	5.7676	4.9678	4.7081	4.4968	4.1616

it is feasible to take a relatively large sample size at each sampling point (with a relatively long time between samples). Increasing the sample size and sampling interval will, of course, affect the ability to detect shifts in μ, so this issue will be investigated in Section 2.4.

2.3.2 GLR Charts with a second restriction on the window

In the standard GLR control chart ($n = 1, m_2 = 0$) with a window size of $m_1 = 400$, the robustness problem gets server when the change point τ over which we are maximizing in Equations (2.1) and 2.2 is close to the current observation k, since a single observation or a few observations of a non-normal distribution has a higher chance to be extremely large or small, resulting in $R_(m_1, m_2, k) > h$ with a higher possibility. This problem can be eased by introducing a second restriction on the window to avoid τ getting too close to k. If the estimate of the change point is restricted to always be less than $k - m_2$, then this means that there will always be at least $m_2 + 1$ observations used to estimate μ_1. Thus, this effectively increases the sample size used to estimate μ_1 in the GLR statistic.

Table 2.3 contains in-control ATS values for the GLR chart with the second restriction on the window ($m_2 > 0$). Similar to what has been found in Table 2.2, the larger the value of m_2, the better the performance in terms of robustness. When $m_2 = 9$ (column [9] in Table 2.3), all the ATS values for non-normal observations except gamma(1) and $t(4)$, are reasonably close to the nominal in-control ATS value of 1481.6 for normal process observations. Thus I need m_2 to be around 9 to get a reasonable level of robustness, which corresponds to using at least 10 observations to estimate the mean. So $m_2 = 9$ will be chosen for additional comparisons done later to assess the ability to detect shifts in μ.

2.3.3 GLR Charts with χ^2 CDF transformed observations

The non-robustness of the GLR chart is mainly due to the increased likelihood of extreme observations when the true distribution is non-normal. In order to make the chart robust, one idea would be to transform the observations, especially the most extreme observations. If the transformed observations are plotted against the original observations then we would

Table 2.3: In-control ATS values for non-normal observations for the GLR control chart for various values of m_2, with $m_1 = 400$, $n = 1$, $d = 1$, and control limits chosen to give 1481.6 for normal observations

$m_2 =$	1	2	3	4	5	6	7	8	9	10
Distribution	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
Normal	1481.9	1482	1481.9	1481.5	1481.7	1481.4	1481.6	1481.9	1481.5	1481.9
Laplace	185.5	505.7	675	814.8	924.5	1012.9	1085.9	1143.5	1192.1	1233.6
t(4)	161.5	360.7	468.1	568.8	659.9	742	816.3	881.4	939.1	992.3
t(10)	431.5	949.7	1091	1178.3	1242.7	1286	1321.5	1348.3	1367.9	1387.4
Gamma(1)	108.5	243.5	327.5	413.5	499.7	581.9	660.7	734.3	801.3	862.3
Gamma(2)	159.8	370.3	496.1	613.3	721.1	818.5	901	972	1032.6	1086.2
Gamma(4)	247.3	580	737.2	865.6	968.8	1051	1116.7	1170.8	1216	1252.8
Beta(4,4)	3079.2	1880.4	1762.5	1684.3	1637.2	1602.2	1577.7	1559.5	1545.1	1533.5
h	7.3294	6.1674	5.9333	5.7468	5.593	5.46	5.3432	5.2383	5.143	5.0562

want the plot to be close to a 45 degree line when the observations are close to the center of the distribution. But, as the observations become more extreme, we want to transform observations to increasingly deviate from this line as they are pulled back to be less extreme.

Cumulative distribution functions (CDF) are convenient and appropriate mathematical functions for developing a transformation of this type. CDFs for various distributions were examined and the CDF of the χ^2 distribution was selected in this chapter for two reasons. First, there is only one parameter (df, the degrees of freedom) in a χ^2 distribution, so it is easy to construct the transformation. Second, while we do not want to make the non-extreme observations larger than their original values using any transformation, we do not want them to be too much smaller than the original neither, since the true data will be underestimated. The CDF of the χ^2 distribution does a good job by transforming the data in the center of the distribution to be close to their original values. The CDF curve of a χ^2 distribution increases relatively slowly before it reaches 1. Thus, by choosing an appropriate value of df, I can simply modify the CDF function such that the obtained curve is close to the 45 degree straight line when the observations are not extreme(the 45 degree straight line represents the

situation where there is no transformation). An illustration of this transformation is shown in Figure 2.1, and will be explained later.

The transformation formula for any observation X_{kj} is

$$X_{kj}^c = \begin{cases} \mu_0 + \sigma_0 bF\left(\frac{X_{kj}-\mu_0}{\sigma_0}\right), & X_{kj} \geqslant \mu_0 \\ \mu_0 - \sigma_0 bF\left(-\frac{X_{kj}-\mu_0}{\sigma_0}\right), & X_{kj} < \mu_0 \end{cases} \quad j = 1, 2, \cdots, n, \tag{2.4}$$

where F is the cumulative distribution function of a χ^2 distribution with a degree of freedom df. Since F ranges from 0 to 1, b is used as a tuning parameter to make X_{kj}^c reasonably close to the original observation X_{kj} when the original observation is not too extreme. Then the transformed observations X_{kj}^c will be used to calculate the MLE of μ_1 in Equation (2.3). The objective of developing the transformation is to have a robust GLR chart for use in the case of $n = 1$, so I consider the transformation in this case.

Figure 2.1 shows how the observed data is transformed for the case of $\mu_0 = 0, \sigma_0 = 1$, and $df = 2$, where the seven curves correspond to $b = 2.0, 2.25, 2.75, 3.25, 3.75, 4.25, 4.75$, respectively. The x-axis is the original observed data X_{kj}, while the y-axis stands for the transformed observation X_{kj}^c. The black line is the 45 degree straight line. It is clear that when using this χ^2 CDF transformation, all the observations are transformed to be within the interval $[\mu_0 - b\sigma_0, \mu_0 + b\sigma_0]$.

Various combinations of df and b were evaluated and some of those that give a reasonable in-control performance are listed in Table 2.4. We can see that different combinations of df and b give different in-control ATS values for non-normal distributions. In general, the performance when $df = 2$ and $b = 4.25$ (column [2]) is better than the other combinations, since the ATS values of all distributions except gamma(1) are larger than $1481.6/1.5 = 987.7$. However, the in-control ATS values of the Laplace and $t(4)$ distributions are significantly larger than expected. Although a reasonably large in-control ATS indicates a low rate of false alarms, we do not want an in-control ATS that is too large because it means that detecting changes in μ

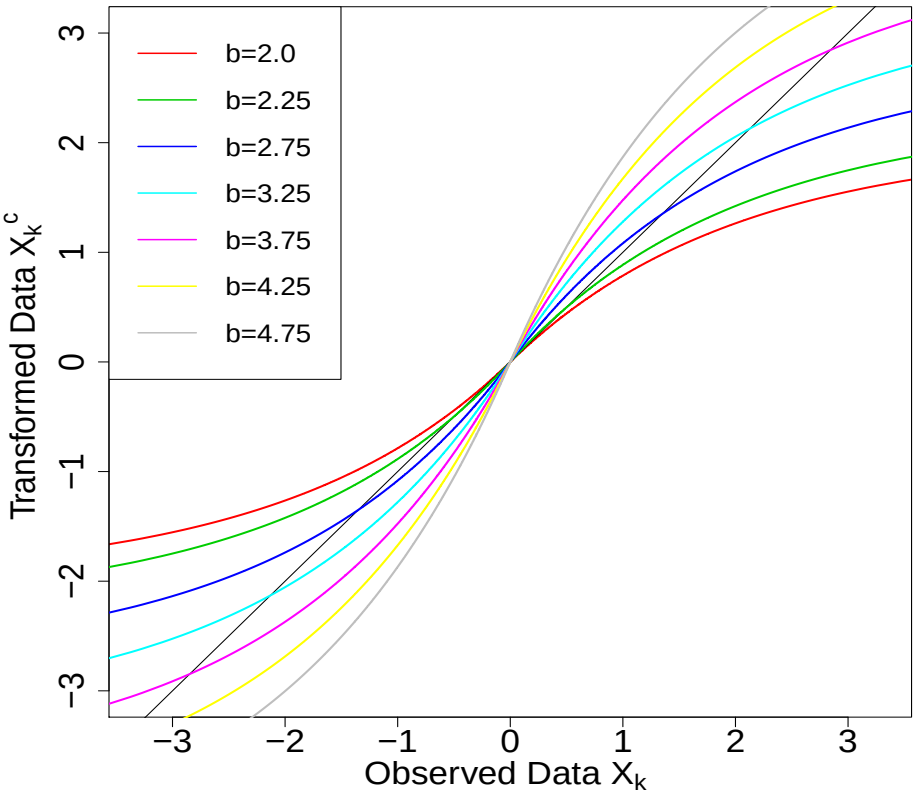


Figure 2.1: The χ^2 transformation of the observed data when $df = 2$.

will take more time than necessary. I will discuss this issue later in Section 2.4. Thus the chart with the combination of $df = 2$ and $b = 4.25$ in column [2] will be recommended and used in future comparisons of different methods. In addition, the chart with $df = 2$ and $b = 2.25$ in column [1] will also be included in our final comparison due to its excellent out-of-control performance.

Both of the recommended charts have the same degrees of freedom $df = 2$, which corresponds to an exponential distribution with mean two. Thus, Equation (2.4) can be simplified using the exact exponential CDF shown below,

$$X_{kj}^c = \begin{cases} \mu_0 + \sigma_0 b\left(1 - \exp\left(-\frac{X_{kj}-\mu_0}{2\sigma_0}\right)\right), & X_{kj} \geqslant \mu_0 \\ \mu_0 - \sigma_0 b\left(1 - \exp\left(\frac{X_{kj}-\mu_0}{2\sigma_0}\right)\right), & X_{kj} < \mu_0 \end{cases} \quad j = 1, 2, \cdots, n. \tag{2.5}$$

Table 2.4: In-control ATS values for non-normal observations for χ^2 CDF transformed control chart for various combinations of b and df, with $m_2 = 0$, $n = 1$, $d = 1$, and control limits chosen to give 1481.6 for normal observations

	$df = 2$		$df = 2.25$		$df = 2.5$	
$b =$	2.25	4.25	2.75	3	3.25	3.5
Distribution	[1]	[2]	[3]	[4]	[6]	[7]
Normal	1481.3	1481.6	1481.4	1481.6	1481.2	1481.7
Laplace	3982.3	2893.2	2725.3	2583.1	1760.9	1688.7
t(4)	2729.3	2705.4	1766.7	1767.6	1001.3	998.2
t(10)	1696.2	1699.8	1509.7	1508.8	1260.2	1259
Gamma(1)	795.2	849.5	679.5	689	483.7	485
Gamma(2)	914.6	993.8	776.5	785.5	577.4	580.2
Gamma(4)	1048.9	1126.3	902.9	911.2	705.1	709
Beta(4,4)	1320	1330.1	1443.3	1444	1643.5	1643.5
h	4.3023	14.7011	5.036	5.95	5.5504	6.3959

2.3.4 GLR Charts with Linear Transformed Observations

In this section the data will be transformed differently according to the distance between τ and k as well as the value of X_{kj}. More reduction of extreme values will be applied to the original observations when $k-\tau$ gets smaller. The transformation formula for any observation X_{kj} is

$$X_{kj}^{l} = \begin{cases} \frac{k-\tau-1}{m_1-1}\left(X_{kj} - (\mu_0 + c\sigma_0)\right) + (\mu_0 + c\sigma_0), & X_{kj} \geqslant \mu_0 + c\sigma_0 \\ X_{kj}, \quad |X_{kj} - \mu_0| \leqslant c\sigma_0 & j = 1, 2, \cdots, n, \\ \frac{k-\tau-1}{m_1-1}\left(X_{kj} - (\mu_0 - c\sigma_0)\right) + (\mu_0 - c\sigma_0), & X_{kj} \leqslant \mu_0 - c\sigma_0 \end{cases} \quad (2.6)$$

where $c > 0$ is a specified constant (which becomes a chart parameter). The idea is that if an original observation is more extreme than c standard deviation from target, it is replaced with a transformed observation where the multiplier is a simple linear function of $k-\tau$. Then the transformed observations will be used to calculate the MLE of μ_1 in Equation (2.3).

For the case of $n = 1$, Figure 2.2 shows how the observed data is transformed for the case of $\mu_0 = 0$, $\sigma_0 = 1$, and $c = 2.5$. The x-axis is for the original observations, the y-axis is for the transformed observations, and the black line is the 45 degree straight line. Several possible values of the slope $\frac{k-\tau-1}{m_1-1}$ chosen are plotted in the figure to show how the transformation is performed.

Different values of c were evaluated in Table 2.5. A small value of c gives relatively large in-control ATS values for all non-normal distributions. However, it is undesirable to have in-control ATS values that are too large. Thus, even though $c = 2.25$ produces the largest in-control ATS values, it may take much more time to detect any changes in μ. From this point of view, $c = 2.50$ might be a better choice. The robustness in terms of the in-control ATS when $c = 2.50$ is the best compared with all other charts I have discussed in Section 2.3 so far, so the chart with $c = 2.50$ will be chosen for use in future comparisons.

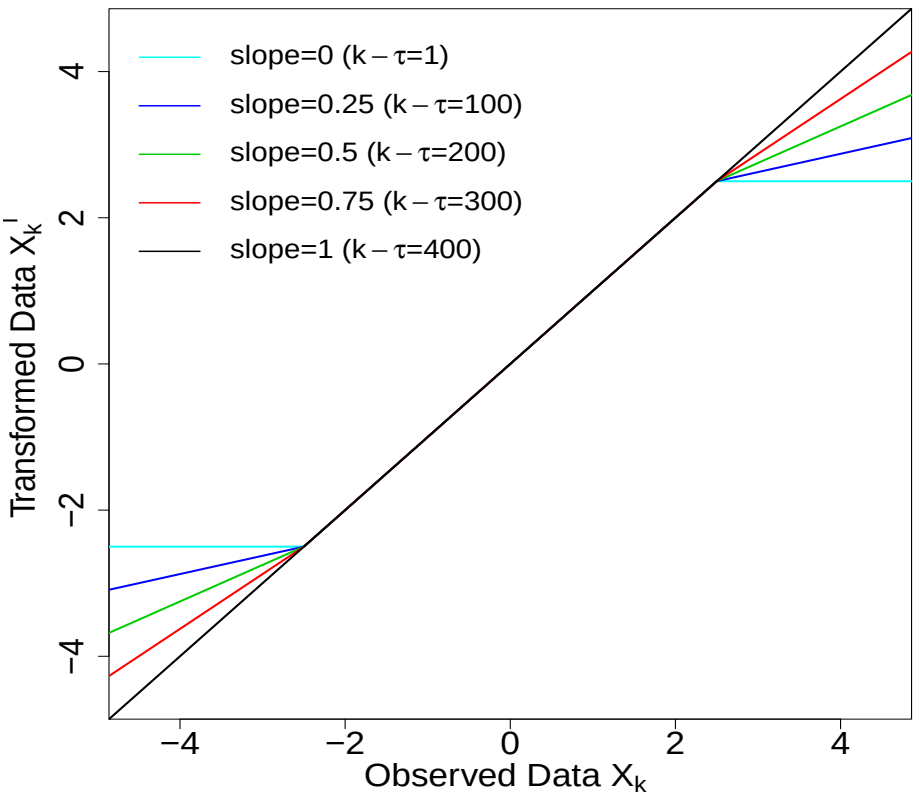


Figure 2.2: Linear transformation of the observed data when $c = 2.5$.

2.3.5 Robust CUSUM Control Chart Tuned to Detect $\delta_1 = |\mu_1 - \mu_0|/\sigma_0$

The CUSUM chart parameter $\delta_1 = |\mu_1 - \mu_0|/\sigma_0$ is the tuning parameter introduced in subection 2.2.3 as the standardized size of the shift that the CUSUM chart is designed to detect. Since the CUSUM chart can be made more robust by reducing δ_1, see Hawkins and Olwell (2012), a robust CUSUM chart with a small value of δ_1 would be an alternative to the robust GLR charts that I propose. The CUSUM results are shown here for completeness and to see how small δ_1 needs to be to achieve reasonable robustness.

Table 2.6 contains in-control ATS values for the CUSUM control chart for several values of δ_1. It is clear to see that a CUSUM control chart tuned to detect a smaller value of δ_1 has better in-control performance. Column [1] corresponds to $\delta_1 = 0.3$ and the in-control ATS values seem to be very close to the desired value of 1481.6. Thus the CUSUM charts tuned to detect $\delta_1 = 0.3$ is chosen for use in future comparisons. In addition, a larger value of δ_1 (such

Table 2.5: In-control ATS values for non-normal observations for linear transformed control chart for various values of c, with $m_2 = 0$, $n = 1$, $d = 1$, and control limits chosen to give 1481.6 for normal observations

	2.25	2.5	2.75	3
Distribution	[1]	[2]	[3]	[4]
Normal	1482	1481.4	1481.3	1482.3
Laplace	2305.1	1766.5	1054.3	864.6
t(4)	3330.8	2636.4	1546.4	1261.6
t(10)	1740.9	1582.1	1314.8	1187.1
Gamma(1)	1914.9	1406.3	679.2	551.8
Gamma(2)	1689.2	1319	754.8	626.9
Gamma(4)	1654.6	1356.4	911.1	780
Beta(4,4)	1386.9	1607.6	1941.7	2189.9
h	6.4064	6.6399	6.8596	6.9819

as, $\delta_1 = 0.6$) seems to give a level of robustness comparable to those transformation charts recommended in subsections 2.3.3 and 2.3.4. To make a fair comparison, I will also include this CUSUM chart tuned to detect $\delta_1 = 0.6$ in the final comparisons.

2.3.6 The CUSUM Sign Chart

In investigating robust control charts, nonparametric charts should be considered as an alternative. In this subsection, I will consider the CUSUM sign chart which is one of the relatively few nonparametric charts that can be used when $n = 1$. The CUSUM sign chart for monitoring the process median (M) and mean (μ) is based on signs computed for each observation. The in-control ATS will be the same for all distributions for which the median equals the target value. If the distribution is symmetric, then, of course, the mean and median are the same if the mean exists. For the asymmetric gamma distribution, the sign test is actually a test for a change in the median of the observations. Please refer to Amin et al. (1995) for more details on nonparametric control charts based on the sign statistic.

Table 2.6: In-control ATS values for non-normal observations for the CUSUM control chart for various values of δ_1, with $n = 1$, $d = 1$, and control limits chosen to give 1481.6 for normal observations

$\delta_1 =$	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Distribution	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]
Normal	1481.2	1482	1481.6	1481.5	1481.6	1481.5	1481.6	1481.4
Laplace	1470.3	1392.6	1288.1	1171.1	1050	931.9	821.1	717.6
t(4)	1397.7	1254	1096.8	946.8	813.8	699.4	604.9	526.1
t(10)	1476.8	1450.7	1409.6	1357.4	1295.9	1229.3	1157.6	1082.9
Gamma(1)	1350.2	1167.5	970.2	793	647.8	533.9	445.1	375.9
Gamma(2)	1416.5	1304.4	1160.8	1007.9	865.1	738.3	631	543.9
Gamma(4)	1446.2	1383.2	1293.3	1187.3	1074.1	962.1	856.5	761.9
Beta(4,4)	1482.3	1500.2	1521.9	1555.3	1599.3	1656	1728.7	1814.3
h	15.2904	12.5742	10.7002	9.3228	8.2613	7.415	6.7241	6.1465

Let M_0 denote the median when the process is in control. Then all observations have equal probabilities to be larger or smaller than M_0 when the process is in control. Thus, when $n = 1$, the CUSUM sign chart can be considered as a Bernoulli CUSUM control chart for monitoring the proportion $p = P(X_k \geqslant M_0)$ when there is a continuous stream of independent binary observations, see Reynolds Jr and Stoumbos (1999). Let us first code the observation X_k into a Bernoulli observation B_k using the formula,

$$B_k^c = \text{sign}\,(X_k - M_0) = \begin{cases} 1, & X_k \geqslant M_0 \\ & \qquad k = 1, 2, \cdots . \\ 0, & X_k < M_0 \end{cases} \tag{2.7}$$

Then, the log-likelihood-ratio-based Bernoulli CUSUM control statistic at observation k is given by,

$$Y_k^U = \max\left\{0, Y_{k-1}^U\right\} + \left\{B_k - \gamma_B\right\}, k = 1, 2, \cdots , \tag{2.8}$$

where $\gamma_B = -ln\left(\frac{1-p_1}{1-p_0}\right)/ln\left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right)$ and $p_1 > p_0$ is a value of p that should be detected quickly. In this chapter, $p_0 = P(X_k \geqslant M_0) = P(X_k < M_0) = 0.5$ when the process is in control, while $p_1 = P(X_k \geqslant M_0)$ when the process is out of control with median M_1. p_1 is thus a function of M_1 and it is different for different distributions. This chart signals if $Y_k^U \geqslant h$ for some upper limit h.

This is a one-sided CUSUM sign chart to detect increases in M. In order to detect both increases and decreases in M, a two-sided chart should be used. By a simple modification, we can get the other side of the chart for detecting the decreases in M. Denote the statistic at observation k by Y_k^L. Then the two CUSUM sign charts are combined as a two-sided CUSUM sign chart, and it will signal if either $Y_k^U \geqslant h$ or $Y_k^L \leqslant -h$.

Since the densities of the normal, Laplace, $t(4)$, $t(10)$ and beta(4,4) distributions are symmetric, their means and medians are the same. Thus the CUSUM sign charts for monitoring the distribution mean and median are identical. However, they are different for the gamma distributions, which are asymmetric. Monitoring the mean of these asymmetric distributions using the non-parametric CUSUM sign chart requires knowledge of the in-control probability that an observation is above the mean. This probability depends on the unknown distribution. Thus, the CUSUM sign chart considered here is assumed to be monitoring the distribution median.

Table 2.7 gives in-control ATS values of the CUSUM sign chart for various values of δ_1, where each δ_1 corresponds to a specific p_1. Since all the in-control ATS values are the same for a given value of p_1, despite of the underlying distribution, only the common values are shown in the table. For different values of p_1, the corresponding control limit h is chosen to achieve approximately the desired in-control ATS value of 1481.6. However, because of the discreteness of the Bernoulli observation B_k and the uniqueness of γ_B (determined when p_1 is specified), it is unlikely to achieve exactly this ATS value. All the h values given in Table 2.7 correspond to an in-control ATS that is as close to the nominal value as possible. The out-of-control performance of the CUSUM sign chart with various values of δ_1 is also evaluated. However, none of them performs well. Thus I will include the chart when $\delta_1 = 0.4$ in the final

comparison for completeness.

Table 2.7: In-control ATS values for non-normal observations for the CUSUM sign control chart for various values of δ_1, with $n = 1$, $d = 1$, and control limits chosen to give 1481.6 for normal observations

$\delta_1 =$	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
$p_1 =$	0.5398	0.5793	0.6179	0.6554	0.6915	0.7257	0.758	0.7881	0.8159	0.8413
	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]
In-control ATS	1476.6	1481.9	1480.6	1472.3	1481.2	1509.8	1478.1	1488.6	1418.3	1528.7
h	15.5975	11.284	8.9307	7.4169	6.3494	5.5928	4.9384	4.4846	3.9358	3.6872

2.4 The Out-of-control Performance

In addition to trying to find a chart whose in-control performance is robust, we also need a chart that will detect shifts quickly. Thus, I need to investigate the out-of-control performance of the various robust charts that I am considering. As shown in Tables 2.1 to 2.6, when the process observations are not normally distributed, the in-control ATS value may not be extremely close to the nominal value of 1481.6. Thus, in order to do fair comparisons of the ability of different charts to detect parameter shifts in non-normal distributions, the control limits of the charts being compared need to be adjusted so that they have the same in-control ATS values (1481.6) for the distribution being considered (Reynolds and Stoumbos, 2010).

2.4.1 Standard GLR Control Chart

Table 2.8 gives SSATS values for the standard GLR chart with adjusted control limits for various shifts. When δ is very large, almost all the SSATS values for each distribution achieve the minimum value 0.5. However, when the δ is small, especially smaller than 1.00, the SSATS values for the $t(4)$ and gamma(1) distributions are much higher than the others. Thus, a longer time is needed to detect the shift. So the conclusion we can get is that, for a fixed false alarm

rate, it takes longer to detect small and moderate shifts in heavy-tailed distributions than it does in a normal distribution. While for very large shifts there is not much difference in the time required for the different distributions.

Table 2.8: SSATS values for shifts in μ for normal and non-normal observations for the standard GLR control chart with $n = 1$, $m_2 = 0$, and $d = 1$

δ	Normal	Laplace	t(4)	t(10)	Gamma(1)	Gamma(2)	Gamma(4)	Beta(4,4)
0.00	1481.9	1480.7	1480.6	1482.3	1481.5	1481.7	1481.4	1482
0.25	151.7	369.4	975.4	234.1	711.4	423.3	297.1	129.9
0.50	43.8	93.8	162.1	64.6	142.4	104.2	80.3	38
0.75	21	43.6	75.4	30.4	66.8	48.8	37.8	18.4
1.00	12.5	25.2	43.4	17.8	38.8	28.4	22.1	11.0
1.50	6.0	11.6	19.8	8.3	17.7	13.2	10.3	5.3
2.00	3.6	6.7	11.3	4.9	10.1	7.6	6.0	3.2
3.00	1.7	3.1	5.1	2.3	4.6	3.4	2.8	1.5
4.00	1	1.7	2.9	1.3	2.6	2.0	1.6	0.9
5.00	0.6	1.2	1.8	0.8	1.7	1.3	1.0	0.6
6.00	0.5	0.7	1.4	0.6	1.2	0.9	0.6	0.5
7.00	0.5	0.5	0.9	0.5	0.8	0.5	0.5	0.5
8.00	0.5	0.5	0.6	0.5	0.5	0.5	0.5	0.5
h	7.3294	13.7136	22.9115	9.9786	20.4356	15.3076	12.1658	6.5864

Now let us compare the out-of-control performance of the proposed robust charts considered in Section 2.3. In order to do a better comparison, the SSATS results are arranged according to the distributions. SSATS values for the normal distribution are shown in Table 2.9. Results for the Laplace, $t(4)$, and gamma(2) distributions are shown in Table 2.10 to Table 2.12, respectively, which represent two heavy-tailed distributions and one skewed distribution. SSATS values were also obtained for other distributions, but, for brevity, are not shown here.

2.4.2 Out of Control Performance Comparison

In Table 2.9, normal observations are considered, and the SSATS values for the standard GLR control chart are listed in column [1], while columns [2]-[6] correspond to the four robust GLR charts developed in subsections 2.3.1 to 2.3.4. In addition, the results for two robust CUSUM charts tuned to detect a small shift ($\delta_1 = 0.3$ and $\delta_1 = 0.6$), and a CUSUM sign chart are shown in columns [7]-[9], respectively, for purposes of comparison. The robust CUSUM chart tuned to detect $\delta_1 = 0.6$ was included because it gives in-control performance comparable to the other robust charts I consider in columns [2]-[6].

We can see that the standard GLR chart in column [1] has a relatively large SSATS when the shift size is small, while all of the robust GLR charts in columns [2]-[6] have relatively better performance. Many robust statistical procedures designed to work well for non-normal data do not work very well for normal data, which is a significant trade-off when selecting robust control chart. However, in the current situation the robust charts actually work better than the standard GLR chart under normality for small or medium shifts ($\delta < 3$). The χ^2 CDF chart with $df = 2$ and $b = 4.25$ in column [4] gives the lowest SSATS. This indicates that it would be reasonable for a practitioner to use the robust chart even with normal observations as long as the primary concern is detecting small or medium shifts. However, the standard GLR chart gives good performance for large shift sizes with a minimum SSATS of 0.5.

The out-of-control performance of the CUSUM sign chart (column [9]) is the worst over all shift sizes. The CUSUM chart with $\delta_1 = 0.3$ (column [7]) is bad for medium and large shifts, while it performs relatively well for small shifts.

Among the robust GLR charts, the GLR chart in column [2] with n increased to 10 works well for small shifts but has very bad performance for large shifts because the sampling interval d is also increased to 10 and the minimum SSATS is 5. The GLR chart with $m_2 = 9$ in column [3] has slightly worse performance than the GLR chart in column [2] when shift sizes are small, while better performance for large shifts. By using $m_2 = 9$ I constrain the estimation of the change point τ to be at least 10 observations before the current observation.

The χ^2 CDF transformation charts in columns [4] and [5] are overall the best for different sizes of shifts. They both outperform the CUSUM control chart tuned to detect $\delta_1 = 0.6$. In particular, the χ^2 CDF chart with $df = 2$ and $b = 2.25$ in column [4] has the smallest SSATS among all the other charts (except the standard GLR chart) including the robust CUSUM control chart tuned to detect $\delta_1 = 0.3$ (column [7]). Besides, compared with other types of robust charts (columns [2]-[3] and [6]-[9]), the χ^2 CDF chart with $df = 2$ and $b = 4.25$ has the best performance when the shift size is larger than 0.25. When the shift size is 0.25, its SSATS is still the smallest except column [4] and column [7]. The linear transformation chart has good performance for medium shifts, but relatively worse performance when the shift is small or large.

Similar results can be found in Table 2.10 to Table 2.12, where SSATS values are given for Laplace, $t(4)$, gamma(2) observations, respectively. What's more, for small and moderate shifts the performance of the robust GLR charts relative to the performance of the standard GLR is better for heavy-tailed and skewed distributions compared to the relative performance for the normal distribution. This means that people would gain more than just robustness of the in-control ATS when using the robust charts instead of the standard GLR and the actual distribution has heavy tails and/or is skewed.

Besides, except for the CUSUM sign chart, the CUSUM chart with $\delta_1 = 0.3$ is the most robust chart among those charts considered here. However, its out-of-control performance is not good, so it would be reasonable to use this chart only if in-control performance is the primary concern.

All in all, the χ^2 CDF transformation chart has very good out-of-control performance over a wide range of shifts. Considering the in-control performance of these two proposed χ^2 CDF transformation charts, the χ^2 CDF chart with $df = 2$ and $b = 4.25$ is chosen as our best robust GLR control chart. Tables of control limits are provided in Section 2.5, so that this GLR chart can be easily set up for use in applications.

Table 2.9: SSATS values for shifts in μ for normal observations under different control charts

	GLR		GLR with $m_2 > 0$	χ^2 CDF transformed observation		Linear transformed observation	CUSUM control chart		CUSUM sign
$n =$	1	10	1	1	1	1	1	1	1
$m_1 =$	400	400	400	400	400	400	-	-	-
$m_2 =$	-	-	9	-	-	-	-	-	-
	-	-	-	$df = 2$ $b = 2.25$	$b = 2.25$ $b = 4.25$	$c = 2.5$	$\delta_1 = 0.3$	$\delta_1 = 0.6$	$\delta_1 = 0.4$
δ	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0.00	1481.9	1481.6	1481.5	1481.3	1481.6	1481.4	1481.2	1481.5	1472.3
0.25	151.7	111.2	118.7	103.8	120.7	136.9	104.7	154.8	159.2
0.50	43.8	34.4	38.4	32.1	36.7	40.4	36.8	36.8	52.5
0.75	21.0	16.9	20.7	16.1	18.2	19.7	21.8	18.5	31.5
1.00	12.5	10.0	13.7	9.9	11.3	12.0	15.4	12.1	23.5
1.50	6.0	5.4	7.9	5.1	5.9	6.0	9.7	7.1	17.3
2.00	3.6	5.0	5.5	3.3	3.9	3.9	7.0	5.0	15.4
3.00	1.7	5.0	3.3	1.9	2.3	2.6	4.6	3.1	14.6
4.00	1.0	5.0	2.4	1.4	1.7	2.4	3.4	2.3	14.6
5.00	0.6	5.0	1.9	1.2	1.4	2.3	2.7	1.7	14.6
6.00	0.5	5.0	1.5	1.1	1.3	2.3	2.2	1.4	14.6
7.00	0.5	5.0	1.3	1.1	1.3	2.3	1.9	1.3	14.6
8.00	0.5	5.0	1.2	1.0	1.3	2.3	1.6	1.2	14.6
h	7.3294	4.7081	5.143	4.3023	14.7011	6.6399	15.2904	9.3228	7.4169

Table 2.10: SSATS values for shifts in μ for Laplace observations under different control charts

	GLR		GLR with $m_2 > 0$	χ^2 CDF transformed observation		Linear transformed observation	CUSUM control chart		CUSUM sign
$n =$	1	10	1	1	1	1	1	1	1
$m_1 =$	400	400	400	400	400	400	-	-	-
$m_2 =$	-	-	9	-	-	-	-	-	-
	-	-	-	$df = 2$ $b = 2.25$	$b = 2.25$ $b = 4.25$	$c = 2.5$	$\delta_1 = 0.3$	$\delta_1 = 0.6$	$\delta_1 = 0.4$
δ	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0.00	1480.7	1482.5	1481.6	1481.7	1481.7	1481.3	1481.5	1481.6	1472.3
0.25	369.4	118.8	126.3	117.2	117.2	147.9	106.8	179.3	78.2
0.50	93.8	36.5	40.5	34.8	34.8	42.8	37.2	39.6	35.0
0.75	43.6	17.8	21.6	17.4	17.4	20.4	22.0	19.6	24.8
1.00	25.2	10.5	14.2	10.9	10.9	12.2	15.5	12.8	20.6
1.50	11.6	5.5	8.1	5.9	5.9	5.9	9.8	7.5	17.0
2.00	6.7	5.0	5.6	4.0	4.0	3.7	7.1	5.3	15.6
3.00	3.1	5.0	3.4	2.6	2.6	2.5	4.6	3.3	14.8
4.00	1.7	5.0	2.4	1.9	1.9	2.4	3.4	2.4	14.6
5.00	1.2	5.0	1.9	1.6	1.6	2.4	2.7	1.9	14.6
6.00	0.7	5.0	1.5	1.5	1.5	2.4	2.2	1.5	14.6
7.00	0.5	5.0	1.3	1.5	1.5	2.4	1.9	1.4	14.6
8.00	0.5	5.0	1.2	1.5	1.5	2.4	1.6	1.3	14.6
h	13.7136	4.9662	5.3334	3.7409	13.3473	6.4573	15.3153	9.7434	7.4169

Table 2.11: SSATS values for shifts in μ for $t(4)$ observations under different control charts

	GLR		GLR with $m_2 > 0$	χ^2 CDF transformed observation		Linear transformed observation	CUSUM control chart		CUSUM sign
$n =$	1	10	1	1	1	1	1	1	1
$m_1 =$	400	400	400	400	400	400	-	-	-
$m_2 =$	-	-	9	-	-	-	-	-	-
	-	-	-	$df = 2$ $b = 2.25$	$b = 2.25$ $b = 4.25$	$c = 2.5$	$\delta_1 = 0.3$	$\delta_1 = 0.6$	$\delta_1 = 0.4$
δ	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0.00	1480.6	1481.4	1481.7	1481.6	1481.3	1359	1482.4	1481.5	1472.3
0.25	975.4	130.0	136.6	114.3	114.3	142.3	109.7	208.3	99.2
0.50	162.1	39.2	43.3	33.8	33.8	41.2	37.9	42.6	37.6
0.75	75.4	19.0	23.0	16.9	16.9	19.7	22.4	20.9	25.0
1.00	43.4	11.1	15.0	10.5	10.5	11.7	15.8	13.6	20.1
1.50	19.8	5.6	8.6	5.7	5.7	5.7	9.9	8.0	16.6
2.00	11.3	5.0	5.9	3.9	3.9	3.5	7.2	5.6	15.4
3.00	5.1	5.0	3.6	2.5	2.5	2.0	4.7	3.5	14.8
4.00	2.9	5.0	2.6	1.8	1.8	1.6	3.4	2.5	14.6
5.00	1.8	5.0	2.0	1.5	1.5	1.5	2.7	2.0	14.6
6.00	1.4	5.0	1.6	1.5	1.5	1.5	2.3	1.6	14.6
7.00	0.9	5.0	1.4	1.5	1.5	1.5	2.0	1.4	14.6
8.00	0.6	5.0	1.3	1.5	1.5	1.5	1.6	1.4	14.6
h	22.9115	5.2711	5.6746	3.6333	12.9628	6.2498	15.4958	10.3006	7.4169

Table 2.12: SSATS values for shifts in μ for gamma(2) observations under different control charts

	GLR		GLR with $m_2 > 0$	χ^2 CDF transformed observation		Linear transformed observation	CUSUM control chart		CUSUM sign
$n =$	1	10	1	1	1	1	1	1	1
$m_1 =$	400	400	400	400	400	400	-	-	-
$m_2 =$	-	-	9	-	-	-	-	-	-
	-	-	-	$df = 2$ $b = 2.25$	$b = 2.25$ $b = 4.25$	$c = 2.5$	$\delta_1 = 0.3$	$\delta_1 = 0.6$	$\delta_1 = 0.4$
δ	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
0.00	1481.7	1481.6	1481.7	1481.4	1481.7	1481.9	1481.6	1481.7	1472.3
0.25	423.3	120.9	133.3	257.2	257.2	177.5	110.5	178.7	116.6
0.50	104.2	37.5	42.9	54.3	54.3	50.1	38.2	42.6	36.5
0.75	48.8	18.3	22.9	23.9	23.9	23.7	22.5	20.8	21.0
1.00	28.4	10.8	15.0	13.9	13.9	14.0	15.8	13.5	15.6
1.50	13.2	5.4	8.5	6.9	6.9	6.8	9.9	7.8	14.6
2.00	7.6	5.0	5.8	4.5	4.5	4.1	7.1	5.4	14.6
3.00	3.4	5.0	3.5	2.8	2.8	2.4	4.6	3.4	14.6
4.00	2.0	5.0	2.5	2.3	2.3	2.4	3.4	2.4	14.6
5.00	1.3	5.0	2.0	2.0	2.0	2.4	2.7	1.9	14.6
6.00	0.9	5.0	1.5	1.5	1.5	2.4	2.2	1.5	14.6
7.00	0.5	5.0	1.3	1.5	1.5	2.4	1.9	1.4	14.6
8.00	0.5	5.0	1.3	1.5	1.5	2.4	1.6	1.3	14.6
h	15.3076	5.0173	5.5093	4.2155	15.0409	6.7788	15.4418	10.102	7.4169

2.5 Choosing the Control Limit of a Robust GLR Control Chart

In this section, I will show how to find values of h for the χ^2 CDF chart with $df = 2$ and $b = 4.25$. For the case in which $n = 1$, $m_1 = 400$, and $m_2 = 0$, the value of h is the only parameter that needs to be determined in order to use this robust control chart.

Simulations are used to find the in-control ATS for a number of values of h that correspond to in-control ATS values between approximately 50 and 10,000. A plot of h and the natural log in-control ATS is given in Figure 2.3, showing an approximately linear relationship. In order to get a more accurate model between the control limit and the in-control ATS value, a 3th-order polynomial equation relating h to the natural log in-control ATS was fitted and is shown in Equation (2.9), where the fitted equation has both the R^2 and adjusted R^2 equal to about one.

The polynomial linear function for the χ^2 CDF chart with $df = 2$ and $b = 4.25$ is,

$$h = 6.154175 - 2.175331\log(\text{ATS}) + 0.767293\left(\log(\text{ATS})\right)^2 - 0.042325\left(\log(\text{ATS})\right)^3. \quad (2.9)$$

Table 2.13 gives fitted values of h for the χ^2 CDF chart, corresponding to some in-control ATS values ranging from 50 to 10,000 constructed based on this 3rd-order polynomial relationship. If the in-control ATS value required for an application is not in Table 2.13, practitioners can easily find the corresponding control limit h by plugging in their desired in-control ATS in Equation (2.9). Both this Equation (2.9) and Table 2.13 give highly accurate values that are more than accurate enough for practical applications.

Note that these control limits are obtained using normally distributed observations. However, the actual underlying distributions are usually unknown to practitioners. Under our proposed robust GLR control chart (the χ^2 CDF chart with $df = 2$ and $b = 4.25$), the obtained h will give an actual in-control ATS value for non-normal distributed observations that is reasonably close to the in-control ATS value for normally distributed observations.

Table 2.13: Values of h corresponding to specified values of the in-control ATS for the χ^2 CDF chart with $df = 2$ and $b = 4.25$, and $n = 1$, $m_1 = 400$, $m_2 = 0$

In-control ATS	h
50	6.85
75	7.66
100	8.28
150	9.19
200	9.87
300	10.86
400	11.56
500	12.11
750	13.1
1000	13.79
1500	14.73
2000	15.36
3000	16.2
5000	17.14
7500	17.77
10,000	18.14

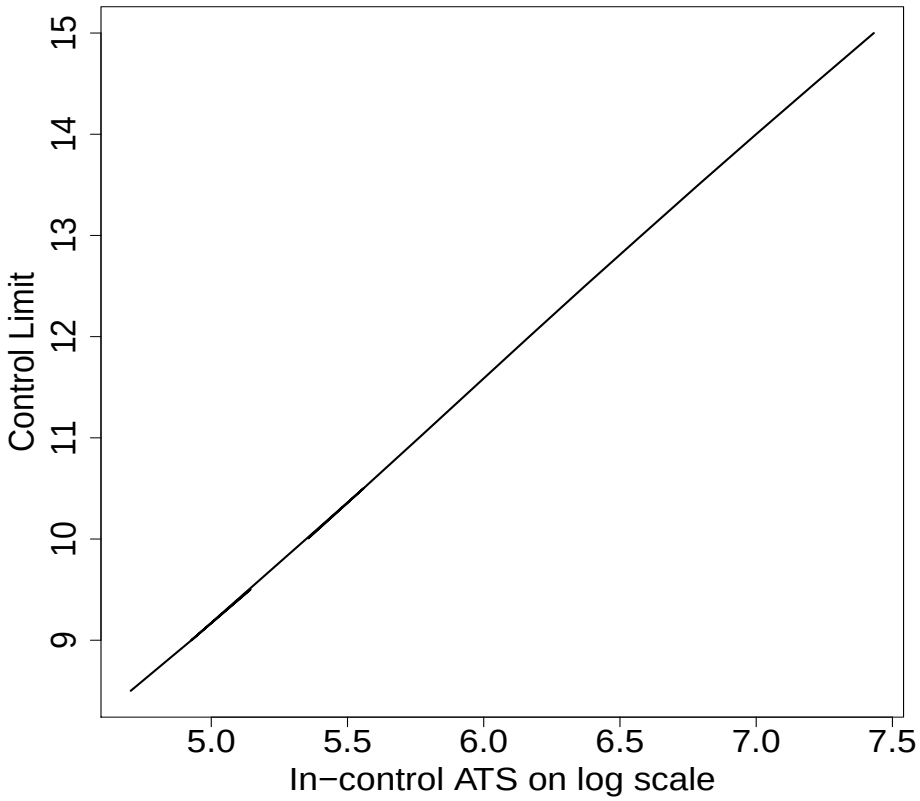


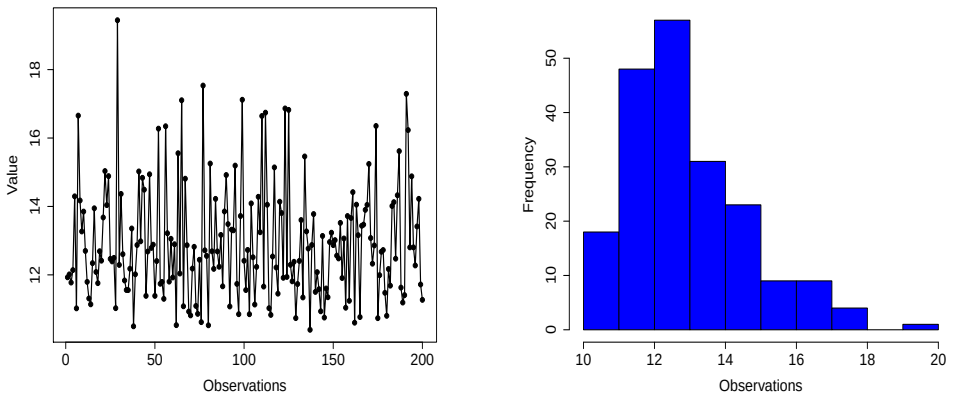
Figure 2.3: The control limit h of the χ^2 CDF chart with $df = 2$ and $b = 4.25$ vs. the in-control ATS on a natural log scale ($n = 1, m_1 = 400, m_2 = 0$).

2.6 An illustration of the application of the χ^2 CDF chart

In this section a numerical example will be used to illustrate how the robust χ^2 CDF chart could be used in Phase I and Phase II.

To first illustrate Phase I, 200 data points are simulated from a gamma(3) distribution with mean ($\mu_0 = 13$) and variance $\sigma_0^2 = 3$. These 200 observations and the corresponding histogram are shown in Figure 2.4. The histogram is skewed to the right and clearly shows non-normality. Many practitioners in such a situation would not have the resources available to fit an appropriate distribution and develop a corresponding control chart, so our robust GLR chart should be a useful option.

If the in-control parameter values μ_0 and σ_0 are unknown then it would be necessary to estimate them using the Phase I data and try to determine whether the process was in control



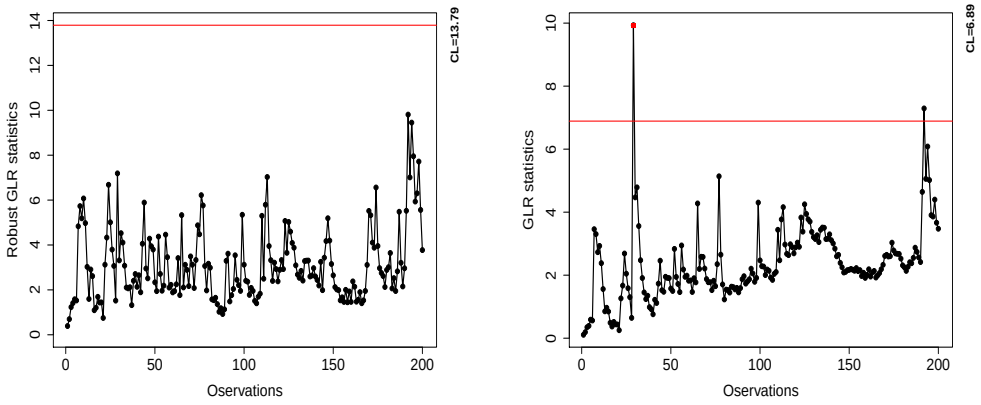
(a) Scatter plot (b) Histogram

Figure 2.4: Observations in Phase I.

when the data were collected. Since the observations do not appear to be from a normal distribution, it is appropriate to use robust estimators of μ_0 and σ_0. The median and median absolute deviation were used here, which gave $\hat{\mu}_0 = 12.64$, and $\hat{\sigma}_0 = 1.53$. Our GLR chart is designed for Phase II monitoring by assuming the parameter mean in Phase I (and overall variance) have already been accurately estimated or known in advance. In this illustration section we inevitably face the issue of Phase I estimation, so we used the sample median and median absolute deviation, which is a relatively simple, widely-used and robust estimators for skewed distribution. Practitioners can apply a different estimator such as Hodges-Lehmann estimator. The issue of estimation in Phase I is beyond the scope of this paper.

Let us apply χ^2 CDF chart with $df = 2$ and $b = 4.25$. For easy comparison, I choose the control limits when the in-control ATS is 1000, which corresponds to 13.79 from Table 2.13. For the standard GLR chart, the control limit is 6.89 obtaining from Table 2 in Reynolds Jr and Lou (2010). These two control charts are shown in Figure 2.5 for Phase I data. The χ^2 CDF chart does not give an indication of a mean change, but the standard GLR chart gives first false alarms at 29th observation. This illustrates the fact that the standard GLR is not robust to non-normal processes.

I also checked for any change in σ in Phase I by applying a robust CUSUM chart of absolute deviations from the target with standard deviation shift size $\delta_\sigma = 1.5$. The control



(a) Robust χ^2 CDF GLR (b) Standard GLR

Figure 2.5: Robust χ^2 CDF GLR and standard GLR charts for mean in Phase I.

limits for this $C|X|$ chart is $10.89\sigma_0/\sqrt{2}$ for in-control ATS = 1000, obtained from Table XII, see Reynolds and Stoumbos (2010). This chart did not show any change in σ, so we conclude that the process was in control during Phase I. Practitioners can use a different chart to monitor the process variance. In this work, we simply choose this robust method and make sure that the Phase I data is indeed in control. The main focus of this chapter is on the detection performance in Phase II. The performance of this variance chart is not intended to be investigated or compared with other variance charts.

There are 50 observations simulated in Phase II. The first 25 observations are simulated in the same way as Phase I and then at time 26, the mean simply shifts to 14.75, while the variance keeps the same. The plot of the out-of-control observations is shown in Figure 2.6. And the mean shift size is $\delta = 1.75/\sqrt{3} = 1.01$.

The robust χ^2 CDF chart with $df = 2$ and $b = 4.25$ is applied to the Phase II observations and shown in Figure 2.7a. The robust CUSUM mean chart is also applied for purposes of comparison, and the results are also shown in Figure 2.7b. The χ^2 CDF chart signals at 32th observation, while the CUSUM chart signals at 37th observation. They are seven and twelve observations later from the true change point. This is consistent with our previous conclusions that the robust χ^2 CDF chart with $df = 2$ and $b = 4.25$ has better out-of-control performance than the CUSUM chart when shifts are medium or large.

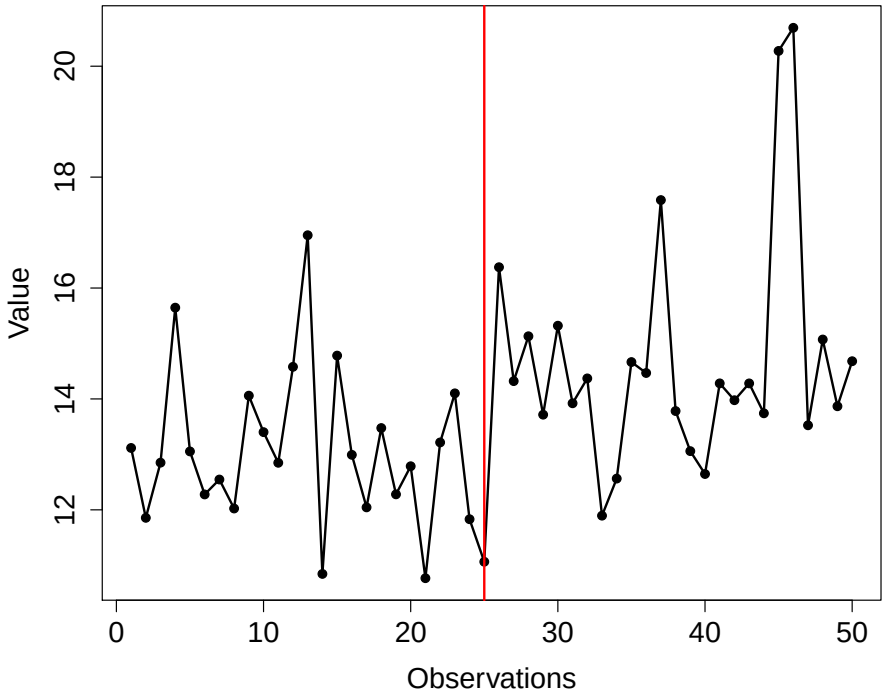
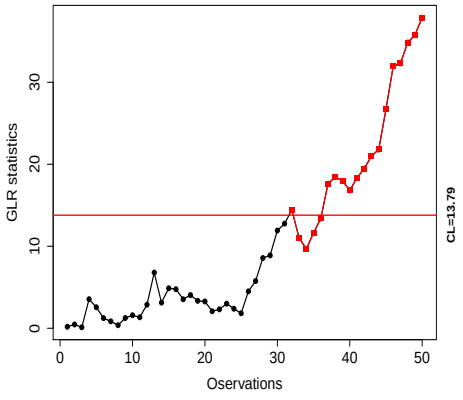
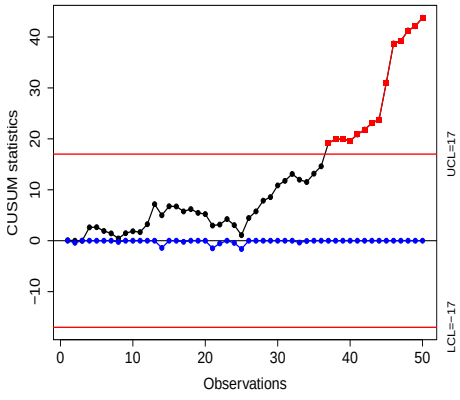


Figure 2.6: Observations in Phase II



(a) Robust χ^2 CDF GLR



(b) Robust CUSUM

Figure 2.7: Robust χ^2 CDF GLR and standard GLR charts for mean in Phase II.

2.7 Conclusions and Discussion

In this chapter, I have shown that the standard GLR control chart based on samples of $n = 1$ is generally not robust to non-normally distributed observations. Even though the GLR chart is not that robust, It appears to be still more robust than the traditional Shewhart chart based on individual observation. But the GLR chart is less robust than the CUSUM chart tuned to detect moderate size shifts.

The CUSUM chart can be made to be very robust by tuning it to detect very small shifts, but then it will perform well only for detecting small shifts. The robust GLR-type charts that investigated are not as robust as this CUSUM chart but offer much better performance in detecting a wide range of shifts.

The best robust GLR control chart that has been investigated is the χ^2 CDF chart with parameters $df = 2$ and $b = 4.25$. This chart gives a relatively robust performance for various non-normally distributed observations, as well as for detecting a wide range of shifts in the process mean. Besides, it is shown that this robust chart also performs better than the standard GLR even when observations are normal when the shift size is small or medium. The significant advantage of this robust GLR control chart is that practitioners do not need to worry about the true distribution of observations and do not need to specify the values of multiple control charts parameters. The only parameter that practitioners need to specify is the in-control ATS, and then they can look up the control limit in Table 2.13. If the value is not in this table, they can calculate the corresponding control limit quickly using Equation (2.9).

There are possible modifications of the robust control charts proposed here that could offer improved performance, such as using a combination of the developed methods. For example, one could improve the robustness of the χ^2 CDF chart with $df = 2$ and $b = 4.25$ by using a sample size larger than one ($n > 1$). I do not intend to cover all possible solutions of robust control charts in this chapter. Instead, I have explored several methods that lead to robust performance which do not sound too complicated to be used in practical applications.

Chapter 3 A Latent Process Approach for Change-point Detection of Mixed-type Observations

3.1 Introduction

In various applications, mixed-type observations are widely present, where continuous and discrete observations are often collected together to evaluate the system performance. For example, in a wafer lapping manufacturing process (Deng and Jin, 2015), the total thickness variation is a continuous quality response to characterize the range of the wafer thickness. The conformity of site total indicator readings (STIR) is a binary response measurement to indicate whether the STIR is larger than the tolerance or not. In the service industry, the average time of job completion and the daily base job frequency are continuous and discrete measurements to monitor its service process in terms of freight amount (Ning and Tsung, 2012). In the example of the civil unrest study (O'brien, 2010; Ramakrishnan et al., 2014), social events are recorded with several measurements, including the frequency of event happened during a certain time interval and a continuous quantity of the average tone of the events. It is known that multiple measurements from a system are often highly correlated and thus can be analyzed simultaneously to monitor the performance of the system effectively.

The change-point problems commonly occur in many processes. Detection of change-points becomes an important task for monitoring the performance of the system. We consider change-points to be those time points which divide data set into distinct homogeneous segments. Generally, change-point analysis can be categorized by offline versus online, parametric versus non-parametric method. This chapter restricts its attention to methods which are offline and parametric. The change-point detection has been developed for over sixty years, with early works including Page (1954), Shiryaev (1963), and Hinkley and Hinkley (1970). Numerous applications are shown on a wide range of disciplines, such as manufacturing pro-

cessing (Ge and Smyth, 2000), bioinformatic applications (Lio and Vannucci, 2000; Erdman and Emerson, 2008), the detection of malware within software (Young and Kuo, 2001), and finance (Spokoiny, 2009). For a more general overview of change-point detection methods, please refer to Eckley et al. (2011), Müller and Siegmund (1994), and Chen and Gupta (2011).

For the continuous observations, various frequentist and Bayesian approaches are developed to detecting change-points in the literature, including Hinkley and Hinkley (1970), Chen and Gupta (1997), Gupta and Chen (1996), Andrieu et al. (2010), Cappé et al. (2009), Fearnhead (1998), Frühwirth-Schnatter (2006), and Whiteley et al. (2010). The frequentist approaches mainly focus on the likelihood-ratio based or penalized likelihood methods, where Bayesian approaches rely on the specification of a prior for the number and position of change-points (Eckley et al., 2011). Among different methodologies, the Bayesian method using switching state-space model (SSSM) shows promising performance. And the Markov chain Monte Carlo (MCMC) method is used for efficient sampling of the posterior distribution for inference (Cappé et al., 2009; Frühwirth-Schnatter, 2006; Eckley et al., 2011). The SSSM assumes that the observations are generated from a latent Markov process, which can be continuous or discrete. If the latent process is continuous, one can use a particle filter approach, sequential Monte Carlo (SMC), to design efficient high-dimensional proposals within the MCMC scheme (Andrieu et al., 2010). If the latent process is discrete, the discrete particle filter (DPF) is an efficient algorithm for the estimation and inference of the latent parameters and has been successfully applied in a variety of fields, see Fearnhead (1998), Whiteley et al. (2010), and Cappé et al. (2009).

However, the change-point problem has not yet received many attentions when there are both discrete or mixed observations. Little work has been done on the change-point detection problem for mixed-type observations. In terms of modeling of mixed-type observations, it is mentioned in Chen and Brown (2013) that the linear Gaussian state-space model can be extended to accommodate the discrete or mixed observation. It is also known that the copula model offers a universal framework to model statistical dependencies among continuous, discrete, or mixed-type random variables (Chen, 2013). de Leon and Wu (2011) develop

a copula-based regression model for a binary and continuous observations, where a latent variable formulation is adopted for the binary observations. However, the model is specially developed for binary and continuous observations, and it is difficult to extend for other types of discrete observations.

For the aforementioned modeling methods, it is also not clear how to combine the modeling method with the change-point detection. There are several frequentist approaches proposed to address the change-point problem for mixed-type data. Ning and Tsung (2012) develop a density-based statistical process control scheme to detect process changes in mixed-type observations. Their key idea is to transform the multi-dimensional observations into a one-dimensional measurement using a local outlier factor (LOF). However, such a method requires several predetermined parameters such as control limits, which could highly depend on the quantity and quality of the data. In Qiu (2008), the mixed-type data are all converted into binary or categorical variables and their distributions are estimated using the log-linear modeling approach. Thus the change-point is found based on changes in the estimated distributions. But this approach requires a reasonable amount of high-quality in-control data to give an accurate estimation of the in-control distribution, which might be difficult to obtain in practice.

In this chapter, I propose a Bayesian approach for the change-point detection with mixed-type observation. Specifically, a switching state-space model is developed for a mixed-type observation with both continuous and discrete observations. The proposed model contains two latent Markov processes, the continuous and discrete latent processes, such that they can jointly describe the unobserved behaviors in the continuous and discrete observations. In particular, a sequence of indicator variables is introduced to indicate significant changes occurred in the data sequence. To enable efficient estimation of the posterior distributions of the latent process, I develop an effective procedure by combining sequential Monte Carlo and discrete particle filter algorithms iteratively, The contributions of this chapter include: (1) to provide a model to quantify the relationship between continuous and discrete variables under various settings of underlying distributions, (2) to propose a combined particle filter algorithm for change-point(s) and parameter estimation, and (3) to conduct Bayesian inference through

MCMC approach.

The rest of this chapter is organized as follows. In Section 3.2, I detail the proposed SSSM model for mixed-type data. Section 3.3 elaborates the efficient sampling algorithm, as well as the pseudo-code of the particle marginal Metropolis-Hastings sampler. Section 3.4 reports several numerical examples to demonstrate the performance of the proposed schemes. A real case study of civil protest is used to illustrate the implementation of the proposed approach in Section 3.5. Finally, a conclusion is presented with some discussion in the future research.

3.2 Switching State-Space Models for Mixed-type Data

3.2.1 Notation

For notation convenience, the capital letters are used for random variables and lowercase letters denote their values. Suppose there are two sequences of observations $\{Y_n; n = 1, \ldots, T\} \subset \mathcal{Y}^{\mathbb{N}}$ and $\{Z_n; n = 1, \ldots, T\} \subset \mathcal{Z}^{\mathbb{N}}$, where $T \geqslant 1$ is the length of the sequence. Here the Y_n is a continuous variable and Z_n is non-negative discrete variable. It is also assumed that the pair Y_n and Z_n are dependent with each other in some unobserved manner. Throughout this chapter, denote $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ as some static parameters involved in the proposed model, which may be multidimensional.

Denote $0 < \tau_1 < \tau_2 < \cdots < \tau_k < T$ to be k change-point locations. It implies that the observations $\{Y_n, Z_n\}$ are homogeneous within each segment $[\tau_j, \tau_{j+1}]$ and heterogeneous across segments. To quantify the relationship between Y_n and Z_n, two latent processes are considered, a continuous latent process $\{X_n\}_{n\geqslant 1}$ and a discrete latent process $\{I_n\}_{n\geqslant 1}$. The values of X_n lie in a real-valued set $\mathcal{X}$ and the values of I_n belong to a finite set $\mathcal{I}$. The process $\{X_n\}_{n\geqslant 1}$ is to explain the dependence between Y_n and Z_n, while the process $\{I_n\}_{n\geqslant 1}$ is to indicate which states that each observation belong to. That is, change-point occurs when two consecutive sets of observations are in different states. Note that if I_n only takes one value, then there will be no change-points anymore. Under the context of change-point detection, we assume that I_n takes at least two different possible values. The value of I_n maintains the

same within each segment and varies across different segments.

Hereafter, let us denote $\boldsymbol{y}_{1:T}$ as a observed vector $\boldsymbol{y}_{1:T} = (y_1, \ldots, y_T)'$ from the random vector $\boldsymbol{Y}_{1:T} = (Y_1, \ldots, Y_T)'$. Similarly, one can define $\boldsymbol{z}_{1:T}$, $\boldsymbol{x}_{1:T}$, and $\boldsymbol{i}_{1:T}$. For a given value of n, the support of $\boldsymbol{I}_{1:n}$ is $\mathcal{I}^n$. Denote $|\mathcal{I}|$ as the cardinality of $\mathcal{I}$. So $\mathcal{I} = \{1, \cdots, |\mathcal{I}|\}$. As n increases, the possible paths of $\boldsymbol{I}_{1:n}$, which is $|\mathcal{I}|^n$ grows exponentially. Thus, we define a parameter N_1 as the maximum number of support points at each time step n. Similarly, N_2 is used as the number of sampling points of $\boldsymbol{X}_n$ at time n. Let us also denote the indicator function as $\delta_A(x)$, which takes value of 1 if $x \in A$, and 0 otherwise.

3.2.2 Proposed Model

Recall that the observed sequences are $\{Y_n\}$ and $\{Z_n\}$, and the latent sequences are $\{I_n\}$ and $\{X_n\}$, where $n \geqslant 1$. The main objective of this chapter is to find the possible change-point location(s), $0 < \tau_1 < \tau_2 < \cdots < \tau_k < T$, $1 \leqslant k \leqslant T$, such that the latent discrete process I_n changes from one state to another. which can be achieved by the estimation of the latent process I_n.

We assume that both latent processes are Markov processes with initial values as $I_1 \sim v_{\boldsymbol{\theta}}(\cdot)$ and $X_1 \sim \mu_{\boldsymbol{\theta}}(\cdot)$, respectively. Their transition probability densities are denoted as $f^I_{\boldsymbol{\theta}}$ and $f^X_{\boldsymbol{\theta}}$, respectively,

$$I_{n+1}|(I_n = i) \sim f^I_{\boldsymbol{\theta}}(\cdot|i), \tag{3.1}$$

$$X_{n+1}|(X_n = x) \sim f^X_{\boldsymbol{\theta}}(\cdot|x), \tag{3.2}$$

where $f^I_{\boldsymbol{\theta}}$ is in fact a stochastic transition matrix. Given $\{I_n\}$ and $\{X_n\}$, I assume that $\{Y_n\}$ and $\{Z_n\}$ are conditionally independent, where the distributions of Y_n and Z_n only depend on the current latent observations of I_n and X_n, for all $n \geqslant 1$, similar to the convention in Whiteley et al. (2010) and Andrieu et al. (2010). This is due to the Markov independence property that knowing the state at any time makes the past, present and future observations statistically independent. By denoting $g_{\boldsymbol{\theta}}(y|x, i)$ and $h_{\boldsymbol{\theta}}(z|x, i)$ as the common marginal

probability densities of Y_n and Z_n given the latent processes, we can have

$$Y_n|(X_1 = x_1, \ldots, X_n = x_n, I_1 = i_1, \ldots, I_n = i_n) \sim g_{\boldsymbol{\theta},i_n}(\cdot|x_n), \tag{3.3}$$

and,

$$Z_n|(X_1 = x_1, \ldots, X_n = x_n, I_1 = i_1, \ldots, I_n = i_n) \sim h_{\boldsymbol{\theta},i_n}(\cdot|x_n). \tag{3.4}$$

It is seen that the relationship between Y_n and Z_n is well defined through Equations (3.3) and (3.4). Changes in the two latent processes I_n and X_n have a direct impact on Y_n and Z_n simultaneously. Thus the mixed-type observations are closely connected through these latent processes. Correspondingly, the complete formulation of the proposed switching state-space model for mixed-type observations is described through Equation (3.1) to Equation (3.4). We call the proposed model as *mixed SSSM*.

The *mixed SSSM* generalizes the original SSSM to observations containing both continuous and discrete variables through two types of latent processes. In the proposed model, the continuous latent process not only controls the latent dynamics but also is developed as a bridge to connect mixed-type observations. Moreover, the proposed model retains the Markov properties and gives more flexibilities on data structures. However, due to the mixture property, the *mixed SSSM* is clearly non-linear and non-Gaussian, which makes the traditional particle filter algorithm no longer appropriate for the model estimation and inference.

Example: A linear mixed Gaussian-Poisson SSSM. Suppose the discrete latent process $\{I_n\}$, where $I_n \in \{0, 1\}$, is a Markov chain on with transition matrix P_I as

$$P_I = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.8 \end{bmatrix}. \tag{3.5}$$

For the continuous latent process $\{X_n\}$, consider the following transition relationship,

$$x_{n+1} = \phi x_n + \sigma I_n V_n, \tag{3.6}$$

where $\{V_n\}$ are independent and identically distributed (i.i.d.) with normal distribution $\mathcal{N}(0,1)$. For the initial distributions, set $I_1 \sim Bern(0.5)$ and $X_1 \sim \mathcal{N}(0,1)$.

With the observed continuous process $\{X_n\}$ and the observed discrete process $\{Z_n\}$. For n = 1, ..., T, consider the mixed Gaussian-Poisson SSSM as follows,

$$y_n = x_n + \gamma V_n, \tag{3.7}$$

$$z_n \sim Poisson(\alpha\sqrt{|x_n|}). \tag{3.8}$$

It means that, given I_n and X_n, the continuous variable Y_n follows a Gaussian distribution, $\mathcal{N}(x_n, \gamma^2)$ and the discrete variable Z_n follows a Poisson distribution with parameter $\alpha\sqrt{|x_n|}$. Clearly, the static parameter vector $\boldsymbol{\theta}$ contains ϕ, σ, γ, and α in this example.

3.2.3 Model Inference

Based on Equations (3.3) and (3.4), we can obtain,

$$g_\theta(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}) = \prod_{n=1}^{T} g_{\theta,i_n}(y_n|x_n), \tag{3.9}$$

$$h_\theta(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}) = \prod_{n=1}^{T} h_{\theta,i_n}(z_n|x_n). \tag{3.10}$$

Consequently, we can get the joint distribution of $\boldsymbol{y}_{1:T}$ and $\boldsymbol{z}_{1:T}$ given $\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}$,

$$p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}) = g_{\boldsymbol{\theta},i_{1:T}}(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T}) \times h_{\boldsymbol{\theta},i_{1:T}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T}) = \prod_{n=1}^{T} g_{\boldsymbol{\theta},i_n}(y_n|x_n) h_{\boldsymbol{\theta},i_n}(z_n|x_n). \tag{3.11}$$

Conditional on the observed data $\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}$ for $T \geqslant 1$, the goal is to conduct the Bayesian

inference of all the unknown parameters, especially the change-point locations, $\boldsymbol{\tau}_{1:k}$ for $1 \leqslant k \leqslant T$. If $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ is known, it is easy to see that the Bayesian inference mainly relies on the posterior density,

$$\begin{aligned} p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T}) &\propto p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T},\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T}) \\ &= f^I_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:T}) f^X_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T}|\boldsymbol{i}_{1:T}) g_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}|\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T}) h_{\boldsymbol{\theta}}(\boldsymbol{z}_{1:T}|\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T}) \\ &= v_{\boldsymbol{\theta}}(i_1)\mu_{\boldsymbol{\theta}}(x_1) \prod_{n=2}^{T} f^I_{\boldsymbol{\theta}}(i_n|i_{n-1}) f^X_{\boldsymbol{\theta},i_{n-1}}(x_n|x_{n-1}) \prod_{n=1}^{T} g_{\boldsymbol{\theta},i_n}(y_n|x_n) h_{\theta,i_n}(z_n|x_n). \end{aligned} \tag{3.12}$$

If $\boldsymbol{\theta}$ is unknown, we can assign a suitable prior density $p(\boldsymbol{\theta})$ and then the Bayesian inference can be conducted based on the joint density

$$p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T}) \propto p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T},\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T}) p(\boldsymbol{\theta}). \tag{3.13}$$

To calculate the posterior distributions of latent processes $\{I_n\}$ and $\{X_n\}$, we will propose to alternately combine the DPF and SMC algorithms, which will be detailed in the next section. If the models defined through Equation (3.1) to Equation (3.4) are non-linear or non-Gaussian, there are often times no explicit expressions for those posterior densities, $p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T},\boldsymbol{i}_{1:T},\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T})$ and $p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T},\boldsymbol{z}_{1:T})$, making exact inference difficult in practice. Thus it is natural to resort to approximations, where the particle Markov chain Monte Carlo (PMCMC) method provides a flexible Bayesian framework to address such difficulties.

3.3 Particle MCMC Algorithm for Mixture SSSM

It is known that particle filter algorithms are commonly used to conduct efficient Bayesian inferences under the latent processes context. Usually, the sequential Monte Carlo(SMC) algorithm is applied for the continuous latent process (Andrieu et al., 2010) and the discrete particle filter(DPF) algorithm is used for the discrete latent process (Whiteley et al., 2010; Fearnhead, 1998; Fearnhead and Clifford, 2003). However, since the *mixed SSSM* contains both continuous and discrete observations, neither of these two algorithms can work well in-

dividually. To the best of our knowledge, there is little work focusing on problems with both continuous and discrete observations existing together. To address this challenge, we propose a new algorithm, so-called "combined DPF & SMC" algorithm, by taking advantage of both SMC and DPF algorithms such that we can jointly estimate the unknown parameter $\boldsymbol{\theta}$ and change-point(s) $\tau_1, \ldots, \tau_k$. In this section, I will illustrate the combined DPF & SMC algorithm in detail in Section 3.1. Using the posterior densities estimated from subsection 3.3.1, subsection 3.3.2 will detail on how to update unknown parameters and latent processes using the particle marginal Metropolis-Hastings (PMMH) sampler.

3.3.1 Combined DPF & SMC Algorithm

A combined DPF & SMC algorithm is developed to make Bayesian inference in *mixed SSSM*, conditional upon the mixed observations $\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}$ and treating the static parameter $\boldsymbol{\theta}$ and both the latent processes $\boldsymbol{X}_{1:T}, \boldsymbol{I}_{1:T}$ unknown for some $T \geqslant 1$. Then the joint density of these unknowns is shown in Equation (3.13). It is easy to see that this posterior can be factorized as follows,

$$p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T} | \boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T} | \boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p_{\boldsymbol{\theta}}(\boldsymbol{x}_{1:T} | \boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}, \boldsymbol{i}_{1:T}) \tag{3.14}$$

where

$$p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T} | \boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = \frac{p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T} | \boldsymbol{i}_{1:T}) p(\boldsymbol{i}_{1:T} | \boldsymbol{\theta}) p(\boldsymbol{\theta})}{\int_{\boldsymbol{\Theta}} \sum_{\boldsymbol{i}'_{1:T} \in \mathcal{I}^T} p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T} | \boldsymbol{i}'_{1:T}) p(\boldsymbol{i}'_{1:T} | \boldsymbol{\theta}) p(\boldsymbol{\theta}) \, d\boldsymbol{\theta}} \tag{3.15}$$

However, the exact computation of $p(\boldsymbol{\theta}, \boldsymbol{i}_{1:T} | \boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ would be very difficult due to the calculation of summation in the denominator of Equation (3.15) over up to $|\mathcal{I}|^T$ values of $\boldsymbol{i}_{1:T}$. Even for a modest value of T, it would be too expensive to compute the exact summation, even if $\boldsymbol{\theta}$ is treated as fixed (Whiteley et al., 2010). In this case, we consider an approximated

computation of $p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ sequentially via the recursive relationship,

$$p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}) = \frac{p_{\boldsymbol{\theta}}(y_n, z_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}, \boldsymbol{i}_{1:n}) f^I_{\boldsymbol{\theta}}(i_n|\boldsymbol{i}_{1:n-1}) p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n-1}|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})}{\sum\limits_{\boldsymbol{i}'_{1:n} \in \mathcal{I}^n} p_{\boldsymbol{\theta}}(y_n, z_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}, \boldsymbol{i}'_{1:n-1}) f^I_{\boldsymbol{\theta}}(i'_n|\boldsymbol{i}'_{1:n-1}) p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n-1}|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})}. \tag{3.16}$$

However, the computation involved in Equation (3.16) increases exponentially in n. Thus approximates the posterior distributions $\{p(\boldsymbol{\theta}, \boldsymbol{i}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}); 1 \leqslant n \leqslant T\}$ sequentially in time using a collection of $N_1|\mathcal{I}|$ weighted trajectories (so-called "particles"),

$$\left\{\boldsymbol{I}^{(k)}_{1:n}; k = 1, \cdots, N_1|\mathcal{I}|\right\},$$

where $|\mathcal{I}|$ is the cardinality of $\mathcal{I}$ and N_1 controls the precision of the algorithm.

Note that for given n and $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, the support of $p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n})$ is $\mathcal{I}^n$. Specifically, at each time step n, we consider to resample N_1 of the $N_1|\mathcal{I}|$ trajectories and then adjust their weights accordingly. Let us denote by $\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_T$ the random support sets generated from the resampling step. In particular, each $\mathbf{S}_n$ is a subset of $\mathcal{I}^n$ and takes a value of $\mathbf{s}_n$. As n increases, the possible paths of $\boldsymbol{I}_{1:n}$ grows exponentially. Then N_1 acts as a pruning parameter to prevent the support from growing too big, through resampling techniques when $|\mathbf{S}_n| > N_1$. On the other hand, the value of N_1 controls the precision of the DPF algorithm. A larger value of N_1 will lead to more accurate (on average) approximation for the target distribution $p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$, and it has been shown that the DPF algorithm works efficiently even with a moderate number of particles (Whiteley et al., 2010; Chen and Liu, 2000; Doucet et al., 2000, 2001b).

Since the objective is to find the change-point(s) locations, we need the approximation of the posterior distribution of $\boldsymbol{I}_{1:n}$, which is approximated by a set of N_1 weighted particles as follows,

$$\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}) := \sum_{k=1}^{|\mathbf{S}_n|} W^k_n \delta_{\boldsymbol{I}^k_{1:n}}(\boldsymbol{i}_{1:n}), \tag{3.17}$$

where W^k_n is a so-called normalized importance weight associated with the k_{th} particle $\boldsymbol{I}^k_{1:n}$ at

time n such that $\sum_{k=1}^{|\mathbf{S}_n|} W_n^k = 1$. The delta indicator function takes value of 1 if $\boldsymbol{i}_{1:n} \in \boldsymbol{I}_{1:n}^k$, and 0 otherwise. Note that the above approximation requires the estimation of the conditional marginal likelihood $p_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})$. However, due to its nonlinear or non-Guassion characteristics, the commonly used Kalman filter techniques (Kalman, 1960) cannot work here. Therefore, given $\boldsymbol{\theta}$ and $\boldsymbol{I}_{1:n} = \boldsymbol{i}_{1:n}$, the SMC technique is adopt dealing with the continuous latent process $\{X_n\}_{n\geqslant 1}$ to approximate the density $p_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:T}||\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})$.

The SMC algorithm aims to approximate the joint posterior density of the continuous latent process $\boldsymbol{X}_{1:n}$, which is $p_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n})$ by a set of N_2 weighted random samples or particles by a discrete density probability distribution,

$$\hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\mathrm{d}\boldsymbol{x}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n}) := \sum_{k=1}^{N_2} \tilde{W}_n^k \delta_{\boldsymbol{X}_{1:n}^k}(\mathrm{d}\boldsymbol{x}_{1:n}), \tag{3.18}$$

where the $\tilde{W}_n^k$ is the normalized weight for the k^{th} particle at time n such that $\sum_{k=1}^{N_2} \tilde{W}_n^k = 1$, and $\tilde{W}_n^k = \frac{\tilde{w}_n(\boldsymbol{X}_{1:n}^k)}{\sum_{m=1}^{N_2} \tilde{w}_n(\boldsymbol{X}_{1:n}^m)}$, where $\tilde{w}_n(\boldsymbol{X}_{1:n}^k)$ is the corresponding unnormalized weights. And $\delta_{\boldsymbol{X}_{1:n}^k}(\cdot)$ takes value of 1 if $\boldsymbol{x}_{1:n} \in \boldsymbol{X}_{1:n}^k$, and 0 otherwise. N_2 is the number of particles $\boldsymbol{X}_{1:n}^k$ sampled at each time n. Similar to N_1, the choice of N_2 controls the precision of the SMC algorithm. Thus, N_1 and N_2 together controls the precision of the combined DPF &SMC algorithm. Then, at time T, we can obtain the estimation of $p_{\boldsymbol{\theta},\boldsymbol{i}_{1:T}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ by SMC as,

$$\hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:T}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = \hat{p}_{\boldsymbol{\theta},i_1}(y_1, z_1) \prod_{n=2}^{T} \hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_n, \boldsymbol{z}_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}), \tag{3.19}$$

$$\hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(y_n, z_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}) = \frac{1}{N_2} \sum_{k=1}^{N_2} \tilde{w}_n(\boldsymbol{x}_{1:n}^k).$$

For the detailed procedure of the DPF or SMC algorithm, please refer to Andrieu et al. (2010), Fearnhead (1998), and Whiteley et al. (2010) for more details.

A pseudo code of the combined DPF & SMC is directly provided below. To alleviate the notational burden, we adopt the similar convention used in Andrieu et al. (2010) that whenever the index k is used we mean 'for all $k \in \{1, \cdots, N_2\}$'for the continuous particle

$\{X_n\}_{n\geq 1}$. The dependence of weights on $\boldsymbol{\theta}$ is also omitted for convenience.

Algorithm 1 (Combined DPF & SMC Algorithm).

Step 1: *at time* $n = 1$,

(a) Set $\mathbf{S}_1 = \mathcal{I}$ *and for each* $i_1 \in \mathcal{I}$, *obtain* $\hat{p}_{\boldsymbol{\theta},i_1}(y_1, z_1)$ *using the SMC algorithm as follows,*

$$\begin{aligned} \tilde{w}_1(x_1^k) &= g_{\boldsymbol{\theta},i_1}(y_1|x_1^k) h_{\boldsymbol{\theta},i_1}(z_1|x_1^k), \\ \hat{p}_{\boldsymbol{\theta},i_1}(y_1, z_1) &= \frac{1}{N_1} \sum_{k=1}^{N_2} \tilde{w}_1(x_1^k). \end{aligned} \tag{3.20}$$

(b) Compute and normalize the weights for discrete particles. For each $i_1 \in \mathcal{I}$,

$$\begin{aligned} w_1(i_1) &= v_{\boldsymbol{\theta}}(i_1) \hat{p}_{\boldsymbol{\theta},i_1}(y_1, z_1), \\ W_1(i_1) &= \frac{w_1(i_1)}{\sum\limits_{i_1' \in \mathcal{I}} w_1(i_1')}. \end{aligned} \tag{3.21}$$

Step 2: *at times* $n = 2, \ldots, T$, *(a) If* $|\mathbf{S}_{n-1}| \leqslant N_1$ *set* $C_{n-1} = \infty$ *otherwise set* C_{n-1} *to the unique solution of*

$$\sum_{\boldsymbol{i}_{1:n-1} \in \mathbf{S_{n-1}}} 1 \wedge C_{n-1} W_{n-1}(i_{1:n-1}) = N_1 \tag{3.22}$$

(b) Maintain the L_{n-1} *trajectories in* $\mathbf{S}_{n-1}$ *which have weights strictly superior to* $1/C_{n-1}$, *then apply the stratified resampling mechanism to the other trajectories to yield* $N_1 - L_{n-1}$ *survivors. Set* $\mathbf{S}'_{n-1}$ *to the set of surviving and maintained trajectories. (c) Set* $\mathbf{S}_n = \mathbf{S}'_{n-1} \times \mathcal{I}$. *(d) For each* $\boldsymbol{i}_{1:n} \in \mathbf{S}_n$, *obtain* $\hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_n, \boldsymbol{z}_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})$ *using the SMC algorithm by,*

$$\begin{aligned} \tilde{w}_n(\boldsymbol{x}_{1:n}^k) &= g_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(y_n|x_n^k) h_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}^k}(z_n|x_n^k), \\ \hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_n, \boldsymbol{z}_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}) &= \frac{1}{N_1} \sum_{k=1}^{N_1} \tilde{w}_n(\boldsymbol{x}_{1:n}^k). \end{aligned} \tag{3.23}$$

(e) Compute and normalize the weights. For each $\boldsymbol{i}_{1:n} \in \mathcal{S}_n$,

$$w_n(\boldsymbol{i}_{1:n}) = f^I_{\boldsymbol{\theta}}(\boldsymbol{i}_n|\boldsymbol{i}_{1:n-1})\hat{p}_{\boldsymbol{\theta},\boldsymbol{i}_{1:n}}(\boldsymbol{y}_n, \boldsymbol{z}_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1})\frac{W_{n-1}(\boldsymbol{i}_{1:n-1})}{1 \wedge C_{n-1}W_{n-1}(\boldsymbol{i}_{1:n-1})},$$
$$W_n(\boldsymbol{i}_{1:n}) = \frac{w_n(\boldsymbol{i}_{1:n})}{\sum\limits_{\boldsymbol{i}'_{1:n}\in\mathcal{S}_n} w_n(\boldsymbol{i}'_{1:n})}. \tag{3.24}$$

In addition, the proposed combined DPF & SMC algorithm also provides an estimate of the marginal likelihood $p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ given by

$$\hat{p}_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = \hat{p}_{\boldsymbol{\theta}}(y_1, z_1)\prod_{n=2}^{T}\hat{p}_{\boldsymbol{\theta}}(y_n, z_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}), \tag{3.25}$$

where

$$\begin{aligned}\hat{p}_{\boldsymbol{\theta}}(y_1, z_1) &= \sum_{i_1\in\mathcal{I}} w_1(i_1),\\ \hat{p}_{\boldsymbol{\theta}}(y_n, z_n|\boldsymbol{y}_{1:n-1}, \boldsymbol{z}_{1:n-1}) &= \sum_{\boldsymbol{i}_{1:n}\in\mathbf{S}_n} w_n(\boldsymbol{i}_{1:n}), n > 1.\end{aligned} \tag{3.26}$$

Note that $p_{\boldsymbol{\theta}}(\boldsymbol{i}_{1:n}|\boldsymbol{y}_{1:n}, \boldsymbol{z}_{1:n})$ can be computed exactly when n is small. However, when n is large enough that $|\mathbf{S}_{n-1}| > N_1$, the stratified resampling mechanism need to be employed to prune the set of trajectories. One can refer to Whiteley et al. (2010) for more details about the resampling techniques.

3.3.2 Particle marginal Metropolis–Hastings sampler for mixed SSSM

A popular and efficient choice for sampling from $p(\boldsymbol{\theta}, \boldsymbol{X}_{1:T}, \boldsymbol{I}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ is the Particle marginal Metropolis-Hastings (PMMH) widely recommended in literatures for parameter estimations and inference purpose (Andrieu et al., 2010; Whiteley et al., 2010). The PMMH sampler can jointly update the unknown static parameters $\boldsymbol{\theta}$, and two latent processes $\boldsymbol{X}_{1:T}$

and $\boldsymbol{I}_{1:T}$ by,

$$q\left\{(\boldsymbol{\theta}^*, \boldsymbol{x}^*_{1:T}, \boldsymbol{i}^*_{1:T})|(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T})\right\} = q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) p_{\boldsymbol{\theta}^*}(\boldsymbol{x}^*_{1:T}, \boldsymbol{i}^*_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}).$$

In this scenario, the proposed $\boldsymbol{x}^*_{1:T}$ and $\boldsymbol{i}^*_{1:T}$ are perfectly 'adapted' to the proposed $\boldsymbol{\theta}^*$, and the only degree of freedom of the algorithm (which will affect its performance) is $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$. The resulting MH acceptance ratio is given by

$$\frac{p(\boldsymbol{\theta}^*, \boldsymbol{x}^*_{1:T}, \boldsymbol{i}^*_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})}{p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})} \frac{q\left\{(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T})|(\boldsymbol{\theta}^*, \boldsymbol{x}^*_{1:T}, \boldsymbol{i}^*_{1:T})\right\}}{q\left\{(\boldsymbol{\theta}^*, \boldsymbol{x}^*_{1:T}, \boldsymbol{i}^*_{1:T})|(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T})\right\}} = \frac{p^*_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p(\boldsymbol{\theta}^*)}{p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p(\boldsymbol{\theta})} \frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}. \tag{3.27}$$

The expression of ratio in Equation (3.27) suggests that the PMMH algorithm effectively targets the marginal density $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) \propto p_{\boldsymbol{\theta}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p(\boldsymbol{\theta})$, justifying the MMH terminology. Moreover, it bypasses the difficulty of samping from $p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}, \boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ by sampling from $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}, z_{1:T})$, which is typically defined on a much smaller space and can be approximated using **Algorithm 1**. The proposed PMMH sampler is summarized as follows.

Algorithm 2 (PMMH Sampler).

Step 1: *initialization,* $j = 0$,

(a) set $\boldsymbol{\theta}(0)$ *arbitrarily and*

(b) run **Algorithm 1** *targeting both* $p_{\boldsymbol{\theta}(0)}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ *and* $p_{\boldsymbol{\theta}(0)}(\boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$, *sample* $\boldsymbol{i}_{1:T}(0) \sim \hat{p}_{\boldsymbol{\theta}(0)}(\cdot|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ *and let* $\hat{p}_{\boldsymbol{\theta}(0)}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ *denote the marginal likelihood estimate.*

Step 2: *for iteration* $j \geqslant 1$,

(a) sample $\boldsymbol{\theta}^* \sim q\left\{\cdot|\boldsymbol{\theta}(j-1)\right\}$,

(b) run **Algorithm 1** *targeting both* $p_{\boldsymbol{\theta}^*}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ *and* $p_{\boldsymbol{\theta}^*}(\boldsymbol{i}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$, *sampling* $\boldsymbol{i}^*_{1:T} \sim \hat{p}_{\boldsymbol{\theta}^*}(\cdot|\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$, *and let* $\hat{p}_{\boldsymbol{\theta}^*}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ *denote the marginal likelihood estimate, and*

(c) with probability

$$1 \wedge \frac{\hat{p}_{\boldsymbol{\theta}^*}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p(\boldsymbol{\theta}^*)}{\hat{p}_{\boldsymbol{\theta}(j-1)}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) p\left\{\boldsymbol{\theta}(j-1)\right\}} \frac{q\left\{\boldsymbol{\theta}(i-1)|\boldsymbol{\theta}^*\right\}}{q\left\{\boldsymbol{\theta}^*|\boldsymbol{\theta}(j-1)\right\}} \tag{3.28}$$

set $\boldsymbol{\theta}(j) = \boldsymbol{\theta}^*$,$\boldsymbol{i}_{1:T}(j) = \boldsymbol{i}^*_{1:T}$ *and* $\hat{p}_{\boldsymbol{\theta}_{(j)}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = \hat{p}_{\boldsymbol{\theta}^*}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$*; otherwise set* $\boldsymbol{\theta}(j) = \boldsymbol{\theta}(j-1)$, $\boldsymbol{i}_{1:T}(j) = \boldsymbol{i}_{1:T}(j-1)$ *and* $\hat{p}_{\boldsymbol{\theta}_{(j)}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T}) = \hat{p}_{\boldsymbol{\theta}_{(j-1)}}(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$.

3.4 Simulation Study

In this subsection, simulation studies are conducted to evaluate the performance of the proposed method for change-point detection. Four scenarios of mixed-type data based on the mixed SSSM will be considered: (S1) mixed Gaussian-Bernoulli, (S2) mixed Gaussian-Poisson, (S3) mixed Gaussian-Gaussian, and (S4) mixed Gaussian-Noncentral t. For each scenario, two cases are considered: one change-point and multiple change-points. Moreover, I also consider three different locations when the change occurs: the beginning of the time period, the middle of the time period, and the end of the time period. Each simulation setting is repeated for 50 iterations.

3.4.1 Data Generation

For simplicity, I assume that Y_n is a continuous random variable following a Gaussian distribution, while Z_n is a discrete random variable following either Bernoulli in (S1) or Poisson distribution in (S2). Since the proposed method also can accommodate the situation with Y_n and Z_n are continuous, I also consider Z_n follows Gaussian in (S3) or noncentral t distribution (S4) in the simulation study. Suppose that $\mathcal{I} = \{0, 1\}$. It means that there are two states for the Markov chain, $\{I_n\}$, with the transition matrix P_I as

$$P_I = \begin{bmatrix} p_1 & 1-p_1 \\ 1-p_2 & p_2 \end{bmatrix}, \tag{3.29}$$

where $P(I_n = 1|I_{n-1} = 0) = 1 - p_1$, $P(I_n = 0|I_{n-1} = 0) = p_1$, $P(I_n = 0|I_{n-1} = 1) = 1 - p_2$, $P(I_n = 1|I_{n-1} = 1) = p_2$, for $n > 1$. Note that p_1 and p_2 are also unknown parameters

need to be updated together with other unknown static parameters. Suppose that the true change-point position $\tau(s)$, which means that the latent discrete variable changes from one state to another at time $t = \tau(\text{s})$. Four scenarios of data generation based on the mixed SSSM are listed as follows,

(S1) **Mixed Gaussian-Bernoulli**

$$\begin{aligned} x_{n+1} &= \phi x_n + \sigma I_n V_n, \\ y_n &= x_n + \gamma V_n, \\ z_n &\sim Bern(p(I_n)), \end{aligned}$$

where $p(I_n) \sim Unif[0.1 + 0.5I_n, 0.4 + 0.5I_n]$. Thus, if $I_n = 0$, the parameter p in the Bernoulli distribution follows a uniform distribution with support [0.1, 0.4]. While, if $I_n = 1$, the support is [0.6, 0.9].

(S2) **Mixed Gaussian-Poisson**

$$\begin{aligned} x_{n+1} &= \phi x_n + \sigma I_n V_n, \\ y_n &= x_n + \gamma V_n, \\ z_n &\sim Poisson(\sqrt{|x_n|}). \end{aligned}$$

(S3) **Mixed Gaussian-Gaussian**

$$\begin{aligned} x_{n+1} &= \phi x_n + \sigma I_n V_n, \\ y_n &= x_n + \gamma V_n, \\ z_n &= x_n + V_n. \end{aligned}$$

(S4) **Mixed Gaussian-Noncentral** t

$$x_{n+1} = \phi x_n + \sigma I_n V_n,$$
$$y_n = x_n + \gamma V_n,$$
$$z_n \sim T(df, x_n),$$

where $T(df, \delta)$ denotes the Noncentrol t distribution with degree of freedom df and noncentrality parameter δ. In the simulation example, $df = 4$ will be used.

In the above four scenarios, $\{V_n\}$ are i.i.d. and $\boldsymbol{\theta} = (\phi, \sigma, \gamma, p_1, p_2)$. For the initialization of SSSM, assume the initial distribution is $X_1 \sim \mathcal{N}(0, 1)$. The initial distribution of I_n is Bernoulli with probability 0.01, i.e., $I_1 \sim Bern(0.01)$. Such a small value in the initial density is used to ensure the initial state is always 0 for easy comparison. Based on the true change-points $\tau_1, \ldots, \tau_k$, I first generate the state process $\boldsymbol{I}_{1:T}$, where T is set as $T = 100$. Then two sets of observations $(\boldsymbol{y}_{1:T}, \boldsymbol{z}_{1:T})$ are generated according the corresponding mixed SSSMs in (S1)-(S4) with $\phi = 0.9, \sigma = 4, \gamma = 1$. Here I use a possible proposal for combined DPF & SMC sampling, i.e. $q_\theta(x_1) = \mu_\theta(x_1)$ and $q_\theta(x_n|y_n, z_n, x_{n-1}) = f_\theta^X(x_n|x_{n-1})$ for $n = 2, \ldots, T$. For the prior of $\boldsymbol{\theta}$, the following independent priors are assigned, $logit(\phi) \sim \mathcal{N}(\mu_\phi, \sigma_\phi^2)$, $\log(\sigma) \sim \mathcal{N}(\mu_\sigma, \sigma_\sigma^2)$, and $\gamma \sim \mathcal{N}(\mu_\gamma, \sigma_\gamma^2)$, where $logit(x) = log(\frac{x}{1-x})$. For $0 < p_1 < 1$ and $0 < p_2 < 1$, the same Dirichlet distribution $Dir([1\ 1])$ is used. Let $N_1 = 128$, $N_2 = 200$ in Algorithm 1. The number of MCMC simulation is $N = 300$ with burn-in 200. A normal random-walk Metropolis-Hastings proposal is used in Algorithm 2 to update the parameters jointly, with the covariance of the proposal proportional to the identity matrix up to some small constant, such as 0.3.

3.4.2 Results for One Change-point Detection

For the case of only one change-point τ, the transition matrix needs certain restriction such that I_n can only change from 0 to 1 once, and not the other way around. This can be accomplished by setting $P(I_n = 0|I_{n-1} = 1) = 0$. Thus the transition matrix in Equation (3.29) can

be rewritten as

$$P_I = \begin{bmatrix} p & 1-p \\ 0 & 1 \end{bmatrix}, \tag{3.30}$$

where the value of $1-p$ is close to zero, ensuring that there is only one change-point in the whole process. Recall that the data contain $T = 100$ observations collected at locations $1, \ldots, T$, respectively. If the change-point location is τ, then the state becomes,

$$I_n = \begin{cases} 0 \quad \text{for} \quad n = 1, \cdots, \tau, \\ 1 \quad \text{for} \quad n = \tau + 1, \cdots, T. \end{cases} \tag{3.31}$$

For each scenario, I consider three different change-point locations $\tau = 21, 51, 91$, separately. Note that the unknown static parameters are $\boldsymbol{\theta} = (\phi, \sigma, \gamma, p)$ with true values set as $\phi = 0.9, \sigma = 4.0, \gamma = 1.0$. Table 3.1 reports the estimation results of ϕ, σ, γ and change-point τ under each model, which are the medians based on 50 iterations. The standard deviations of the estimates of τ, $sd(\hat{\tau})$ are also listed. Compared with the true values of change-point (τ) listed in the first row of the table, the estimated change-points are accurate with only one or two times delay across all scenarios. Their standard deviations are about 1 when the process changes at a relatively early time stage ($\tau = 21$), and slightly increase if change-point occurs late. The standard deviations are relatively large when the true change-point is $\tau = 91$. One possible explanation is that the accuracy of the change-point estimation is highly related to the estimation of static parameters. However, there are only nine data points after the change-point. Such little data information after the change makes it difficult to give an accurate estimation of static parameters. It can be seen in the table that the estimation for static parameters is relatively worse when $\tau = 91$.

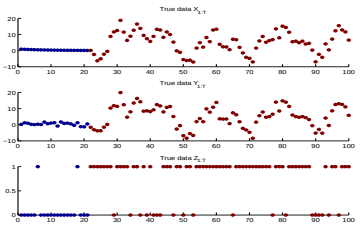
Moreover, the acceptance rates and computation time (mins) are also shown in the table. The acceptance rates are well maintained around 20% to 32%, which is within a reasonable range for efficient MCMC updates. A high acceptance rate means that nearly all candidate

samples occur right around the current data point. Thus the Markov chain is moving rather slowly and not exploring the parameter space fully. On the other hand, a low acceptance rate means that the proposed samples are always rejected and the chain is not moving much. An efficient Metropolis sampler should have an acceptance rate that is neither too high nor too low. Roberts et al. (1997) shows that for random-walk Metropolis algorithms, the optimal acceptance probability for the Markov chain should be around 0.234 in high dimensions. Besides, it is also noted that the simulation time for mixed-type observations (S1, S2) is competitive compared with that for continuous observations (S3, S4).

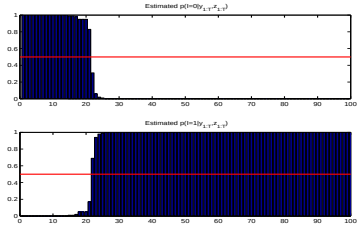
In addition, the data simulated under each scenario are shown in Figures 3.1 to 3.4. The plotted data includes the continuous latent process $\boldsymbol{X}_{1:T}$, the continuous observation $\boldsymbol{Y}_{1:T}$ and the other observation sequence $\boldsymbol{Z}_{1:T}$. At each time point n, $1 \leqslant n \leqslant T$, the probability of its corresponding state $p(I_n = 1)$ and $p(I_n = 0) = 1 - p(I_n = 1)$ is calculated based on 100 estimates of the latent process $\boldsymbol{I}_{1:T}$ from MCMC simulations, and their plots are also shown in these figures. These plots show that the process is in state 0 at the beginning with a high probability of $p(I_1 = 0)$, and this high probability value is maintained until a certain time point when it decreases drastically to almost 0. Thus this time point is estimated as the change-point.

Table 3.1: Simulation results for one change-point scenario

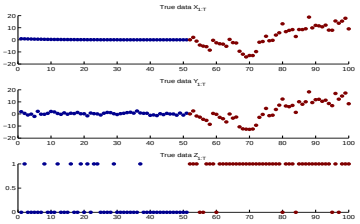
	S1			S2			S3			S4		
τ	21	51	91	21	51	91	21	51	91	21	51	91
$\hat{\tau}$	22	53	93	23	53	93	23	53	93	23	53	93
$sd(\hat{\tau})$	0.93	1.91	3.05	0.84	1.99	6.39	0.62	1.04	5.48	1.03	2.15	6.27
$\hat{\phi}$	0.82	0.81	0.73	0.90	0.89	0.89	0.88	0.87	0.83	0.85	0.87	0.79
$\hat{\sigma}$	3.36	3.80	3.03	3.94	3.82	2.64	4.01	4.01	3.38	4.22	4.11	3.05
$\hat{\gamma}$	1.63	1.06	1.10	1.03	1.01	1.06	0.97	0.99	0.97	1.46	1.17	1.04
acceptance rate	0.24	0.23	0.20	0.18	0.18	0.20	0.19	0.22	0.25	0.28	0.28	0.32
computation time (mins)	7.63	8.73	9.03	6.59	11.62	12.55	5.38	6.51	7.27	8.45	9.83	11.66



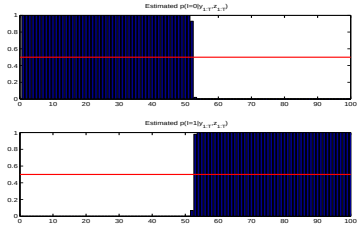
(a) Simulation Data with $\tau = 21$



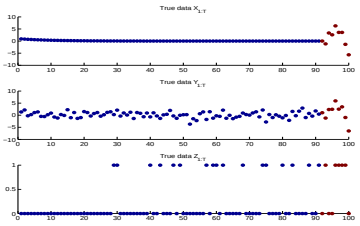
(b) Change point estimation



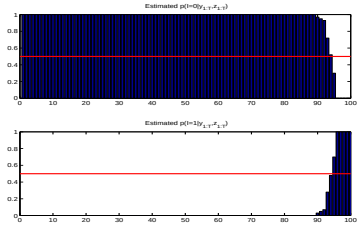
(c) Simulation Data with $\tau = 51$



(d) Change point estimation

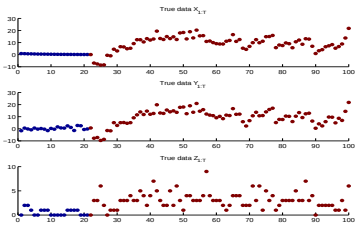


(e) Simulation Data with $\tau = 91$

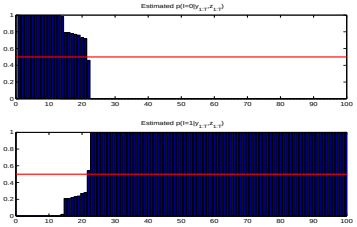


(f) Change point estimation

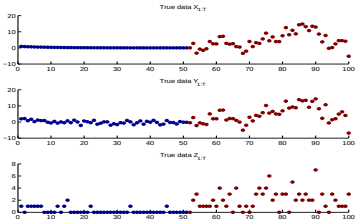
Figure 3.1: One change-point detection for Gaussian-Bernoulli (S1).



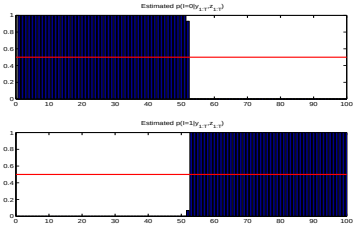
(a) Simulation data with $\tau = 21$



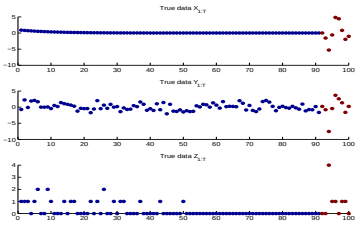
(b) Change-point estimation



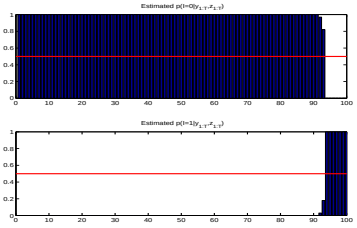
(c) Simulation data with $\tau = 51$



(d) Change-point estimation

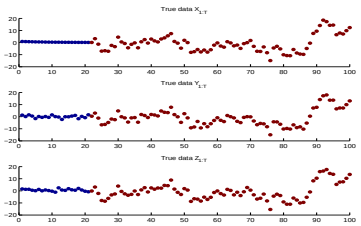


(e) Simulation data with $\tau = 91$

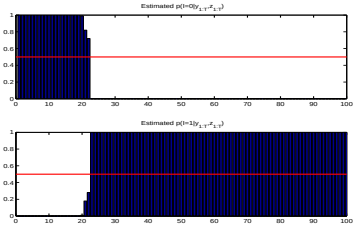


(f) Chang-point estimation

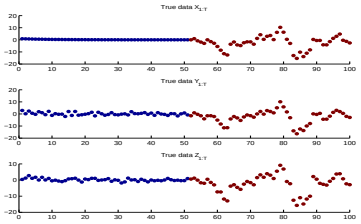
Figure 3.2: One change-point detection for Gaussian-Poisson (S2).



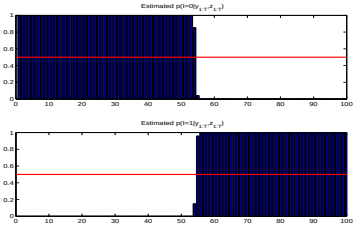
(a) Simulation data with $\tau = 21$



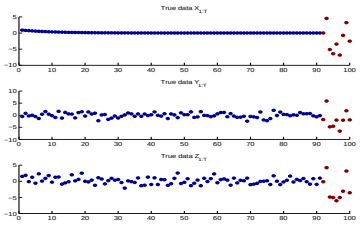
(b) Change-point estimation



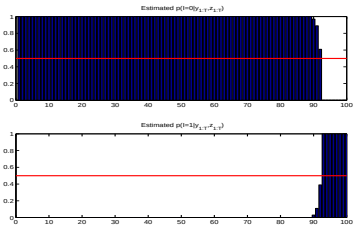
(c) Simulation data with $\tau = 51$



(d) Change-point estimation

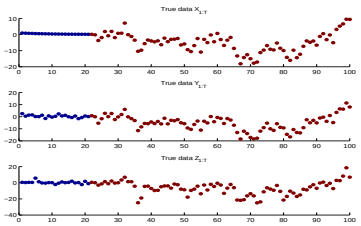


(e) Simulation data with $\tau = 91$

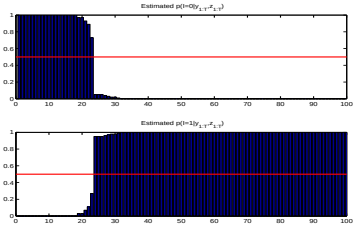


(f) Change-point estimation

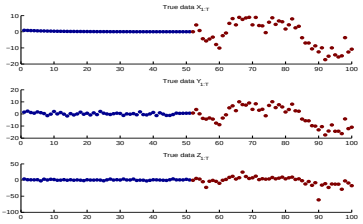
Figure 3.3: One change-point detection for Gaussian-Gaussian (S3).



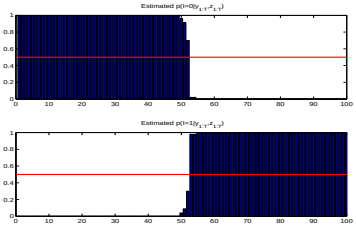
(a) Simulation data with $\tau = 21$



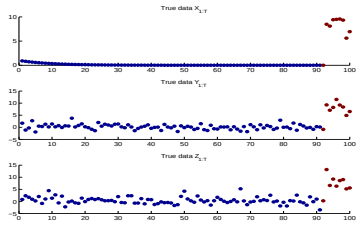
(b) Change point estimation



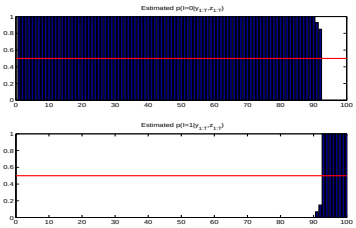
(c) Simulation data with $\tau = 51$



(d) Change point estimation



(e) Simulation data with $\tau = 91$



(f) Change point estimation

Figure 3.4: One change-point detection for Gaussian-Noncentral t (S4).

3.4.3 Results for Multiple Change-point Detection

Under the multiple change-point scenario, the key difference from the one change-point detection is the structure of the transition matrix P_I. Now the state parameter I_n can change from state 0 to 1 and then back to 0. For simplicity, let $p_1 = p_2$, and then the transition matrix can be specified as,

$$P_I = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}. \tag{3.32}$$

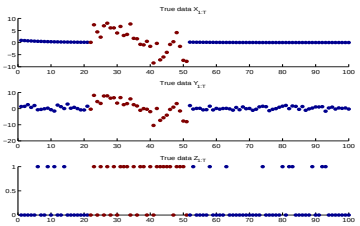
The unknown parameters are still the same, $\boldsymbol{\theta} = (\phi, \sigma, \gamma, p)$. I consider the two change-points and three change-points scenarios here. Three sets of true change-points are selected under each scenario. If there are two change-points, the combinations of $21, 51, 91$ are set as the true change-points. If there are three change-points, $[21, 51, 91]$ is chosen as one scenario, and two additional sets of change-points are simulated randomly under each case. All the true change-points and estimation results based on 50 iterations are shown in Table 3.2 and Table 3.3 below. The simulation data and the change-point estimation details are also shown in Figure 3.5 and Figure 3.6 for two and three change-points respectively. In general, estimated change-points are all close to the truth with only one time delay. The estimations of static parameters are almost the same as the true values. Despite a relatively longer computation time, models for mixed-type observations are able to obtain similar estimation accuracy (S1, S2) compared with those models for observations that are all continuous (S3, S4). Compared with Table 3.1, the overall performance for multiple change-points detection is more stable than that for one change-point. One possible explanation is that there are enough observations under each state, thus leading to relatively more accurate estimation for all unknown parameters.

Table 3.2: Simulation results for two change-points scenario

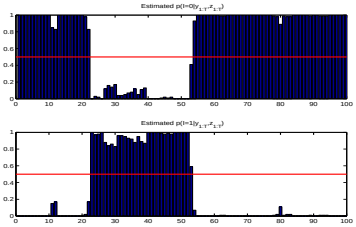
	S1			S2			S3			S4		
τ	[21 51]	[51 91]	[21 91]	[21 51]	[51 91]	[21 91]	[21 51]	[51 91]	[21 91]	[21 51]	[51 91]	[21 91]
$\hat{\tau}$	[22 52]	[52 92]	[22 92]	[22 52]	[52 92]	[22 92]	[22 52]	[52 92]	[22 92]	[22 52]	[52 92]	[22 92]
$sd(\hat{\tau})$	[0.73 1.02]	[0.95 0.53]	[0.92 0.96]	[0.35 1.04]	[0.86 0.72]	[0.80 1.67]	[0.11 0.37]	[0.16 1.38]	[1.16 1.22]	[0.11 0.37]	[0.11 0.37]	[1.14 1.20]
$\hat{\phi}$	0.89	0.89	0.90	0.90	0.89	0.90	0.90	0.90	0.90	0.90	0.89	0.90
$\hat{\sigma}$	3.92	4.08	4.00	4.14	3.92	4.20	3.96	4.03	4.00	4.14	4.08	4.28
$\hat{\gamma}$	1.03	1.07	1.00	0.99	1.03	1.03	1.04	1.00	0.94	1.06	1.09	1.46
acceptance rate	0.20	0.19	0.20	0.22	0.33	0.23	0.26	0.28	0.28	0.31	0.35	0.34
computation time(mins)	42.52	41.07	43.55	39.01	44.37	38.04	17.95	19.34	18.28	97.02	22.82	88.51

Table 3.3: Simulation results for three change-points scenario

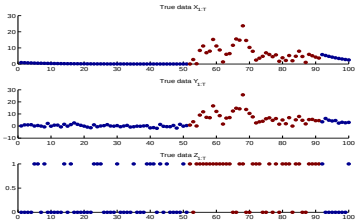
	S1			S2			S3			S4		
τ	[21 51 91]	[17 32 77]	[21 52 72]	[21 51 91]	[14 51 90]	[37 63 88]	[21 51 91]	[12 22 75]	[42 61 94]	[21 51 91]	[7 37 64]	[41 76 85]
$\hat{\tau}$	[22 52 91]	[18 33 78]	[22 53 72]	[22 52 92]	[15 52 91]	[38 64 89]	[22 52 92]	13 23 76]	[43 62 95]	[22 52 92]	[8 38 66]	[42 77 86]
$sd(\hat{\tau})$	[0.73 1.02 1.02]	[0.50 1.02 1.02]	[0.58 1.02 1.02]	[0.34 0.92 0.14]	[0.59 0.93 0.53]	[0.76 0.93 0.36]	[0.55 1.44 0.21]	0.24 2.10 0.85]	[0.12 0.35 0.14]	[0.60 1.60 0.24]	[0.71 1.50 1.59]	[0.94 1.53 1.57]
$\hat{\phi}$	0.87	0.88	0.88	0.89	0.90	0.89	0.90	0.89	0.90	0.90	0.90	0.89
$\hat{\sigma}$	3.91	3.88	4.04	4.09	4.08	4.16	4.18	4.31	4.19	4.24	4.27	4.10
$\hat{\gamma}$	1.07	0.92	1.11	0.99	1.03	0.95	1.03	0.96	0.95	1.12	1.33	0.95
acceptance rate	0.29	0.30	0.30	0.20	0.20	0.20	0.26	0.27	0.27	0.31	0.33	0.34
computation time(mins)	48.81	40.23	44.51	52.36	42.99	42.34	19.02	18.21	18.16	24.50	26.88	26.86



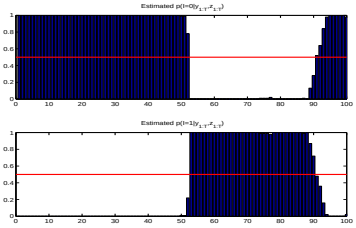
(a) Simulation Data with $\tau = 21, 51$



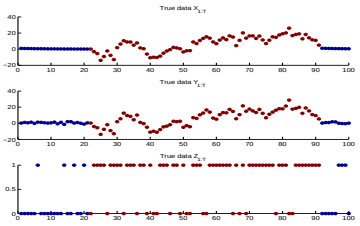
(b) Change point estimation



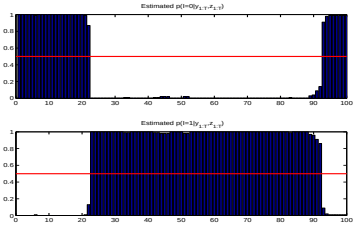
(c) Simulation Data with $\tau = 51, 91$



(d) Change point estimation

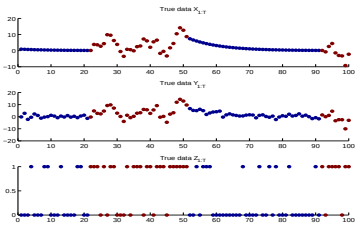


(e) Simulation Data with $\tau = 21, 91$

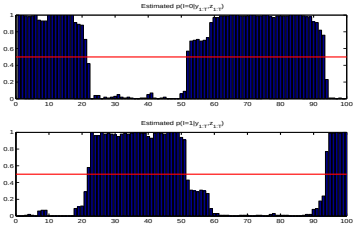


(f) Change point estimation

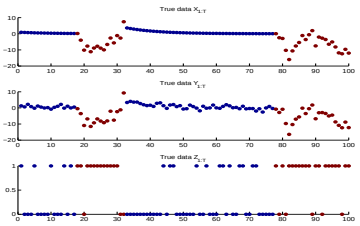
Figure 3.5: Two change-points detection for Gaussian-Bernoulli (S1).



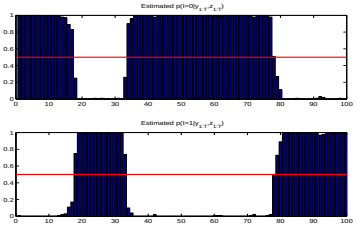
(a) Simulation Data with $\tau = 21, 51, 91$



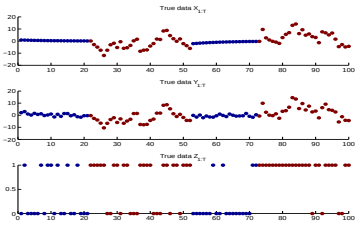
(b) Change-points estimation



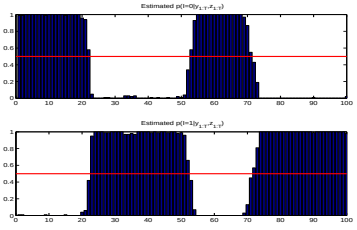
(c) Simulation Data with $\tau = 17, 32, 77$



(d) Change-points estimation



(e) Simulation Data with $\tau = 21, 52, 72$



(f) Change-points estimation

Figure 3.6: Three change-points detection for Gaussian-Bernoulli (S1).

3.5 Applications-Real civil unrest data

In this section, the proposed method is applied for change-point detection of the civil unrest data. Political events, including threatens, protests, event fights are happening every minute in the world, especially in Middle East or Latin American countries. Millions of life could be saved if people are aware of these crises. Two open resources, ICEWS (Integrated Conflict Early Warning System) and GDELT (Global Database of Events, Language, and Tone) are prominent systems for event coding and significant work has been built upon them to develop predictive systems.

GDELT is a global database of events which has been coded from vast quantities of publicly available text that is produced by the world's new media. It has created a great deal of excitement in the social science community, especially within the field of international relations. While, ICEWS is an early warning system designed to help US policy analysts predict a variety of international crisis to which the US might have to respond. These include international and domestic crisis, ethnic and religious violence, as well as rebellion and insurgency. GDELT and ICEWS are based on similar, though not identical methods and sources. GDELT includes data from 1979 to the present, while ICEWS program was launched in 2008. The data files use Conflict and Mediation Event Observations (CAMEO) coding for recording events. There are 20 event types in total, and it has an ordinal increase in cooperation as one goes from category 01 to 09, and an ordinal increase in conflict as one goes from 10 to 20. In this chapter, I will mainly focus on the protest (the 14th event type) in three Latin America countries, Argentina, Brazil, and Venezuela, since instabilities constantly occur in those Latin American countries.

The observations used for this modeling is a weekly binary data indicating whether there are protests occurred during a week, and a weekly continuous data measuring the AverageTone of all protests in each week. The value of AverageTone is the average tone of all documents containing one or more mentions of one event. The value of AverageTone commonly ranges from -10 and $+10$, with 0 indicating neutral. These two observations are highly related in

such a way that if a protest has an extremely negative average tone, it suggests a far more serious occurrence, which will usually spread and cause a series of new riots.

The goal is to detect those change times (in weeks) when more frequent or serious protests are going to happen. The time period of the data used is from January 2011 to April 2014. Within this time frame, there is one outbreak of protests occurred in each of the three countries recorded in Wikipedia. These protests are: (1) the protest during September 2012, at Cacerolazo in Argentina, (2) the 2013 Brazil protest happened during April and July 2013, (3) the protest in Venezuela started on February 12, 2014 up to present. For more details about these protests, please refer to Wikipedia (2015), Wikipedia (2016a), and Wikipedia (2016b).

By applying the proposed method using the Gaussian-Bernoulli setting with one change-point defined in subsection 3.4.1, the change-point is successfully detected. The observations for each country and their change-point detection plots are shown in Figure 3.7. The change-point indicates the week after which those three protests would happen. The results are as follows, (1) for Argentina, the estimated change-point is around week 08/26/2012, which is just one week before the outbreak recorded in Wikipedia (2015). (2) for Brazil, the estimated week is 02/24/2013, and it is one month before the so-called Brazilian Spring movement (Wikipedia, 2016a). (3) for Venezuela, the estimated week is around 01/26/2014, which is just one week before the February protest reported in Wikipedia (2016b).

These results show that the proposed model and algorithm can detect changes in social events in a timely manner (in the case of Argentina and Venezuela). Moreover, it also has the ability to provide early warnings. The model detects the outbreak one month earlier in Brazil.

3.5.1 Comparison

Since the proposed method can make full use of both the discrete and continuous information among the mixed observations, it has the ability to detect changes in an efficient manner. To show this advantage, I compare the proposed method with the conventional SSSM with the continuous observation or the binary observation separately for the same protest data.

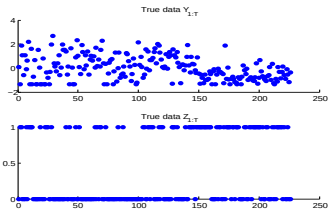
The results indicate that only using the continuous observations can not detect any changes at all for three countries. While, if only binary observations are used, changes can still be detected for Argentina and Venezuela. However, for Brazil, it fails to detect any changes. For Argentina and Venezuela, the binary data might include enough information to detect the changes. But for Brazil, neither the continuous nor the binary observation can provide useful information about the big outbreak of protests. Thus, by combing these two types of information together, the proposed method can make a better estimation of the change(s).

3.6 Conclusions and Discussion

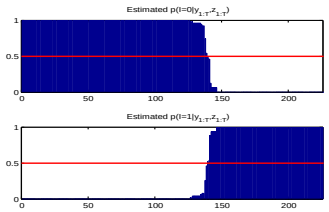
The mixed-type data problem attracts broad interests as its increasing popularity in modern society. However, dealing with them is quite challenging due to the difficulty of quantifying the correlations between continuous and discrete variables in an appropriate manner. The proposed method quantifies the mixed-type data by the latent processes in state-space models. And an indicator variable is used to detect possible changes in the entire process. Bayesian estimation and inference are conducted efficiently by the proposed combined DPF & SMC algorithm.

Various models are considered in this chapter, including mixed Gaussian-Bernoulli and mixed Gaussian-Poisson under different change-point scenarios. Both numerical examples and real case studies are analyzed to elaborate the performance of the proposed method in terms of estimation accuracy of parameter and change-points. Moreover, the proposed method can also be extended to the analysis of high-dimensional data. One may need latent processes in the SSSM to quantify the associations in the high-dimensional data. Inference methods will be very similar to the current approach in this chapter.

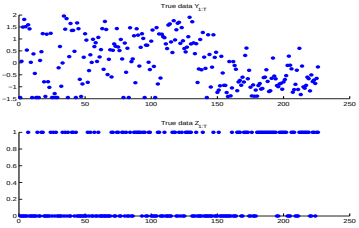
Note that the current method is an offline detection approach. Online change-point detection may be preferable in some areas, such as manufacturing industry. The current method has distribution assumptions for Gaussian and Bernoulli or Poisson. Further researches will be investigated to check the robustness of the proposed method.



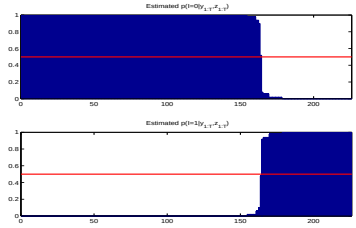
(a) Observations of Argentina



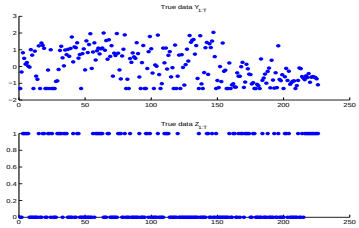
(b) Change-point estimation



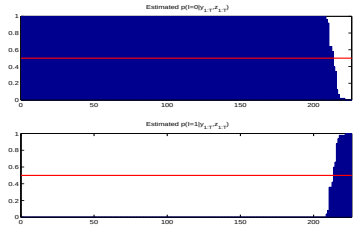
(c) Observations of Brazil



(d) Change-point estimation



(e) Observations of Venezuela



(f) Change-point estimation

Figure 3.7: Protests detection for Argentina, Brazil, and Venezuela.

If it is known that there is only one change-point in the data, the transition matrix P_I is restricted in Equation (3.30), such that the only one estimated change-point is guaranteed. However, if there are more than one change-point, the construction in Equation (3.32) may not give the exactly same number of change-points. In this situation, if we know the true number of change-points K, one can perform K-means clustering on the estimated change-points. However, in practice, the value of K is usually unknown. Suggestions from domain experts provide a good estimation of K. Alternatively, other clustering methods such as Hierarchical clustering can be applied.

Chapter 4 Self-segmented Classification via Adaptive Network LASSO

4.1 Introduction

In modern business, one major goal is to find an optimal pricing strategy to improve a company's revenue and profit. Therefore, it is essential to accurately predict the win probability of each Request-For-Quote(RFQ) from a client. The win probability is actually the likelihood that a prospective buyer will make a purchase after the quote. In practice, a seller provides a variety of products for which customers can construct a personalized bundle (combination of products) and submit a RFQ to the seller. For each RFQ, the seller has to determine an optimal price such that it increases the purchase probability of the client as well as the corresponding revenue.

To accurately estimate the purchase likelihood, one big challenge is the unlimited possible configurations of the bundle and possible correlations among each bundle. Xue et al. (2015) proposed a novel top-down and bottom-up approach to first decompose the bundle and then aggregate back to define important features of the bundle. While the widely used modeling method are logistic regressions, another challenge is that one global logistic regression model over all historical data may no longer be appropriate. Products and customers are heterogeneous in various ways. For example, different types or brand of products (hardware or software), changing price elasticity of demand, different types of customers (company or government) will lead to very different purchase behaviors. Thus, Xue et al. (2015) proposed to first segment incoming RFQs by their constructed bundle features. Next they fit logistic regression models within each segmentation independently based on historical data.

However, one major disadvantage of the approach proposed by Xue et al. (2015) is that the segmentation of RFQs and logistic regression fitting are conducted separately in two steps. It is possible that the RFQs clustered in the same segment do not share a common logistic model, which can result in inefficient and inaccurate model estimation as well as probability

prediction.

To ensure that data clusters are identified based on the similarities of their model structures, one natural idea is to conduct data segmentation and model fitting simultaneously. In the literature, one growing interest to identify homogeneous subpopulations within the larger heterogeneous population is the mixture modeling (Jung and Wickrama, 2008; Muthen, 2001). While in network graph point of view, combing data segmentation and modeling is related to the probabilistic graphical models (PGMs) (Nylund et al., 2007; Meila and Jordan, 2000). In most of those methods, the clustering membership is modeled by a latent variable. Then conditioning on the latent variable, a certain model structure is learned for each cluster. However, one commonly encountered issue is how to correctly determine the number of clusters, which is often addressed by using some criteria, such as, BIC (Bayesian information criteria), AIC (Akaike's Information Criterion), likelihood ratio tests, etc.. However, the class number suggested by those criteria can differ with each other as well as in the types of models considered. The other well-known problem related to those approaches is coming from the maximization algorithm. The likelihood based maximization often has the difficulties of converging to the global solutions especially when the dataset is large.

In order to automatically determine the segments based on modeling behavior, Hallac et al. (2015) proposed a network lasso approach that allows for simultaneous clustering and optimization on large graphs. The network lasso algorithm can be considered as a generalization of fused lasso (Tibshirani et al., 2005), group lasso (Yuan and Lin, 2006), and total variation methods (Wahlberg et al., 2012; Weinberger et al., 2007; Yang et al., 2013). By imposing a network lasso penalty on a set of model coefficient vectors, data sharing the same coefficient values are clustered into the same segment automatically. However, the lasso based penalty tends to inappropriately shrink model coefficient estimates to be relatively small. Thus it leads to biased model estimation and could be suboptimal in terms of estimation risk (Zou, 2006). Meinshausen and Bühlmann (2006) also showed the inconsistence of optimal prediction and estimation of the true model in the lasso problem.

In this chapter, I create an adaptive network lasso aiming to take care of the shrinkage

problem. The proposed method ensures that the estimated models under different segments are distinctive and data points with a common model structure are grouped into the same segment. The key idea of the proposed method is to impose an adaptive network lasso penalty on a set of model coefficient vectors, encouraging their homogeneity within each segment and heterogeneity across different segments. The adaptive penalty weight aims to alleviate the shrinkage problems of model coefficients based on how likely the corresponding data belong to the same segment. Besides, a convex optimization algorithm based on the Alternating Direction Method of Multipliers (ADMM) was developed in Hallac et al. (2015) to solve the optimization problem efficiently. Even though the ADMM based algorithm shows superior convex optimization performance for linear objective function, it is easy to stick on local optimum especially when the objective function is nonlinear. Therefore, in this chapter, an iteratively weighted least squares (IWLS) based algorithm is developed to achieve faster convergence rate in parameter estimation by linearizing the nonlinear objective functions.

The rest of this chapter is organized as follows. Section 4.2 gives a brief review of the network lasso model. Section 4.3 discusses the shrinkage problems in network lasso and introduces the proposed adaptive network lasso model. I also re-interpret the proposed approach from the Bayesian perspective. Section 4.4 provides a brief review of ADMM algorithm followed by the proposed optimization algorithm, IWLS-based algorithm. Section 4.5 shows several numerical examples to demonstrate the performance of the proposed schemes. And in Section 4.6, the proposed approach is applied to the IBM pricing data, which produces better predictions for the purchase probability. In the end, a conclusion is presented together with some discussions on future research directions in Section 5.6

4.2 Review of Network Lasso Model

Under the network setting, consider a graph $\mathcal{G}$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Denote $N = |\mathcal{V}|$ as the total number of nodes and $E = |\mathcal{E}|$ as the total number of edges. For example, Figure 4.1 shows an example of a complete network with $N = 6$ and $E = 15$, where all nodes

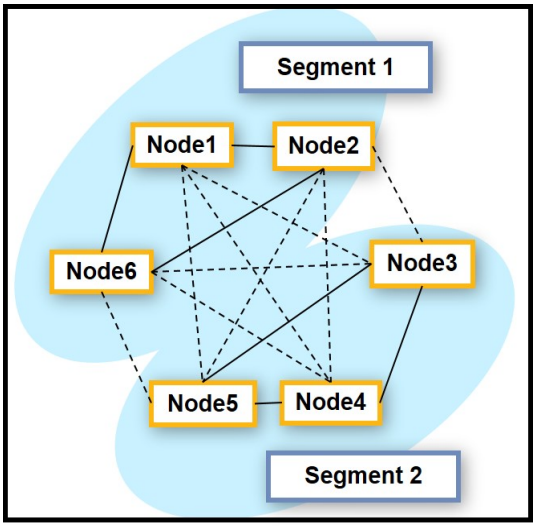


Figure 4.1: Network Representation.

are connected to each other. Assume the true number of segments is two. Let nodes $\{1, 2, 6\}$ belong to segment 1, then the rest nodes are in segment 2. We use solid edges to represent "correct" edges that connect nodes within the same segment, while dashed edges are "wrong" edges connecting nodes across different segments. The goal in network lasso problem is to correctly discover the true segments depending on their modeling behaviors. Note that each node in the network should have at least one edge in order to be segmented.

Thus the optimization problem we are interested in is as follows,

$$\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{\beta}_i; y_i, \boldsymbol{x}_i) + \lambda \sum_{(j,k) \in \mathcal{E}} w_{jk} ||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2, \tag{4.1}$$

where $\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N \in \mathbb{R}^p$ are independent variables related to each node, and $y_1, \cdots, y_N$ are the corresponding responses. $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_N \in \mathbb{R}^{p+1}$ are unknown coefficient vectors at each node. Let f_i denote the cost function at node i, which is restricted as convex in this chapter. In addition, $g_{jk}(\boldsymbol{x}_j, \boldsymbol{x}_k) = \lambda w_{jk} ||\boldsymbol{x}_j - \boldsymbol{x}_k||_2$ denotes the cost function associated with edge (j, k), where $\lambda \geqslant 0$ is an overall regularization parameter and $w_{jk} \geqslant 0$ are user-defined weights.

In this chapter, I specially consider the logistic regression scenario where the response

$y_i \in \{-1, 1\}$. Thus the objective loss function f_i can be the negative log likelihood as follows,

$$f_i = \log(1 + \exp(-y_i p_i)), \tag{4.2}$$

where, $p_i = \frac{\exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_i)}{1+\exp(\boldsymbol{x}_i^T \boldsymbol{\beta}_i)}$. Thus the first part of the network lasso model in Equation 4.1 becomes,

$$\sum_{i \in \mathcal{V}} f_i(\boldsymbol{\beta}_i; y_i, \boldsymbol{x}_i) = \sum_{i \in \mathcal{V}} \log(1 + \exp(-y_i p_i)),$$

which is the negative log likelihood function or the logistic loss measuring the discrepancy between the data and the logistic model.

The second part in Equation 4.1 is a penalty term for the logistic coefficients $(\boldsymbol{\beta}_j, \boldsymbol{\beta}_k)$ across edge (j, k). This penalty term aims to encourage those data sharing similar model coefficients to be exactly the same. Then segments can be automatically formed according to the coefficient values related to each node. In the penalty term, λw_{jk} is the edge weight between node j and node k. The overall regularization parameter λ controls the amount of penalization, and remains the same across all edges, while w_{jk} is a pre-defined weight allowing heterogeneity among different edges. The value of w_{jk} is determined by how "close/similar" nodes j and k are, which can be obtained from a prior knowledge of how likely they will belong to the same segment. It is often defined to be inversely proportional to a certain norm (eg. euclidean distance) of several features associated with nodes j and k. For example, if the domain knowledge indicates that the first two independent variables $\boldsymbol{x}_1, \boldsymbol{x}_2$ can potentially influence the segmentation of the data. Then $\boldsymbol{x}_1, \boldsymbol{x}_2$ can be used to measure the distance between two nodes as well as the edge weights. Two nodes that are close will be connected by an edge in the network construction and the corresponding weight formula is shown as follows,

$$w_{jk} = \frac{1}{||\tilde{\boldsymbol{x}}_j - \tilde{\boldsymbol{x}}_k||_2}, \tag{4.3}$$

where, $\tilde{\boldsymbol{x}}_j = [x_{j1}, x_{j2}]^T$. That is, we assume data having similar values of $\boldsymbol{x}_1, \boldsymbol{x}_2$ tend to belong to the same segment, and thus more weights is put on the penalty in order to push the corresponding coefficient values to be the same. While, on the other hand, data with very different values of $\boldsymbol{x}_1, \boldsymbol{x}_2$ may be less likely to share the same model coefficients. Then, the weight value will be relatively small in this situation.

4.3 The Proposed Adaptive Network Lasso

4.3.1 The Shrinkage Problem in Network Lasso

The network lasso uses a global shrinkage parameter λ to control the overall penalization and the pre-defined w_{jk} on each edge is fixed throughout the optimization. One disadvantage of using fixed penalty weights is that it leads to inappropriate shrinkage problem of the coefficient estimates, especially when two connected nodes j, k are actually coming from different segments.

During the network construction procedure, two nodes j, k are connected based on prior knowledge or the similarity measure between $\tilde{\boldsymbol{x}}_j$ and $\tilde{\boldsymbol{x}}_k$, regardless of whether they truly belong to the same segment or not. Thus "wrong" edges are inevitably created, which corresponds to the dashed edges in Figure 4.1. If the underlying truth is that nodes j, k are in the same segment, then λw_{jk} is correctly pushing $||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2$ to be zero. However, if nodes j, k don't belong to the same segment, then $||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2$ is always positive. Therefore, the absolute values of $\boldsymbol{\beta}_j$ and $\boldsymbol{\beta}_k$ are seriously shrunk in order to minimize the penalty term. This inappropriate lasso shrinkage leads to biased coefficient estimates for connected nodes that are in fact from different segments and thus it could be suboptimal in terms of estimation risk (Zou, 2006).

Regarding the lasso shrinkage problem, Meinshausen and Bühlmann (2006) also showed the disagreement of optimal prediction and the accurate estimation of the true model in the lasso. They proved that the optimal λ for prediction gives inconsistent variable selection results.

Since both data segmentation and model prediction are highly influenced by the estimation accuracy of model coefficients, the estimation performance is essential in the purchase likelihood prediction problem. To alleviate the shrinkage problem, an adaptive network lasso model will be illustrated in section 4.3.2, where an adaptive shrinkage parameter λ_{jk} is introduced to penalize coefficient differences $||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2$.

4.3.2 The Adaptive Network Lasso

As discussed in the previous section, the shrinkage problem in network lasso mainly comes from the fixed penalization of all coefficient differences. Similar to the proposed adaptive lasso in Zou (2006), a simple and effective remedy is to assign different weights to different edges (coefficient differences). The proposed adaptive network lasso is shown in Equation 4.4, and can be considered as a generalization of network lasso. Particularly, the regularization parameter λ_{jk} can take different values on different edges, which is adaptive. Then, the major objective is to cleverly choose the adaptive λ_{jk} values depending on how likely nodes j, k are in the same segment. Thus the optimization problem becomes,

$$\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{\beta}_i; y_i, \boldsymbol{x}_i) + \sum_{(j,k) \in \mathcal{E}} \lambda_{jk} w_{jk} ||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2, \tag{4.4}$$

where, $\lambda_{jk} \propto \frac{1}{||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2}$.

In addition, it is equivalent to using one global λ with adaptive penalty weight,

$$\text{minimize} \quad \sum_{i \in \mathcal{V}} f_i(\boldsymbol{\beta}_i; y_i, \boldsymbol{x}_i) + \lambda \sum_{(j,k) \in \mathcal{E}} \tilde{w}_{jk} ||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2, \tag{4.5}$$

where, $\tilde{w}_{jk} \propto \frac{w_{jk}}{||\boldsymbol{\beta}_j - \boldsymbol{\beta}_k||_2}$.

Next, we justify our proposed adaptive network lasso model from the Bayesian viewpoint. Since the construction of adaptive weights is equivalent to assigning an appropriate prior to coefficients $\boldsymbol{\beta}$, let us denote the prior density function as $p(\boldsymbol{\beta}|\boldsymbol{\theta})$. Then the posterior density

conditioning on observations $\boldsymbol{y}, \boldsymbol{X}$ and $\boldsymbol{\theta}$ is,

$$p(\boldsymbol{\beta}|\boldsymbol{y}, \boldsymbol{X}, \boldsymbol{\theta}) \propto f(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\theta}),$$

where, $\boldsymbol{y} \in \mathbb{R}^N$ is the response variable, $\boldsymbol{X}$ is a N by p input matrix, $\boldsymbol{\beta}$ is a p by 1 unknown vector, and $\boldsymbol{\theta} \in \Theta$ contains all unknown hierarchical parameters. Then the goal is to find the MAP estimates of $\boldsymbol{\beta}$ by solving,

$$\hat{\boldsymbol{\beta}}_{MAP} = \underset{\boldsymbol{\beta}}{\operatorname{argmax}}\, f(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta}) p(\boldsymbol{\beta}|\boldsymbol{\theta}),$$

or, equivalently,

$$\hat{\boldsymbol{\beta}}_{MAP} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} -\log f(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta}) - \log p(\boldsymbol{\beta}|\boldsymbol{\theta}).$$

The $\log p(\boldsymbol{\beta}|\boldsymbol{\theta})$ can be thought of as a penalization term in optimizing the log-likelihood of the data $\log f(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta})$ (Lee et al., 2010).

Denote $\boldsymbol{\beta}_e = \boldsymbol{\beta}_j - \boldsymbol{\beta}_k$, $e = 1, \cdots, E, (j,k) \in \mathcal{E}$ as the difference of coefficients on each edge. According to Lee et al. (2010) and Jiang et al. (2012), the hierarchical priors given to the coefficient difference on each edge $\boldsymbol{\beta}_e$ are listed below,

$$\begin{aligned} &\beta_{e,i}|\sigma_e^2 \sim N(0, \sigma_e^2), \quad i = 1, 2, \cdots, p, \\ &\sigma_e^2|\tau_e \sim G(\frac{p+1}{2}, 2\tau_e^2), \\ &\tau_e|a_e, b_e \sim IG(a_e, b_e), \end{aligned} \tag{4.6}$$

where, $G(a,b)$ denotes the Gamma distribution with density function $f(x) = x^{a-1} b^{-a} \Gamma(a)^{-1} \exp\left(-\frac{x}{b}\right)$, and $IG(a,b)$ represents the Inverse Gamma distribution with the density form, $f(x) = \frac{b^a}{\Gamma(a)} x^{-a-1} \exp\left(-\frac{b}{x}\right)$.

For each edge,we have,

$$p(\beta_{e,1}, \cdots, \beta_{e,p}|\tau_e) = \frac{(2\tau_e)^{-p} \pi^{-(p-1)/2} \Gamma(a_e + p)}{\Gamma((p+1)/2) \Gamma(a_e)} \exp(-\frac{\sqrt{\sum_{i=1}^p |\beta_{e,i}|^2}}{\tau_e}),$$

or,

$$= \frac{(2\tau_e)^{-p}\pi^{-(p-1)/2}\Gamma(a_e+p)}{\Gamma((p+1)/2)\Gamma(a_e)}\exp(-\frac{||\boldsymbol{\beta}_e||_2}{\tau_e}).$$

Besides,

$$\tau_e|\boldsymbol{\beta}_e, a_e, b_e \sim IG(a_e+p, b_e+||\boldsymbol{\beta}_e||_2).$$

Then the marginal prior on coefficients $\boldsymbol{\beta}_j, \boldsymbol{\beta}_k$ on each edge e is,

$$p(\boldsymbol{\beta}_j, \boldsymbol{\beta}_k|a_e, b_e) = \frac{(2b_e)^{-p}\pi^{-(p-1)/2}\Gamma(a_e+p)}{\Gamma((p+1)/2)\Gamma(a_e)}\left(\frac{||\boldsymbol{\beta}_j^{(t)}-\boldsymbol{\beta}_k^{(t)}||_2}{b_e}+1\right)^{(-a_e-p)}.$$

The prior distribution of all regression coefficients $\boldsymbol{\beta} = \left(\boldsymbol{\beta}_1^T, \cdots, \boldsymbol{\beta}_N^T\right)^T$, which is Np by 1 vector, is defined as follows,

$$\pi(\boldsymbol{\beta}|a_1, \cdots, a_E, b_1, \cdots, b_E) \propto \prod_{(i,k)\in\mathcal{E}} p(\boldsymbol{\beta}_j, \boldsymbol{\beta}_k|a_e, b_e),$$

where a_e, b_e for $e = 1, \cdots, E$ are the hierarchical parameters at each edge e.

Finally, the hierarchical representation for the prior can be expressed as,

$$\begin{aligned}
&\pi(\boldsymbol{\beta}|a_1, \cdots, a_E, b_1, \cdots, b_E)\\
&\propto \prod_{(i,j)\in\mathcal{E}, e=1,\cdots,E}\int_{\tau_e}\int_{\sigma_e^2}\frac{1}{\sqrt{2\pi\sigma_e^2}}\exp\left(-\frac{(\boldsymbol{\beta}_j-\boldsymbol{\beta}_k)^T(\boldsymbol{\beta}_j-\boldsymbol{\beta}_k)}{2\sigma_e^2}\right)\pi(\sigma_e^2)\pi(\tau_e)d\sigma_e^2 d\tau_e\\
&\propto \int\int\prod_{e=1}^{E}(\sigma_e^2)^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\boldsymbol{\beta}^{\boldsymbol{T}}\Sigma_{\boldsymbol{\beta}}^{-1}\boldsymbol{\beta}\right)\prod_{e=1}^{E}\pi(\sigma_e^2)\prod_{e=1}^{E}\pi(\tau_e)\prod_{e=1}^{E}d\sigma_e^2\prod_{e=1}^{E}d\tau_e,
\end{aligned}$$

where, $\Sigma_{\boldsymbol{\beta}}^{-1}$ is the $N(p+1)\times N(p+1)$ symmetric precision matrix with the following

structure,

$$\Sigma_{\boldsymbol{\beta}}^{-1} = \begin{bmatrix} \sum_{j\in\mathcal{N}(1)} \frac{1}{\sigma^2_{(1,j)}} & -\frac{1}{\sigma^2_{(1,2)}} & 0 & \dots & 0 \\ -\frac{1}{\sigma^2_{(2,1)}} & \sum_{j\in\mathcal{N}(2)} \frac{1}{\sigma^2_{(2,j)}} & 0 & \dots & -\frac{1}{\sigma^2_{(2,N)}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & -\frac{1}{\sigma^2_{(N,2)}} & -\frac{1}{\sigma^2_{(N,3)}} & \dots & \sum_{j\in\mathcal{N}(N)} \frac{1}{\sigma^2_{(N,j)}} \end{bmatrix} \otimes \mathbf{1}_{p+1},$$

where, $\mathcal{N}(i)$ denotes the neighbors of node i, the subscript (i,j) denotes edge (i,j), and $\sigma^2_{(i,j)} = \sigma^2_{(j,i)}$. The symbol $\otimes$ represents Kronecker product. The off-diagonal element $-\frac{1}{\sigma^2_{(i,j)}}$ is non-zero if and only if edge $(i,j) \in \mathcal{E}$ (i.e. nodes i and j are connected).

Under this scenario, the corresponding iterative procedure to solve for $\boldsymbol{\beta}$ is,

$$\boldsymbol{\beta}^{(t+1)} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} -\log f(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\theta}) + \sum_{e\in\mathcal{E}} w_e^{(t+1)} ||\boldsymbol{\beta}_e^{(t)}||_2,$$

where, $w_e^{(t+1)} = \frac{a_e+p}{||\boldsymbol{\beta}_e^{(t)}||_2+b_e}$, which is in consistent with the proposed adaptive weight format in Equation 4.5.

4.4 Computational Algorithm

For a relatively large graph problem where p, $N = |\mathcal{V}|$, $E = |\mathcal{E}|$ are potentially large, the general convex optimization methods can not work well. Thus, Hallac et al. (2015) developed an algorithm based on the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Parikh and Boyd, 2014) to solve the network lasso problem efficiently. However, since the ADMM-based algorithm tries to find the optimal solutions by going through each node and edge sequentially, the chance of sticking on local optimum is relatively high, especially when the objective function is nonlinear. To solve this problem, we develop an IWLS (Iteratively Weighted Least Square) based algorithm for adaptive network lasso model. The proposed

algorithm is especially efficient when the objective function is a logistic loss function. And the key idea is to linearize the nonlinear objective function. Moreover, by updating all the unknown parameters simultaneously, the proposed IWLS-based algorithm can converge to the global optimum efficiently and effectively.

In this section, the ADMM-based algorithm for network lasso is briefly introduced in section 4.4.1. Then our proposed IWLS-based algorithm will be illustrated in section 4.4.2.

4.4.1 Brief Description of ADMM-based Algorithm

ADMM is a well-established method for solving distributed convex optimization problems. To solve via ADMM, we introduce a copy of β_i, called z_{ij}, at each edge (i, j). Note that the same edge also has a z_{ji}, a copy of β_j. Let us rewrite the problem as an equivalent problem below,

$$\begin{aligned} \text{minimize} \quad & \sum_{i\in\mathcal{V}} f_i(x_i;\beta_i) + \lambda \sum_{(j,k)\in\mathcal{E}} w_{jk}||z_{jk} - z_{kj}||_2, \\ \text{subject to} \quad & \beta_i = z_{ij}, \quad i = 1,\dots,N, \quad j \in N(i), \end{aligned} \tag{4.7}$$

where $N(j)$ is the set of neighbors of node j. Deriving this problem's augmented Lagrangian (Hestenes, 1969), we get,

$$\begin{aligned} L_\rho(\beta, z, u) = &\sum_{i\in\mathcal{V}} f_i(\beta_i) + \sum_{(j,k)\in\mathcal{E}} (\lambda w_{jk}||z_{jk} - z_{kj}||_2 - (\rho/2)(||u_{jk}||_2^2 + ||u_{kj}||_2^2) \\ &+ (\rho/2)(||\beta_j - z_{jk} + u_{jk}||_2^2 + ||\beta_k - z_{kj} + u_{kj}||_2^2)), \end{aligned}$$

where u is the scaled dual variable at each edge and ρ is a scalar penalty parameter that determines the trade-off between primal and dual convergence. ADMM consists of the following steps, with t denoting the iteration number,

$$\beta^{t+1} = \underset{\beta}{\text{argmin}} \quad L_\rho(\beta, z^t, u^t),$$
$$z^{t+1} = \underset{z}{\text{argmin}} \quad L_\rho(\beta^{t+1}, z, u^t),$$
$$u^{t+2} = u^t + (\beta^{t+1} - z^{t+1}).$$

The details of these 3 steps are listed below.

$\beta-$**Update**.

In the $\beta-$update, we minimize a separable sum of functions, one per node, so it can be calculated independently at each node and solved in parallel. At node i, this is,

$$\beta_i^{t+1} = \underset{\beta_i}{\text{argmin}} \left(f_i(\beta_i) + \sum_{j \in N(i)} \rho/2 ||\beta_i - z_{ij}^t + u_{ij}^t||_2^2 \right).$$

$z-$**Update**.

The $z-$update is separable across the edges. Note that for edge ij, we need to jointly update z_{ij} and z_{ji}. This becomes,

$$z_{ij}^{t+1}, z_{ji}^{t+1} = \underset{z_{ij}, z_{ji}}{\text{argmin}} \left(\lambda w_{ij} ||z_{ij} - z_{ji}||_2 + (\rho/2) \left(||\beta_i^{t+1} - z_{ij} + u_{ij}^t||_2^2 + ||\beta_j^{t+1} - z_{ji} + u_{ji}^t||_2^2 \right) \right).$$

This problem has a closed-form analytical solution,

$$z_{ij}^* = \theta(\beta_i + u_{ij}) + (1 - \theta)(\beta_j + u_{ji}),$$
$$z_{ji}^* = (1 - \theta)(\beta_i + u_{ij}) + \theta(\beta_j + u_{ji}),$$

where,

$$\theta = \max \left(1 - \frac{\lambda w_{ij}}{\rho ||\beta_i + u_{ij} - (\beta_j + u_{ji})||_2}, \, 0.5 \right).$$

$u-$**Update**.

The $u-$update is also edge-separable. For each variable, it looks like,

$$u_{ij}^{t+1} = u_{ij}^{t} + (\beta_j^{t+1} - z_{ij}^{t+1}).$$

More detailed information about stopping criteria can be found in Boyd et al. (2011).

4.4.2 The Developed IWLS-based Algorithm

While the ADMM-based algorithm works perfectly when the objective function is linear, it may stick on local optimum when the objective function is non-linear. This issue will be further discussed in section 4.5. Thus, I propose an IWLS (Iteratively Weighted Least Square) based algorithm particularly designed for the logistic loss function under adaptive network lasso construction. The key idea is to linearize the logistic loss function. The detailed procedure is shown below. Assume each response Y_i follows a Bernoulli distribution with probability, $p_i = \frac{\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_i)}{1+\exp(\boldsymbol{x}_i^T\boldsymbol{\beta}_i)}$. The corresponding non-linear log likelihood function becomes,

$$\log(f(y_1, y_2, \cdots, y_N)) = \sum_{i\in\mathcal{V}} y_i \log(\frac{p_i}{1-p_i})) + \sum_{i\in\mathcal{V}} \log(1-p_i)). \tag{4.8}$$

Take the first derivative of Equation 4.8 to solve $\boldsymbol{\beta}_i$ using Newton's Method. Then, at iteration $t+1$ we can get,

$$\hat{\boldsymbol{\beta}}_i^{t+1} = \hat{\boldsymbol{\beta}}_i^{t} + (\boldsymbol{x}_i\hat{p}_i^t(1-\hat{p}_i^t)\boldsymbol{x}_i^T)^{-1}\boldsymbol{x}_i(y_i - \hat{p}_i^t),$$

where, $\hat{\boldsymbol{\beta}}_i^t$ and $\hat{p}_i^t$ are estimates from the previous iteration t, and $\hat{p}_i = \frac{\exp(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_i)}{1+\exp(\boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_i)}$.

This equation can also be arranged to,

$$\hat{\boldsymbol{\beta}}_i^{t+1} = (\boldsymbol{x}_i\hat{p}_i^t(1-\hat{p}_i^t)\boldsymbol{x}_i^T)^{-1}\boldsymbol{x}_i\hat{p}_i^t(1-\hat{p}_i^t)\hat{z}_i^t,$$

where $\hat{z}_i^t = \boldsymbol{x}_i^T\hat{\boldsymbol{\beta}}_i^t + \frac{y_i-\hat{p}_i^t}{\hat{p}_i^t(1-\hat{p}_i^t)}$.

Let weight $\hat{\pi}_i^t = \hat{p}_i^t(1-\hat{p}_i^t)$, then the estimate of $\boldsymbol{\beta}_i$ at each iteration is considered as the

weighted least square estimate for an adjusted observation z_i and independent variables $\boldsymbol{x}_i$,

$$\hat{\boldsymbol{\beta}}_i^{t+1} = (\boldsymbol{x}_i \hat{\pi}_i^t \boldsymbol{x}_i^T)^{-1} \boldsymbol{x}_i \hat{\pi}_i^t \hat{z}_i^t.$$

In adaptive network lasso problem, the corresponding convex objective function in Equation 4.5 becomes,

$$f_i^{t+1} = \hat{\pi}_i^t (\hat{z}_i^t - \boldsymbol{x}_i^T \boldsymbol{\beta}_i^{t+1})^2, \tag{4.9}$$

where $\hat{z}_i^t$ is the estimated value in previous iteration t.

By linearizing the non-linear function in Equation 4.8, the proposed algorithm updates unknown parameters $\boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_N$ simultaneously. Compared with the ADMM based algorithm which updates parameters edge by edge, the proposed algorithm produces better estimation performance and faster global convergence rate. Detailed comparisons will be shown in section 4.5.

Note that the objective function in the proposed adaptive network lasso would change with optimization iterations t due to the changes of adaptive weights $\tilde{w}_{jk}^t$. Particularly, the weights for zero-coefficient differences get inflated (to infinity), whereas the weights for nonzero-coefficient differences converge to a constant value. Thus the objective function keeps to be convex and the coefficient estimates will converge to the logistic regression estimates under each estimated segment.

4.5 Numerical Study

In this section, simulation studies are conducted to evaluate the performance of the proposed model and algorithm. The simulated data contains N nodes and each node i has its own independent variables and responses. Particularly, three continuous features, X_1, X_2, X_3 are simulated for each node. Besides, X_1 and X_2 are defined as weight features such that nodes with similar values of X_1 and X_2 are more likely to be in the same segment. Edges in the

network are constructed by connecting five nearest neighbors to each node (Hallac et al., 2015). The Euclidean distance calculated by (X_1, X_2) is used to defined the nearest neighbors. Thus, each node has at least five connected nodes. For simplicity, I only include X_3 in the logistic regression model in the simulation study. The responses are binary at each node, $y_i \in \{1, -1\}$. In the simulation study, three scenarios (listed below) are considered regarding the number of true clusters K and how close these clusters are (separated or adjacent).

(D1) **Separated Clusters with $K = 2$:** $N = 500$ observations of (X_1, X_2) are simulated shown in Figure 4.2a. The two clusters are clearly separated from each other.

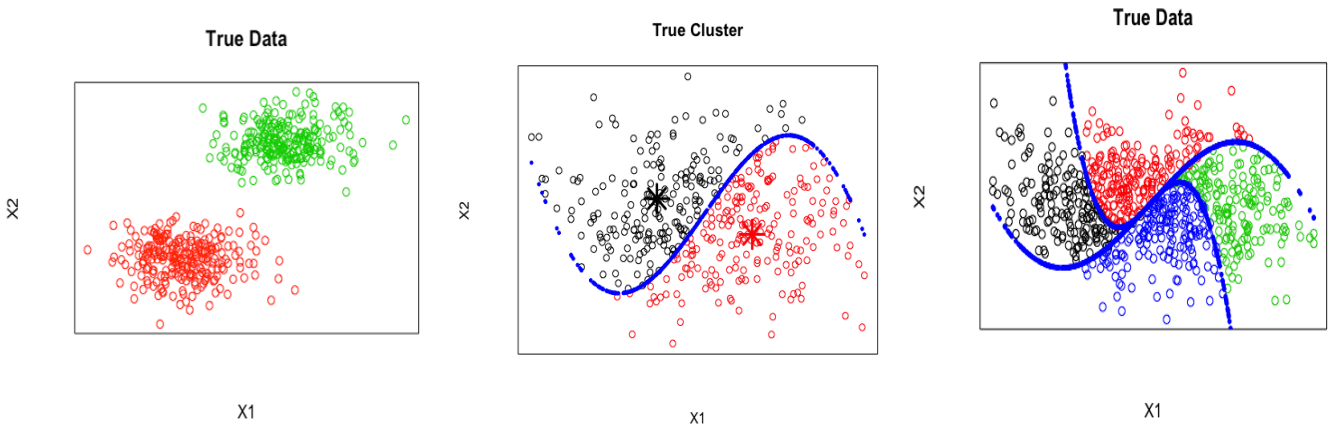
(D2) **Adjacent Clusters with $K = 2$:** $N = 500$ observations of (X_1, X_2) are simulated and the data is split by a nonlinear function of X_1 and X_2, which is the blue curve shown in Figure 4.2b.

(D3) **Adjacent Clusters with $K = 4$:** $N = 900$ observations of (X_1, X_2) are simulated shown in Figure 4.2c. The data is split by two nonlinear functions of X_1 and X_2 (two blue curves).

Note that all independent variables X_1, X_2, X_3 are simulated randomly from normal distributions with appropriate mean and variance values. Moreover, X_3 is simulated without any restriction. Next, a unique set of true coefficients is assigned to each segment/cluster obtained previously, $\boldsymbol{\beta_k} = [\beta_{k0}, \beta_{k1}]^T, k = 1, \cdots, K$. Finally, the responses $\boldsymbol{y}$ are simulated based on the independent variables X_3 as well as the corresponding true coefficients. Let us denote (b_0, b_1) as the estimated coefficients.

For each simulation, data is randomly split into training (80%) and testing (20%) data sets. Then the 5-fold cross validation is conducted for the training data in order to find the optimal λ value which maximizes the prediction accuracy based on AUC values. At the same time, the cutting point c in the logistic regression is chosen such that the sum of sensitivity and specificity is maximized.

In the simulation study, seven models/algorithms are considered as described below,



(a) Separated, $K = 2$ (b) Adjacent, $K = 2$ (c) Adjacent, $K = 4$

Figure 4.2: Simulation X_1, X_2.

(M1) **Optimal Model**: Fit logistic regression model under each true segment.

(M2) **Global Model**: Fit one logistic regression model.

(M3) **K-Means**: Cluster data by K-means according to X_1, X_2 and fit logistic regression under each cluster.

(M4) **ADMM**: Fit network Lasso model by ADMM algorithm.

(M5) **IWLS**: Fit network Lasso model by IWLS algorithm.

(M6) **Adaptive ADMM**: Fit adaptive network Lasso model by ADMM algorithm.

(M7) **Adaptive IWLS**: Fit adaptive network Lasso model by IWLS algorithm.

Note that M7 is the proposed approach.

4.5.1 Simulation Results for D1

Table 4.1 shows the estimation results under Separated Clusters with $K = 2$ (D1) scenario. The true coefficients under each segment are listed in column [1]. Columns [3] and [5] correspond to M4 and M5 respectively, while columns [2] and [4] refit logistic regression model under segments obtained from M4 and M5 respectively.

Since the simulated data is clearly separated by (X_1, X_2) as shown in Figure 4.2a, there are no shrinkage problems. Under this situation, both ADMM and IWLS algorithms work well in terms of coefficients estimation and data segmentation. The true clusters are accurately found in both methods, thus two GLM results are exactly the same. Besides, IWLS method gives exactly the same coefficient estimates compared with GLM. While ADMM do not find the global optimal. Moreover, IWLS takes less than half of the computation time to converge compared with ADMM. This result shows that compared with ADMM, the IWLS based algorithm is more efficient regarding estimation accuracy as well as convergence rate.

Table 4.1: Coefficients Estimation results for Separated Clusters with $K = 2$ (D1)

Segment		True	GLM(ADMM)	ADMM	GLM(IWLS)	IWLS
		[1]	[2]	[3]	[4]	[5]
1	b_0	-1	-1.047	-0.882	-1.047	-1.047
	b_1	2.5	2.385	2.01	2.385	2.384
2	b_0	1.5	1.751	1.146	1.751	1.746
	b_1	-3.5	-3.56	-2.213	-3.56	-3.55
Computation Time(hr)			-	2.52	-	1.17

4.5.2 Simulation Results for D2

Table 4.2 shows the estimation results under Adjacent Clusters scenario with $K = 2$ (D2). The segmentation and estimation are relatively difficult compared with D1, especially for the data near the boundary between two segments. The reason is that those data are considered as "close" neighbors in terms of (X_1, X_2) values. While they actually belong to different segments. "Wrong" edges are connected across two segments through which fixed penalty weights are imposed in network lasso. Thus it leads to inappropriate shrinkage of coefficient estimates as discussed in section 4.3.

Comparing with the true coefficient values, GLM fits in both columns [2] and [4] are not working well. GLM fits based on ADMM segments give totally wrong signs. It indicates

that the ADMM algorithm may not converge to the global optimum. Besides, comparing ADMM and IWLS with their corresponding GLM fits, both algorithms give smaller absolute coefficient values. This is the sign of shrinkage problems in network lasso model.

Table 4.2: Coefficients Estimation results for Adjacent Clusters with $K = 2$ (D2)

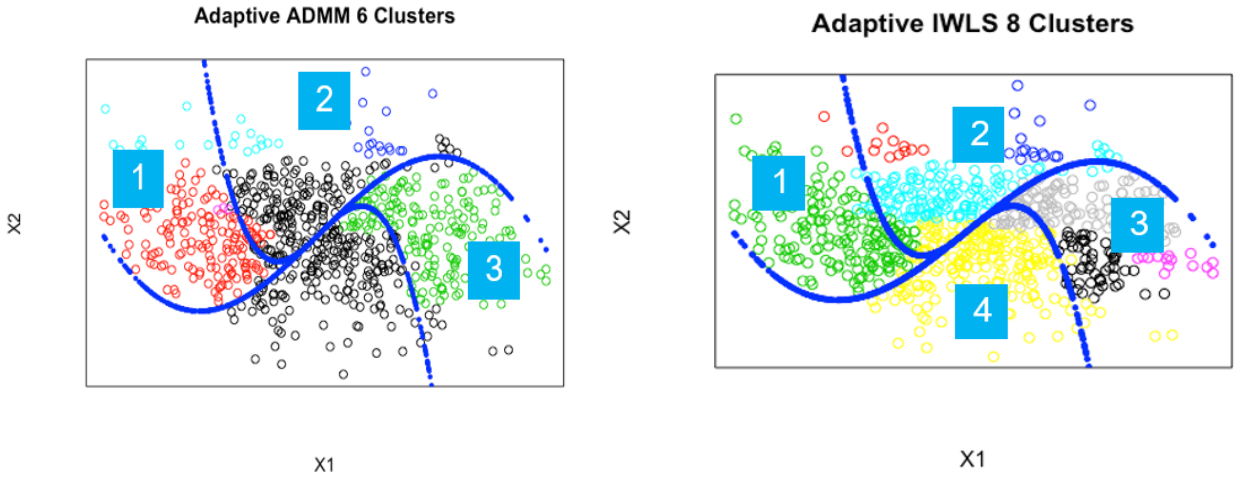
Segment		True	GLM(ADMM)	ADMM	GLM(IWLS)	IWLS
		[1]	[2]	[3]	[4]	[5]
1	b_0	-1	1.353	-0.51	-0.646	-0.601
	b_1	2.5	-33.405	1.229	1.901	1.414
2	b_0	1.5	-0.628	0.509	1.365	0.719
	b_1	-3.5	1.646	-1.26	-3.504	-1.798

The results shown in D1 and D2 conclude that the network lasso model with ADMM based algorithm encounters both shrinkage and convergence difficulties for logistic objective functions.

4.5.3 Simulation Results for D3

This section tries to show the advantages of the proposed approach (M7) using the Adjacent Clusters data with $K = 4$ (D3) corresponding to Figure 4.2c. Table 4.3 shows the true coefficients simulated within each segment. Figure 4.3 shows the segmentation results by fitting adaptive network lasso using ADMM (M6) and IWLS (M7) respectively. Figure 4.3a gives 3 major clusters while figure 4.3b produces 4 major clusters. The estimated coefficients corresponding to these major clusters are listed in Table 4.4 within columns [2] and [4]. The coefficients fitted by GLM within each major clusters are shown in columns [1] and [3]. The overall estimation performance using IWLS is superior compared with ADMM, indicating a good convergence property toward the truth. Besides, comparing both algorithms with their GLM fits, it is clear that the ADMM shows much more serious shrinkage problem than IWLS.

Furthermore, in order to show the overall prediction performance of M7, Table 4.5 compares all the 7 models/algorithms in terms of six criteria calculated for testing data. The criteria are



(a) Adaptive ADMM

(b) Adaptive IWLS

Figure 4.3: Results for D3.

listed in columns [1]-[6]. Particularly, the Frobenius norm, Fnorm $= \sqrt{\sum_{i=1}^{n}\sum_{j=0}^{p}(b_{ij}-\beta_{ij})^2}$, measures the estimation accuracy of coefficients, where n is the number of testing observations. And the AUC is the area under the Receiver Operating Characteristic (ROC) Curve.

The optimal model (M1) in the first row corresponds to the best results obtained when the segmentation is exactly the same as truth. Since it is impossible to achieve in reality, it is listed here as a reference. The result shows that the global model (M2) gives the worst performance with largest classification rate, Fnorm and smallest AUC, F-score, Precision, and recall. Kmeans (M3) performs slightly better than global but still worse than others. IWLS algorithm always works better than ADMM regardless of the model structure. While adaptive model generally outperforms the original model with no adaptive penalty weights. In conclusion, besides the optimal model, the proposed adaptive network lasso model with IWLS based algorithm (M7) produces the best model fitting, data segmentation as well as prediction results.

Table 4.3: True Coefficients for Adjacent Clusters with $K = 4$ (D3)

Segment	β_0	β_1
1	-1	2.5
2	1.5	-3.5
3	0.5	1.5
4	-0.5	-1.5

Table 4.4: Coefficients Estimation results for Adjacent Clusters with $K = 4$ (D3)

Segment		True	GLM(Adaptive ADMM)	Adaptive ADMM	GLM(Adaptive IWLS)	Adaptive IWLS
			[1]	[2]	[3]	[4]
1	b_0	-1	-0.584	-0.299	-0.615	-0.548
	b_1	2.5	2.290	1.341	2.050	1.861
2	b_0	1.5	0.235	0.188	1.500	0.965
	b_1	-3.5	-1.468	-0.866	-1.959	-1.685
3	b_0	0.5	0.289	0.222	0.329	0.307
	b_1	1.5	1.150	0.850	1.566	1.309
4	b_0	-0.5	0.235	0.188	-0.087	-0.062
	b_1	-1.5	-1.468	-0.886	-1.284	-1.150

4.6 IBM Pricing Data Application

In this section, the proposed methodology is applied to the historical pricing data of IBM. Specifically, I evaluate two different pricing data sets corresponding to two leading brands, analytics platform and security. For simplicity, let us denote those two data sets as Brand1 and Brand2 throughout this chapter. The major goal is to fit logistic regression models to predict the purchase likelihood at each RFQ, such that data sharing the similar purchase behaviors can be clustered together. The adaptive network lasso model will be performed for each brand separately.

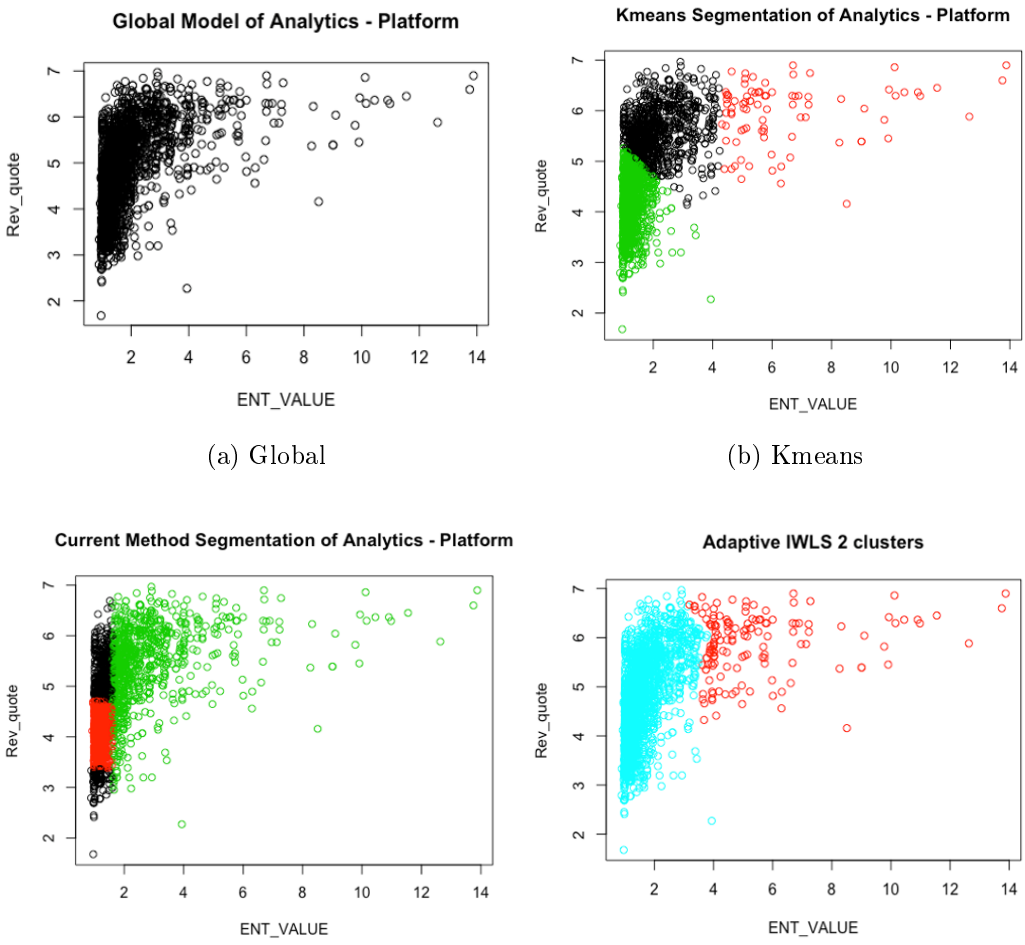
Brand1 contains $N_1 = 2682$ observations/RFQs, while Brand2 has $N_2 = 2642$ observations/RFQs. Under the network setting, each node i is a unique RFQ. Two RFQs are

Table 4.5: Modeling fitting results for Adjacent Clusters with $K = 4$ (D3)

	Classification Error	Fnorm	AUC	F-score	Precision	Recall
	[1]	[2]	[3]	[4]	[5]	[6]
Optimal	0.27	6.13	0.864	0.71	0.83	0.62
Global	0.51	71.35	0.488	0.52	0.53	0.52
Kmeans	0.44	68.43	0.601	0.63	0.58	0.69
ADMM	0.38	62.46	0.670	0.60	0.69	0.55
IWLS	0.38	55.72	0.670	0.63	0.68	0.59
Adaptive ADMM	0.36	52.45	0.708	0.62	0.72	0.55
Adaptive IWLS	0.36	47.65	0.715	0.68	0.65	0.70

connected if there is an edge between two corresponding nodes. Each node i is associated with three features constructed in Xue et al. (2015), ENT_VALUE (=Entitled Price/Value Score; X_1), Revenue (=log(Total Entitled Price); X_2), GAP_V*_Q (=Quote Price/Value Score; X_3). All of those features are modeled in the logistic regression model to predict purchase likelihood, while X_1 and X_2 are also chosen as weight features according to domain knowledge. To begin the adaptive network lasso modeling, each nodes (RFQ) is connected to its five nearest neighbors decided by the euclidean distance of (X_1, X_2). The binary response (Y_i) at each node is coded as $1/-1$ indicating whether or not the corresponding client made a purchase. At each node i, the goal is to solve for $\boldsymbol{\beta}_i = [\beta_{0i}, \beta_{1i}, \beta_{2i}, \beta_{3i}]^T$, which are the logistic regression coefficients, corresponding to historical observations $y_i, x_{i1}, x_{i2}, x_{i3}$.

For each brand, RFQs are randomly split into training (80%) and testing (20%) dataset. 5-fold cross validation is conducted to each training dataset in order to choose the optimal λ value as well as cutting point c in logistic regression model. Specifically, λ is chosen to maximize the median AUC values in 5-fold CV, while c is chosen such that the sum of sensitivity and specificity is maximized. Those optimal values are directly used in the testing data. I compare the proposed approach (M7) with Global model (M1), Kmeans (M2), as well as the current method implemented in Xue et al. (2015). The current method first conducts tree regression



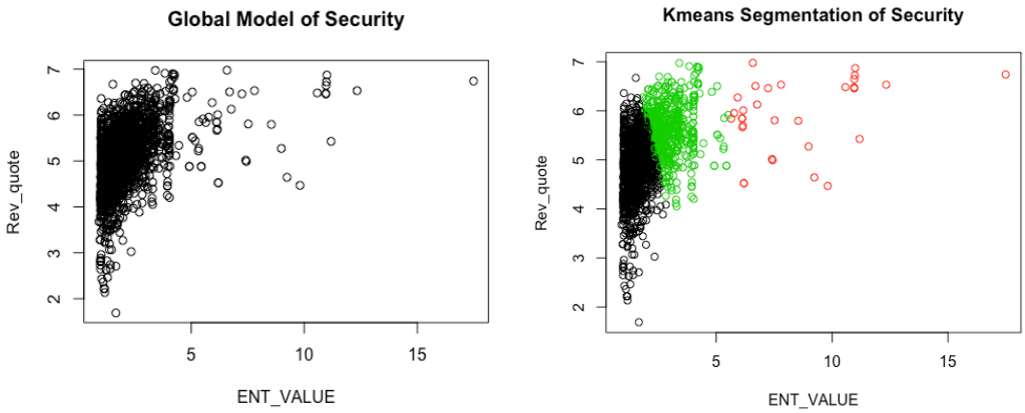
(a) Global (b) Kmeans

(c) Current (d) Adaptive Network Lasso

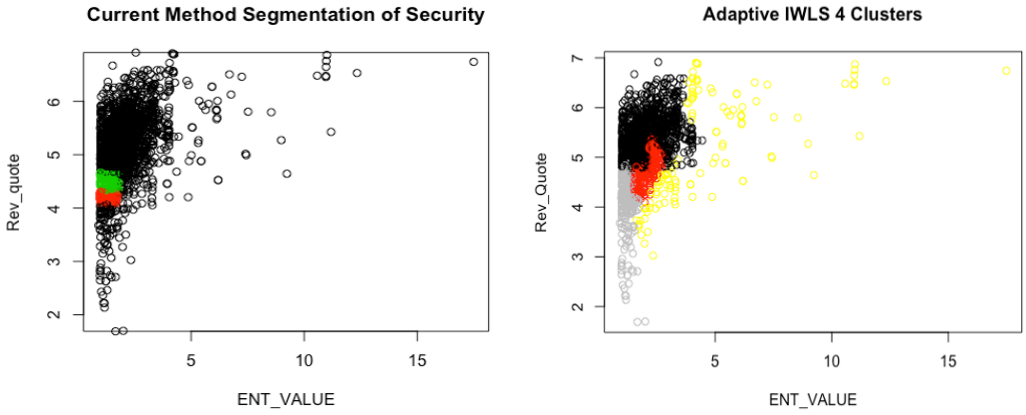
Figure 4.4: Segmentation Results Comparison for Brand1.

classification based on product features and then fits logistic regression models under each segment. The number of clusters used in Kmeans (M2) is the same as that obtained from the current method. All the fitting results for those four models are compared in Figure 4.4 and Figure 4.5, Table 4.6 and Table 4.7 for Brand1 and Brand2 respectively.

For Brand1, the adaptive network lasso only gives two segments. The prediction performance in Table 4.6 indicates that the proposed method is comparable to global and Kmeans methods with a slightly higher AUC. One possible explanation is that the underlying truth of Brand1 does not contain clear clustering structure. Thus data segmentation does not improve the prediction performance significantly compared with global model. However, the current



(a) Global (b) Kmeans



(c) Current (d) Adaptive Network Lasso

Figure 4.5: Segmentation Results Comparison for Brand2.

Table 4.6: Modeling performance comparison for Brand1

	Classification Error	F-score	AUC	Precision	Recall
	[1]	[2]	[3]	[4]	[5]
Global	0.40	0.53	0.653	0.47	0.62
Kmeans	0.39	0.55	0.652	0.48	0.65
Current	0.44	0.49	0.584	0.43	0.58
Adaptive network lasso	0.40	0.55	0.659	0.47	0.67

Table 4.7: Modeling performance comparison for Brand2

	Classification Error	F-score	AUC	Precision	Recall
	[1]	[2]	[3]	[4]	[5]
Global	0.40	0.41	0.614	0.32	0.57
Kmeans	0.41	0.41	0.614	0.32	0.56
Current	0.47	0.37	0.542	0.28	0.56
Adaptive network lasso	0.32	0.39	0.659	0.37	0.41

method makes the performance even worse compared to global, which is unacceptable.

For Brand2, the proposed method shows significant improvement especially in terms of the classification rate and AUC. While the current method still has the worst performance. In this scenario, the Brand2 does have clear hidden cluster structures and the proposed method shows its advantages in automatically detecting the segments as well as fitting the model.

4.7 Discussion and Conclusion

One type of data heterogeneity falls in the situation where different subsets of the data share heterogeneous model structures. This chapter considers this problem under a network setting, which is widely used for convex optimization. However, the traditional network lasso shows serious shrinkage problems when nodes from different segments are connected. Besides, the ADMM-based algorithm used in network lasso has convergence problems when the objec-

tive function is non-linear. To overcome those challenges, I proposed the adaptive network lasso with IWLS based algorithm. To alleviate the shrinkage problem, the penalty weights are iteratively re-weighted based on the similarity of their corresponding model coefficients. The construction of adaptive network lasso is also verified through the Bayesian perspective. What's more, the IWLS based algorithm is constructed to ensure the global convergence of coefficients estimation.

As mentioned in the introduction, customer attributes are also important characteristics that can impact the segmentation of data as well as model fitting. Unfortunately, the available data only contains products' attributes. The current approach may be improved if more customer attributes can be collected. Moreover, the initial network construction determines the edges based on the similarity measures between the nodes. Thus it is relatively unlikely to directly have an edge between the two nodes that are far way from each other in terms of their similarity meausres. In addition, the current network construction works quite well for separate or adjacent clusters. Identify overlapping segments is still challenging. However, using proper weight variables when connecting nodes may help reducing overlapping segments.

There are many other potential research ideas to build on. For example, in the real business application, it is desirable to obtain a relatively balanced segmentation structure among RFQs. Thus in order to avoid dominating segments or extreme small segments, one future research direction is to include the similarity of $\boldsymbol{X}$ between two connected nodes in the weight formula as follows,

$$w_{(j,k)}^{(t+1)} = \frac{1}{\frac{||\boldsymbol{\beta}_j^t - \boldsymbol{\beta}_k^t||_2}{\text{median}_{j,k}(||\boldsymbol{\beta}_j^t - \boldsymbol{\beta}_k^t||_2)} + \frac{||\boldsymbol{x}_j^t - \boldsymbol{x}_k^t||_2}{\text{median}_{j,k}(||\boldsymbol{x}_j^t - \boldsymbol{x}_k^t||_2)}}, \tag{4.10}$$

where, $\boldsymbol{x}_j = [x_{j1}, x_{j2}, x_{j3}]^T$, $\boldsymbol{\beta}_j = [\beta_{j0}, \beta_{j1}, \beta_{j2}, \beta_{j3}]$. At iteration t, the similarities for $\boldsymbol{\beta}^t$ and $\boldsymbol{x}^t$ are scaled by their median across all possible edges to ensure that two similarities are at the same scale.

Moreover, another open research topic is the stability of cross-validation for binary data. It is interesting to analyze how the model performance will vary across multiple CVs. And

how to stabilize it if large variations exist.

Chapter 5 Change-point Detection for Spatio-temporal Organ Image Data

5.1 Introduction

The demand for organ transplantation has rapidly increased all over the world during the last decades (Abouna, 2008). However, poor preservation and evaluation methods cause many organs to be discarded.

Current evaluation methods include doctors' visual inspection and pathologists' analysis of biopsy samples. However, doctors' evaluations are subjective, while taking biopsy samples could damage the organ itself. Thus, it is important to develop an objective and non-invasive method for evaluating the quality of organs. In this chapter, I focus on detecting quality changes in organs under preservation by only using a sequence of biomedical thermal images as shown in Figure 5.1. The images are taken at several different times of the same organ object under proper preservation. The objective is to identify the time point when organ quality transfers from normal to non-normal status.

In the literature, a wide range of methodologies have been discovered regarding the change detection in image sequences (Radke et al., 2005; Zhou et al., 2014). While most of those work focus on detecting regions (collection of pixels) of change in images of the same scene (Radke et al., 2005; Lu et al., 2004), this chapter is mainly interested in detecting the changes in time along image sequences.

One commonly used method is to apply time series models on individual pixels in the same location at different times (Elfishawy et al., 1991; Jain and Chau, 1995). However, those methods often overlook the spatial correlation among pixels. Another type of method is to reduce the high dimensional image data into one-dimension or lower-dimensional data before conducting traditional change detection method. For example, Kleynhans et al. (2014)

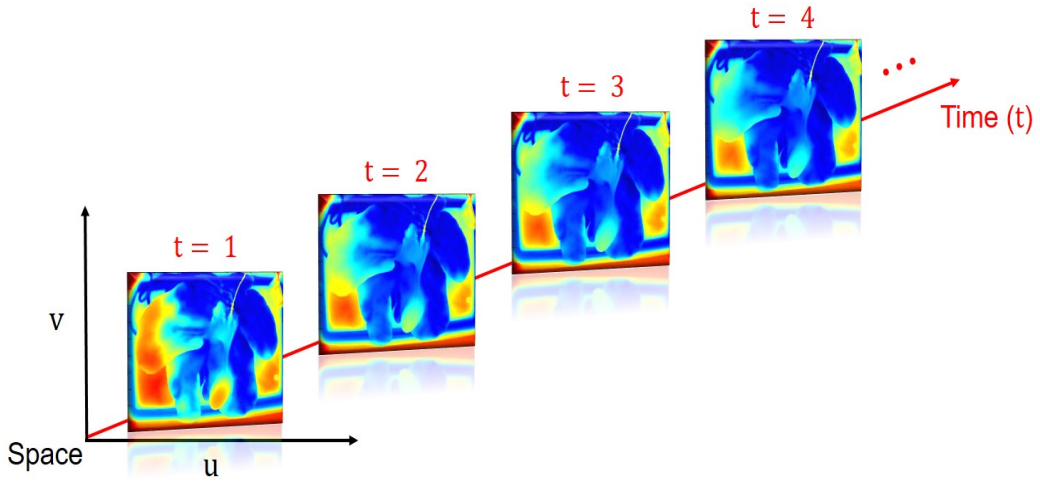


Figure 5.1: Thermal images of liver.

summarize spatial correlation among specific pixels at each time point using the Mahalanobis distance. Then change detection is directly applied on the Mahalanobis time series. In Wang et al. (2017), image features are extracted from each image upon which time series models are applied directly. Similarly, the information extraction technique is introduced by Mello et al. (2013), which synthesizes the full image dataset to one single synthetic multi-coefficient image.

Despite directly working on raw pixels or transferred information data, the other type of method depends on thresholding the pixel differences in consecutive images. For example, Clifton (2003) applies the neural network technique for predictive modeling in order to identify unusual changes in images. Neural networks are trained and then used to predict expected values for the same images used in training. A significant difference between the predictor and actual values is considered as an unusual change. Similarly, certain attributes are developed in Fang et al. (2006) and quality control techniques are applied to differenced attribute values for change detection. In other cases, an underlying process is assumed and changes are defined as the statistical parameter shifts from one value to another (Basseville and Nikiforov, 1993).

However, those models often are either hard to interpret or fail to explore the spatio-temporal dependency among pixels. In order to integrate spatial information into temporal

data analysis, Cressie and Wikle (2015) jointly model image sequence data by hierarchical statistical modeling, while Bolin et al. (2009) focus on random fields. But both of them are interested in detecting gradual changes. Besides, detecting changes from a sequence of high-resolution image data is also challenging due to the expensive computation of model estimation.

Gaussian process (GPs) are well-known approaches to automatically learn a wide variety of data structures especially for spatio-temporal images (Rasmussen, 1997; Williams and Rasmussen, 2006). The smoothness and generalization properties of a GP are mainly determined by the kernel function structure and a well-chosen kernel leads to impressive empirical performances (Rasmussen, 1997; Wilson et al., 2014). In order to fully explore the underlying correlation structure of image sequences, Wilson and Adams (2013) introduce a class of expressive spectral mixture kernels. This kernel structure contains a large set of stationary kernels and is able to approximate any stationary covariance kernel with required precisions. The flexible property in capturing a wide range of complex statistical structures makes it appropriate enough to explore the hidden dynamic in organ image sequences. Moreover, the computational cost in model estimation can be significantly reduced by making the kernel structure multiplicative across different dimensions (Wilson et al., 2014).

This chapter aims to discover the underlying data correlation structures of thermal image sequences. Based on the learned hidden dynamics, an interpretable approach is developed to estimate the change time when the organ transitions to non-normal status in a timely manner. A spatio-temporal Gaussian process model with spectral mixture kernels is constructed for model fitting and inference. Moreover, in order to capture the dynamic change in time, a moving window of image sequences is proposed, where the spatio-temporal Gaussian process model is applied in each window. Inspired by the area under the semi-variogram curve (AUC), I constructed an instructive statistic (V-statistic), to reveal the hidden characteristic of the organ dynamics automatically. Such a statistic is used to directly estimate the change-point when the status of the organ switches from normal to non-normal.

The change detection of image sequences commonly occurs in many processes. The diverse

applications include remote sensing, climate science, public health, ecology and environment science, etc. (Singh, 1989; Hansen et al., 2012; Zhou et al., 2014). Thus, the developed method in this chapter can also be useful for understanding hidden patterns in streaming data with complex structures arising in a broad range of applications.

The rest of this chapter is organized as follows. Section 5.2 gives a brief review of the Gaussian Process. Section 5.3 introduces the spectral mixture kernels. Section 5.4 details the proposed change detection approach image sequence observations. Then, in Section 5.5, the proposed approach is applied on liver image sequence data. Section 5.6 finishes with discussions and future work.

5.2 Review of Gaussian Process

This section provides a brief review of Gaussian processes(GP). A Gaussian process is a collection of random variables, any finite number of which have a joint Gaussian distribution.

Given data $(\boldsymbol{y}, \boldsymbol{X})$, where $\boldsymbol{y} = y_1, \ldots, y_N$, are responses or dependent variables, and $\boldsymbol{X} = [\boldsymbol{x}_1 \ldots \boldsymbol{x}_N]^T$, $\boldsymbol{x}_i \in \mathbb{R}^P$, are covariates or independent variables, each of dimension P. Assume that the responses $\boldsymbol{y}$ are generated from the covariates by an underlying function through a Gaussian noise model $y = f(\boldsymbol{x}) + \epsilon$, where, $f(\boldsymbol{x}) \sim GP(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}'))$, $\epsilon \sim N(0, \sigma^2)$. The mean function $m(\boldsymbol{x})$ and covariance kernel $k(\boldsymbol{x}, \boldsymbol{x}')$ are defined as,

$$m(\boldsymbol{x}) = E[f(\boldsymbol{x})],$$

$$k(\boldsymbol{x}, \boldsymbol{x}') = cov(f(\boldsymbol{x}), f(\boldsymbol{x}')).$$

Thus any collection of function values has a joint Gaussian distribution,

$$[f(\boldsymbol{x}_1), f(\boldsymbol{x}_2), \ldots, f(\boldsymbol{x}_N)]^T \sim \mathcal{N}(\boldsymbol{\mu}, K),$$

where $\mu_i = m(\boldsymbol{x}_i)$, and K is the $N \times N$ covariance matrix with entries $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$. The smoothness and generalization properties of the GP are encoded by the covariance kernel

function $k(\boldsymbol{x}, \boldsymbol{x}')$ and its hyperparameters $\boldsymbol{\theta}$.

In order to learn hyperparameters, we aim to optimize the log marginal likelihood of the data, conditional on kernel hyperparamters $\boldsymbol{\theta}$, and inputs $\boldsymbol{X}$,

$$\log p(\mathbf{y}|\boldsymbol{\theta}, \boldsymbol{X}) \propto -\log|K + \sigma^2 I| - \boldsymbol{y}^T (K + \sigma^2 I)^{-1} \boldsymbol{y}, \tag{5.1}$$

where all the responses $\boldsymbol{y}$ are centered for simplicity.

The choice of kernel functions are fundamental in Gaussian process modeling. Particularly, the stationary kernel functions have been extensively used in time series and spatial statistics (Genton, 2001). A stationary kernel is a function that is invariant to translation of the inputs, $k(\boldsymbol{x}, \boldsymbol{x}') = k(\boldsymbol{\tau}), \boldsymbol{\tau} = \boldsymbol{x} - \boldsymbol{x}'$. Thus it only depends on the lag vector $\boldsymbol{\tau}$. One popular and simple stationary kernel function is the squared exponential (SE),

$$k_{SE}(\boldsymbol{x}, \boldsymbol{x}') = \exp(-0.5||\boldsymbol{x} - \boldsymbol{x}'||^2 / l^2),$$

where the length scale l determines how quickly a Gaussian process function varies with $\boldsymbol{x} \in \mathbb{R}^P$. Other widely used kernel functions include, Matérn kernel, rational quadratic, etc.. However, those kernels' function structure are fixed and thus can only model the particular correlation structure in the data. While, the underlying true covariance structure is unknown and might be quite different.

5.3 Spectral Mixture Kernels

Finding a flexible kernel structure to model the underlying correlation among data is crucial in Gaussian process modeling. This chapter introduces a general set of kernel function, spectral mixture kernels. Directly modeling on $k(\boldsymbol{\tau})$ is nontrivial due to its covariance constraints. Thus one need to refer to some transformations.

Any stationary kernel functions can be constructed from their spectral representation

derived by Bochner (1955). Based on Chatfield (1959), one can get,

$$k(\boldsymbol{\tau}) = \int S(\boldsymbol{s}) e^{2\pi i \boldsymbol{s}^T \boldsymbol{\tau}} d\boldsymbol{s}, \tag{5.2}$$

$$S(\boldsymbol{s}) = \int k(\boldsymbol{\tau}) e^{2\pi i \boldsymbol{s}^T \boldsymbol{\tau}} d\boldsymbol{\tau}, \tag{5.3}$$

where $S(\boldsymbol{s})$ is called the spectral density or power spectrum of $k(\boldsymbol{\tau})$. And the $k(\boldsymbol{\tau})$ and $S(\boldsymbol{s})$ are Fourier duals. To get a stationary kernel $k(\boldsymbol{\tau})$, $S(\boldsymbol{s})$ must be symmetric (Rasmussen, 2006).

Thus a spectral density entirely determines the properties of a stationary kernel (Wilson and Adams, 2013). In order to achieve a relatively comprehensive representation of the kernel structure, Wilson and Adams (2013) suggest using a mixture of Gaussians for the spectral density. Since mixture Gaussians are able to approximate any functions given enough mixture components in the spectral representation, the corresponding kernel function is expected to approximate any stationary covariance kernels.

A simple construction for the symmetric density function $S(\boldsymbol{s})$ is, $S(\boldsymbol{s}) = [\phi(\boldsymbol{s}) + \phi(-\boldsymbol{s})]/2$. Let $\phi(\boldsymbol{s})$ be a mixture of Q Gaussians on $\mathbb{R}^P$, then

$$\phi(\boldsymbol{s}) = \sum_{q=1}^{Q} w_q |2\pi|^{-\frac{P}{2}} |M_q|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{s} - \boldsymbol{\mu}_q)^T M_q^{-1} (\boldsymbol{s} - \boldsymbol{\mu}_q)\right), \tag{5.4}$$

where the q^{th} component has mean vector $\boldsymbol{\mu}_q = (\mu_q^1, \cdots, \mu_q^P)$ and covariance matrix $M_q = diag(v_q^1, \cdots, v_q^P)$, and weight parameter w_q. Substituting $S(\boldsymbol{s})$ into Equation 5.2, we get,

$$k(\boldsymbol{\tau}) = \sum_{q=1}^{Q} w_q \cos(2\pi \boldsymbol{\tau}^T \boldsymbol{\mu}_q) \exp\left(-2\pi^2 \boldsymbol{\tau}^T M_q \boldsymbol{\tau}\right). \tag{5.5}$$

The weight w_q specifies the relative contribution of each mixture component. The inverse mean $1/\mu_q^i$ are the component periods, while the inverse standard deviations $1/\sqrt{v_q^i}$ are length scales. The class of kernel functions shown in Equation 5.5 is expressive in terms of containing many stationary kernels. Besides, it has a simple closed-form expression which can be directly

used in the Gaussian process for analysis and inference.

5.4 The Proposed Change Detection Method

As discussed in the previous sections, Gaussian process models with mixture kernels can learn complex correlation structures. Thus, it can be useful to discover the underlying dynamics of image sequences. However, under the change point detection scenario, the data correlation structure may change somewhere in the data sequence. Therefore, fitting one global GP model to the whole image sequence is too smooth and fails to detect the change time location. While fitting one GP model to each image may introduce too much noises making the estimation of true change point difficult. Thus, it is crucial to build a smooth and efficient framework to learn the hidden dynamics sequentially in time. Moreover, the estimated data correlation structures could not be used directly to conduct change detection. How to transfer the correlation structure into an interpretable statistic appropriately is also nontrivial.

To overcome those challenges, a moving window framework is introduced in section 5.4.1 based on which the local and global dynamic data structures can be explored continuously. In addition, a V-statistic is developed in section 5.4.2 to learn the underlying dynamics in a comprehensive manner. At last, the model fitting and inference of Gaussian process models are discussed in section 5.4.3.

We denote the observations in the image sequence data as $y(\mathrm{u}, \mathrm{v}, t)$. It is the gray intensity at each pixel located at (u, v) and time t as shown in Figure 5.2. Thus the thermal image data has three dimensions $P = 3$. Denote the total number of time as T, and total number of observations as n. The vector $\boldsymbol{y}_n$ is denoted as the $n \times 1$ vector containing all the observations.

5.4.1 Moving Window

In order to ensure local stationarity as well as provide a framework of change points monitoring sequentially in a timely and continuous manner, I introduce a moving window of images. Figure 5.3 shows how the overlapping moving window is constructed. Each moving window

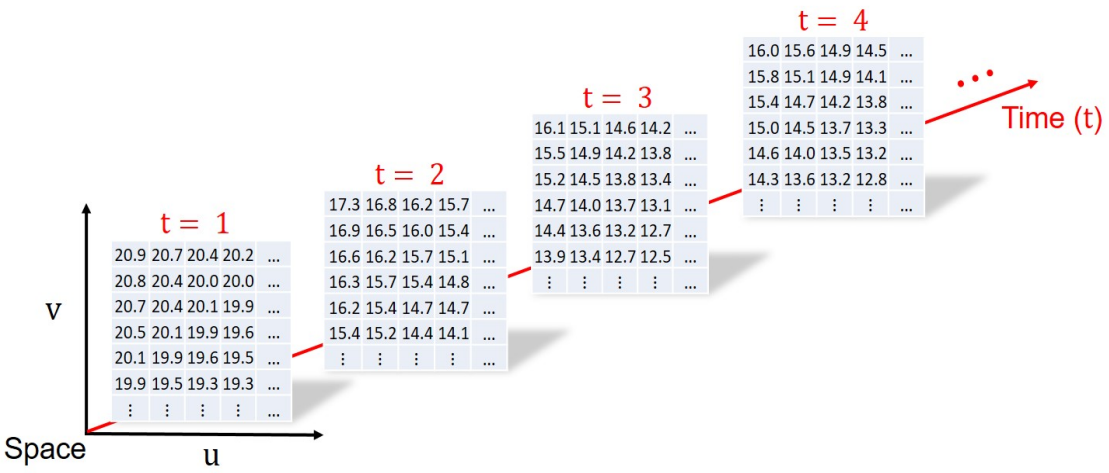


Figure 5.2: Data representation of thermal images.

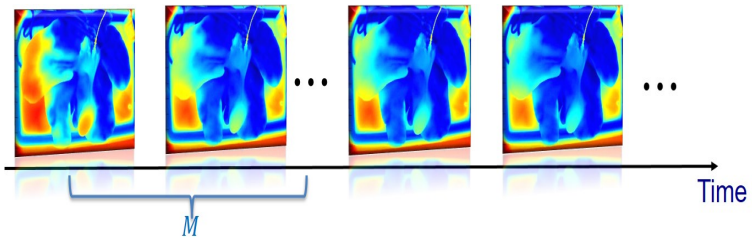
contains M consecutive images and we move one step ahead at each time. Thus T total images result in $(T - M + 1)$ moving windows with size M. The choice of window size value is a compromise of the ability to model the hidden structure of organ dynamic accurately and detect the changes quickly. Then spatio-temporal Gaussian process models with expressive mixture kernel functions will be fitted within each window.

5.4.2 V-Statistics and Image Change Detection

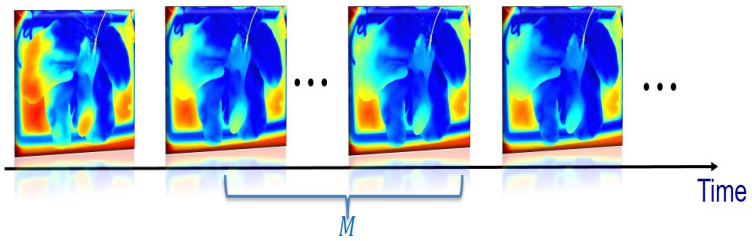
After exploring the Gaussian process model structures under each overlapping moving window, it is essential to find an instructive statistic summarizing the dynamic properties.

In spatio-temporal Gaussian process models, the semi-variogram is particularly used to show how the correlation within input data change across distances. It is defined as $r(\boldsymbol{h}) = K(\mathbf{0}) - K(\boldsymbol{h})$, where $\boldsymbol{h} = [h_{\mathrm{u}}, h_{\mathrm{v}}, h_t]^T$ is the distance vector. Given the Kronecker product assumption in this chapter, $K = K_{\mathrm{u}} \bigotimes K_{\mathrm{v}} \bigotimes K_t$, the corresponding semi-variogram can be written as,

$$\begin{aligned} r(h_{\mathrm{u}}, h_{\mathrm{v}}, h_t) &= K(0,0,0) - K(h_{\mathrm{u}}, h_{\mathrm{v}}, h_t) \\ &= k_{\mathrm{u}}(0)k_{\mathrm{v}}(0)k_t(0) - k_{\mathrm{u}}(h_{\mathrm{u}})k_{\mathrm{v}}(h_{\mathrm{v}})k_t(h_t). \end{aligned} \tag{5.6}$$



(a) First window of images



(b) Second window of images

Figure 5.3: Moving window of image sequences.

In one-dimensional scenario, the area under the semi-variogram (AUC) is often used to summarize the overall variation among observations (Langevin et al., 2007). Motivated by the idea of AUC as well as to construct a comprehensive statistic that can incorporate variations across all possible distance vectors, let us take the integral of Equation 5.6 in terms of $h_{\mathrm{u}}, h_{\mathrm{v}}, h_t$. Assume $l_{\mathrm{u}}, l_{\mathrm{v}}, l_t$ are the maximum distance values on each dimension, the proposed V-statistic is shown as,

$$\begin{aligned} V &\equiv \int_0^{l_{\mathrm{u}}} \int_0^{l_{\mathrm{v}}} \int_0^{l_t} r(h_{\mathrm{u}}, h_{\mathrm{v}}, h_t) d(h_{\mathrm{u}}) d(h_{\mathrm{v}}) d(h_t) \\ &= l_{\mathrm{u}} l_{\mathrm{v}} l_t k_{\mathrm{u}}(0) k_{\mathrm{v}}(0) k_t(0) - \int_0^{l_{\mathrm{u}}} k_{\mathrm{u}}(h_{\mathrm{u}}) d(h_{\mathrm{u}}) \int_0^{l_{\mathrm{v}}} k_{\mathrm{v}}(h_{\mathrm{v}}) d(h_{\mathrm{v}}) \int_0^{l_t} k_t(h_t) d(h_t). \end{aligned} \tag{5.7}$$

Next, let us standardize the V-statistic by $l_{\mathrm{u}}, l_{\mathrm{v}}, l_t$. Then the final proposed V-statistic is,

$$V \equiv k_{\mathrm{u}}(0) k_{\mathrm{v}}(0) k_t(0) - \frac{\int_0^{l_{\mathrm{u}}} k_{\mathrm{u}}(h_{\mathrm{u}}) d(h_{\mathrm{u}}) \int_0^{l_{\mathrm{v}}} k_{\mathrm{v}}(h_{\mathrm{v}}) d(h_{\mathrm{v}}) \int_0^{l_t} k_t(h_t) d(h_t)}{l_{\mathrm{u}} l_{\mathrm{v}} l_t}. \tag{5.8}$$

According to Equation 5.8, one V-statistic value can be obtained from each moving window

resulting in $(T - M + 1)$ statistic values in total. Those values can be directly used to conduct change detection.

5.4.3 Model Fitting and Inference

The parameter estimation mainly focuses on the log likelihood of the observations in Equation 5.1, where the kernel hyperparameters $\boldsymbol{\theta} = \left(w_q, \boldsymbol{\mu}_q, \boldsymbol{v}_q\right)_{q=1,\cdots,Q}$, and $\boldsymbol{v}_q = \left(v_q^1, \cdots, v_q^P\right)$.

Nonlinear conjugate gradients are used to conduct the likelihood optimization. One major computational challenge comes from the calculation of $(K+\sigma^2 I)^{-1}$ for high-dimensional image data.

If K is $N \times N$ with full grid structure, the computational burden can be reduced by assuming a Kronecker product $K = K_1 \bigotimes K_2 \bigotimes \cdots \bigotimes K_P$. On each dimension, K_i is square, positive definite, and having the mixture structure in Equation 5.5 with one dimension. Thus, the complete kernel structure K has Q mixture component on each dimension with a total of $3PQ$ kernels hyperparameters $\boldsymbol{\theta}$ and one noise hyperparameter σ^2.

However, if K is not on a complete grid and only n original data points are observed, one can form a complete grid using w imaginary or missing observations, $\boldsymbol{y}_w \sim N(\boldsymbol{f}_w, \epsilon^{-1} I_w)$, with $\epsilon \to 0$, (Wilson et al., 2014). Thus the total data vector $\boldsymbol{y}_N = [\boldsymbol{y}_n, \boldsymbol{y}_w]$ with $N = n + w$, $\boldsymbol{y}_N \sim N(\boldsymbol{f}, D_N)$, where $D_N = \text{diag}(\sigma^2 I_n, \epsilon^{-1} I_w)$. Furthermore, It is proved in Wilson et al. (2014) that,

$$\lim_{\epsilon \to 0} (K_N + D_N)^{-1} \boldsymbol{y}_N = (K_n + D_n)^{-1} \boldsymbol{y}_n, \lim_{\epsilon \to 0} \boldsymbol{y}_N (K_N + D_N)^{-1} \boldsymbol{y}_N = \boldsymbol{y}_n (K_n + D_n)^{-1} \boldsymbol{y}_n, \quad (5.9)$$

where $D_n = \sigma^2 I_n$. Thus, by introducing the imaginary observations, $(K_n + D_n)^{-1} \boldsymbol{y}_n$ can be obtained through the computation of $(K_N + D_N)^{-1} \boldsymbol{y}_N$ by preconditioned conjugate gradients (PCG) (Süli and Mayers, 2003).

The model estimation also requires the calculation of $\log|K_n + \sigma^2 I|$. Since it is non-trivial

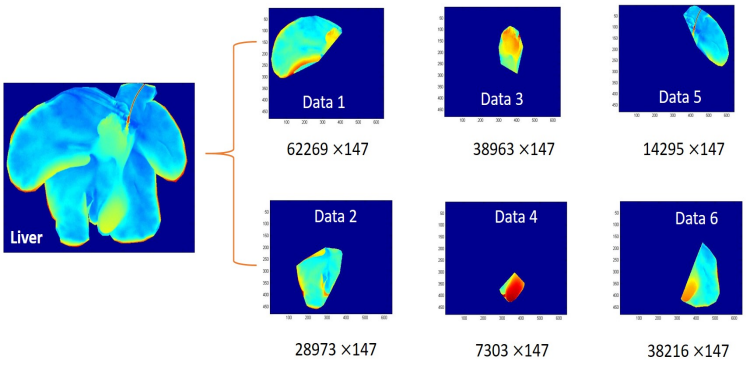


Figure 5.4: Liver Segmentation.

to decompose $K_n + \sigma^2 I$, Wilson et al. (2014) propose an approximation approach by,

$$\log|K_n + \sigma^2 I| = \sum_{i=1}^{n} \log\left(\lambda_i^n + \sigma^2\right) \approx \sum_{i=1}^{n} \log\left(\tilde{\lambda}_i^n + \sigma^2\right), \tag{5.10}$$

where $\tilde{\lambda} = \frac{n}{N}\lambda_i^N$, $i = 1, \cdots, n$. That is, the eigenvalues of K_n, λ_i^n is approximated using the eigenvalues of K_N, λ_i^N. This approximation is proved to be effective particularly when n is large (e.g. $n > 1000$) (Williams and Seeger, 2000).

5.5 Application of Biomedical Thermal Image Data

5.5.1 Image Data Pre-processing and Segmentation

Images of a liver under preservation are taken every 10 minutes resulting in $T = 147$ images in total. The observation $y(\mathrm{u}, \mathrm{v}, t)$ is the intensity of pixels at each location and time with $P = 3$. The shape of liver is irregular and thus the corresponding kernel matrix K is not on a full gird. The background is removed from the image and according to domain experts, each liver can be further segmented into six regions as shown in Figure 5.4. This can be achieved by the Image Processing Toolbox (IPT) in MATLAB.

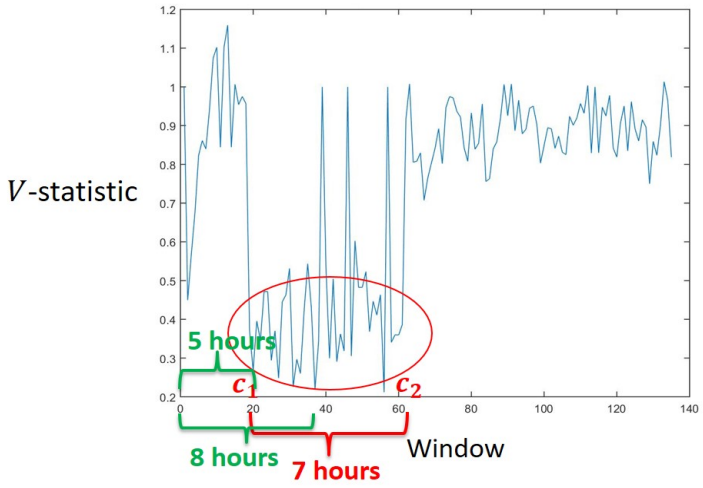


Figure 5.5: V-statistics for all moving windows.

5.5.2 Model Fitting and Change Detection Results

The Gaussian process models with mixture kernel function are fitted window each moving window for each liver segment. The number of mixture component in the kernel function is fixed at $Q = 10$ according to the recommendation in Wilson and Adams (2013); Wilson et al. (2014); Wilson and Nickisch (2015). In this application, the moving window size $W = 10$ is chosen due to its relative good performance.

For each segment, $(T - M + 1) = 138$ V-statistic values are calculated in total. For example, all those results for Segment 3 are shown in Figure 5.5. It clearly shows that the whole procedure can be segmented into three different periods by time points $c_1 = 5$hrs and $c_2 = 12$hrs. Note that the pattern is quite obvious such that any basic quality control charts such as Shewhart can detect those changing periods. Specially, the middle period between c_1 and c_2 has relatively low V-statistic which corresponds to the time period when the organ transits from normal to non-normal status. The relatively low values of V-statistic may be due to the high variations in the corresponding semi-variogram, which leads to low V-statistic values. While a relative stable semi-variogram corresponds to a larger value of V-statistic, such as the initial period $0 - c_1$ hrs and the last period $c_2 - 24$ hrs. At time goes on, the organ will become stable due to the organ dying. The images will be similar in terms of

intensity values, resulting in stable covariance structures in both space and time. Specifically, the correlation between two pixels decreases to zero rapidly as their distance is larger than zero on each dimension. Thus the V-statistic will finally converge to a constant value.

According to the domain expert's suggestion, the liver is no longer appropriate to be transplanted after 8-hrs' preservation, which is right in the middle of the transition period we estimated. However, our proposed method indicates that the liver quality begins to change after 5-hrs' preservation. To be more conservative, our recommendation is 3-hrs earlier than doctor's suggestion.

5.6 Discussion

Spatial-temporal model with scalable Gaussian processes is applied to biomedical thermal image data as an accurate and non-invasive evaluation method. Kronecker product is approximated for incomplete grid observations, making the computation of large multidimensional image data possible. A moving window is proposed to ensure local stationary and monitor organ quality changes in a timely manner. Similar to the idea of area under semi-variogram (AUC), I developed the V-statistic to represent the total variation changes in organ dynamics under each moving window. The proposed methodology is also useful for understanding hidden patterns in streaming data with complex structures arising in other applications, such as heart physiology, brain activity, live video streaming, and meteorological monitoring.

Although the moving window size(M) is fixed in the current work, one future topic is focusing on automatically choosing the value by including M as a model parameter. Other future research directions include finding the optimal number of mixture components(Q). One possible solution is to employ a Dirichlet process prior on mixture weights (Görür and Edward Rasmussen, 2010; Neal, 2000). Specifically, the Dirichlet process mixture model is,

$$y_i|\theta_i \sim F(\theta_i),$$

$$\theta_i|G \sim G,$$

$$G \sim DP(G_0, \alpha),$$

where α is the concentration parameter, G_0 is the base distribution and $\boldsymbol{\theta}_i$ denotes the set of parameters for component i including the mixing proportions π_i (which must be positive and sum to one). The inference can be handled by appropriate Markov chain methods for sampling from the posterior distribution.

Chapter 6 General Conclusion

In this dissertation, I develop a robust GLR (generalized likelihood ratio) control chart that can work for non-normal observations for monitoring the process mean, a mixed SSSM (switching state-space model) to detect change-point(s) for observations containing both continuous and discrete observations, an adaptive network lasso methodology to simultaneously conduct data segmentation and model fitting, and a Gaussian process based approach to detect changes on a sequence of image data.

In Chapter 2, several methods are investigated to refine the standard GLR control chart robust to non-normal observations for monitoring the process mean. The proposed robust GLR chart with χ^2 transformation is to make the transformed data points to behave more like following a normal distribution, such that the 'extreme' observations will not trigger a signal as the standard GLR control chart could do. Extensive simulations have been conducted to show that the proposed chart has an overall better performance in detecting mean changes over a wide range of shifts. The proposed method can detect changes efficiently and reduce the false alarm rates significantly. Moreover, the proposed robust GLR chart only contains one chart parameter and thus can be easily implemented in a wide of applications.

In Chapter 3, I consider the change-point detection for mixed-type observations using a Bayesian approach. A latent process method, so-called mixed switching state-space model (mixed SSSM), is proposed to jointly model the mixed-type observations and effectively detect the change-points. By including a continuous process and a discrete latent process in the mixed SSSM, the proposed method can quantify the hidden association in mixed-type observations and accommodate change-points simultaneously. Efficient parameter estimation and Bayesian inference are developed by iteratively combining discrete particle filter and sequential Monte Carlo algorithms. The proposed approach is applied to analyze the civil unrest data in three Latin American countries, showing a superior performance of effectively detecting

events related to civil unrest such as strike and protests. The proposed method can also be applied to change-point detection problems in other areas, such as internet of things, network surveillance, and service industry.

Chapter 4 considers a special case of the heterogeneous data structure. The heterogeneity arises when underlying models are heterogeneous for different data segments. I propose a self-segmented classification method to perform segmentation and model fitting in a simultaneous fashion. This method ensures that the estimated models under different segments are distinctive and data points with a common model structure are grouped into the same segment. The key idea of the proposed method is to impose an adaptive network LASSO penalty on a set of model coefficient vectors, encouraging their homogeneity within each segment and heterogeneity across different segments. Furthermore, an iteratively weighted least squares (IWLS) algorithm is developed to achieve a fast convergence rate in parameter estimation by linearizing the nonlinear objective functions. I apply the proposed method to the IBM pricing data, and it produced better predictions for the purchase probability. Although the current work adopts logistic regression for classification, it can also accommodate other classification methods such as support vector machines. The proposed method can be easily extended to allow variable selection by imposing additional penalties. This self-segmented modeling approach can also be applied in many other areas with model heterogeneity across multiple data segments, such as customer retention prediction in marketing analyses and dose limiting toxicity in clinical trials studies.

In Chapter 5, I aim to find an accurate and non-invasive method for evaluating the quality of organs based on a sequence of thermal image data. A spatio-temporal Gaussian process model with spectral mixture kernels is constructed for model fitting and inference. The ability to approximate any stationary covariance kernel with required precisions makes this kernel structure flexible in capturing a wide range of complex statistical structures. Moreover, the computational cost in model estimation is significantly reduced by making the kernel structure multiplicative across different dimensions. By applying the spatio-temporal Gaussian process models in a moving window of image sequences, I construct an instructive statistic

(V-statistics), to reveal the hidden characteristic of the organ dynamics automatically. Such a statistic is used to directly estimate the change-point when the status of the organ switches from normal to non-normal. This method can also be useful for understanding hidden patterns in streaming data with complex structures arising in other applications, such as heart physiology, brain activity, live video streaming, and meteorological monitoring.

References

Abouna, G. M. (2008), "Organ shortage crisis: problems and possible solutions," in *Transplantation Proceedings*, Elsevier, vol. 40, pp. 34–38.

Amin, R. W., Reynolds Jr, M. R., and Saad, B. (1995), "Nonparametric quality control charts based on the sign statistic," *Communications in Statistics-Theory and Methods*, 24, 1597–1623.

Andrieu, C., Doucet, A., and Holenstein, R. (2010), "Particle markov chain monte carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72, 269–342.

Basseville, M. and Nikiforov, I. V. (1993), *Detection of abrupt changes: theory and application*, vol. 104, Englewood Cliffs, NJ: Prentice Hall.

Bochner, S. (1955), *Harmonic analysis and the theory of probability*, Los Angeles, California: University of California Press.

Bolin, D., Lindström, J., Eklundh, L., and Lindgren, F. (2009), "Fast estimation of spatially dependent temporal vegetation trends using Gaussian Markov random fields," *Computational Statistics & Data Analysis*, 53, 2885–2896.

Borror, C. M., Montgomery, D. C., and Runger, G. C. (1999), "Robustness of the EWMA control chart to non-normality," *Journal of Quality Technology*, 31, 309–316.

Boyd, S., Parikh, N., Chu, E., Peleato, B., and Eckstein, J. (2011), "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, 3, 1–122.

Cappé, O., Moulines, E., and Rydén, T. (2009), "Inference in hidden markov models," in *Proceedings of EUSFLAT Conference*, pp. 14–16.

Carter, C. K. and Kohn, R. (1994), "On Gibbs sampling for state space models," *Biometrika*, 81, 541–553.

Chakraborti, S., Van der Laan, P., and Bakir, S. (2001), "Nonparametric control charts: an overview and some results," *Journal of Quality Technology*, 33, 304–315.

Chatfield, C. (1959), *The analysis of time series: an introduction*, Princeton, New Jersey: Princeton University Press.

Chen, J. and Gupta, A. K. (1997), "Testing and locating variance changepoints with application to stock prices," *Journal of the American Statistical Association*, 92, 739–747.

Chen, J. and Gupta, A. K. (2011), *Parametric statistical change point analysis: with applications to genetics, medicine, and finance*, New York, NY: Springer.

Chen, R. and Liu, J. S. (2000), "Mixture kalman filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62, 493–508.

Chen, Z. (2013), "An overview of bayesian methods for neural spike train analysis," *Computational Intelligence and Neuroscience*, 2013, 1.

Chen, Z. and Brown, E. N. (2013), "State space model," *Scholarpedia*, 8, 30868.

Clifton, C. (2003), "Change detection in overhead imagery using neural networks," *Applied Intelligence*, 18, 215–234.

Cressie, N. and Wikle, C. K. (2015), *Statistics for spatio-temporal data*, John Wiley & Sons.

Crosier, R. B. (1986), "A new two-sided cumulative sum quality control scheme," *Technometrics*, 28, 187–194.

de Leon, A. R. and Wu, B. (2011), "Copula-based regression models for a bivariate mixed discrete and continuous outcome," *Statistics in Medicine*, 30, 175–185.

Deng, X. and Jin, R. (2015), "QQ Models: Joint modeling for quantitative and qualitative quality responses in manufacturing systems," *Technometrics*, 57, 320–331.

Doucet, A., De Freitas, N., and Gordon, N. (2001a), "An introduction to sequential Monte Carlo methods," *SMC in Practice*.

Doucet, A., Godsill, S., and Andrieu, C. (2000), "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statistics and Computing*, 10, 197–208.

Doucet, A., Gordon, N. J., and Krishnamurthy, V. (2001b), "Particle filters for state estimation of jump Markov linear systems," *IEEE Transactions on Signal Processing*, 49, 613–624.

Eckley, I. A., Fearnhead, P., and Killick, R. (2011), "Analysis of changepoint models," *Bayesian Time Series Models*, 205–224.

Elfishawy, A., Kesler, S., and Abutaleb, A. (1991), "Adaptive algorithms for change detection in image sequence," *Signal Processing*, 23, 179–191.

Erdman, C. and Emerson, J. W. (2008), "A fast Bayesian change point analysis for the segmentation of microarray data," *Bioinformatics*, 24, 2143–2148.

Fang, Y., Ganguly, A. R., Singh, N., Vijayaraj, V., Feierabend, N., and Potere, D. T. (2006), "Online change detection: Monitoring land cover from remotely sensed data," in *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, IEEE, pp. 626–631.

Fearnhead, P. (1998), "Sequential Monte Carlo methods in filter theory," Ph.D. thesis, University of Oxford.

Fearnhead, P. and Clifford, P. (2003), "On-line inference for hidden Markov models via particle filters," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65, 887–899.

Frühwirth-Schnatter, S. (2006), *Finite mixture and Markov switching models*, New York, NY: Springer.

Gan, F. (1994), "Design of optimal exponential CUSUM control charts," *Journal of Quality Technology*, 26, 109–124.

Ge, X. and Smyth, P. (2000), "Segmental semi-Markov models for change-point detection with applications to semiconductor manufacturing," Tech. rep., University of California at Irvine, March.

Genton, M. G. (2001), "Classes of kernels for machine learning: A statistics perspective," *Journal of Machine Learning Research*, 2, 299–312.

Görür, D. and Edward Rasmussen, C. (2010), "Dirichlet process gaussian mixture models: Choice of the base distribution," *Journal of Computer Science and Technology*, 25, 653–664.

Gupta, A. and Chen, J. (1996), "Detecting changes of mean in multidimensional normal sequences with applications to literature and geology," *Computational Statistics*, 11, 211–221.

Hallac, D., Leskovec, J., and Boyd, S. (2015), "Network lasso: clustering and optimization in large graphs," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 387–396.

Hansen, J., Sato, M., and Ruedy, R. (2012), "Perception of climate change," *Proceedings of the National Academy of Sciences*, 109, E2415–E2423.

Hartikainen, J., Riihimäki, J., and Särkkä, S. (2011), "Sparse spatio-temporal Gaussian processes with general likelihoods," in *International Conference on Artificial Neural Networks*, Springer, pp. 193–200.

Hawkins, D. M. and Olwell, D. H. (2012), *Cumulative sum charts and charting for quality improvement*, New York, NY: Springer-Verlag.

Hawkins, D. M., Qiu, P., and Kang, C. W. (2003), "The changepoint model for statistical process control," *Journal of Quality Technology*, 35, 355–366.

Hestenes, M. R. (1969), "Multiplier and gradient methods," *Journal of Optimization Theory and Applications*, 4, 303–320.

Hinkley, D. V. and Hinkley, E. A. (1970), "Inference about the change-point in a sequence of binomial variables," *Biometrika*, 57, 477–488.

Holenstein, R. (2009), "Particle markov chain monte carlo," Ph.D. thesis, University of British Columbia.

Hwang, S. L., Lin, J. T., Liang, G. F., Yau, Y. J., Yenn, T. C., and Hsu, C. C. (2008), "Application control chart concepts of designing a pre-alarm system in the nuclear power plant control room," *Nuclear Engineering and Design*, 238, 3522–3527.

Jain, Z.-S. and Chau, Y. A. (1995), "Optimum multisensor data fusion for image change detection," *IEEE Transactions on Systems, Man, and Cybernetics*, 25, 1340–1347.

Jiang, H., Lozano, A. C., and Liu, F. (2012), "A Bayesian Markov-switching model for sparse dynamic network estimation." in *SDM*, SIAM, pp. 506–515.

Jung, T. and Wickrama, K. (2008), "An introduction to latent class growth analysis and growth mixture modeling," *Social and Personality Psychology Compass*, 2, 302–317.

Kalman, R. E. (1960), "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, 82, 35–45.

Kleynhans, W., Salmon, B. P., and Wessels, K. J. (2014), "A novel spatio-temporal change detection approach using hyper-temporal satellite data," in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, IEEE, pp. 4208–4211.

Koller, D. and Friedman, N. (2009), *Probabilistic graphical models: principles and techniques*, Cambridge, MA: The MIT Press.

Lai, T. L. (1998), "Information bounds and quick detection of parameter changes in stochastic systems," *IEEE Transactions on Information Theory*, 44, 2917–2929.

Lai, T. L. (2001), "Sequential analysis: some classical problems and new challenges," *Statistica Sinica*, 11, 303–350.

Langevin, H. M., Rizzo, D. M., Fox, J. R., Badger, G. J., Wu, J., Konofagou, E. E., Stevens-Tuttle, D., Bouffard, N. A., and Krag, M. H. (2007), "Dynamic morphometric characterization of local connective tissue network structure in humans using ultrasound," *BMC Systems Biology*, 1, 25.

Lee, A., Caron, F., Doucet, A., and Holmes, C. (2010), "A hierarchical Bayesian framework for constructing sparsity-inducing priors," *arXiv preprint arXiv:1009.1914*.

Liesenfeld, R. and Richard, J.-F. (2008), "Improving MCMC, using efficient importance sampling," *Computational Statistics & Data Analysis*, 53, 272–288.

Lio, P. and Vannucci, M. (2000), "Wavelet change-point prediction of transmembrane proteins," *Bioinformatics*, 16, 376–382.

Liu, J. S. (2008), *Monte Carlo strategies in scientific computing*, New York, NY: Springer.

Lu, D., Mausel, P., Brondizio, E., and Moran, E. (2004), "Change detection techniques," *International Journal of Remote Sensing*, 25, 2365–2401.

Meila, M. and Jordan, M. I. (2000), "Learning with mixtures of trees," *Journal of Machine Learning Research*, 1, 1–48.

Meinshausen, N. and Bühlmann, P. (2006), "High-Dimensional Graphs and Variable Selection with the Lasso," *The Annals of Statistics*, 34, 1436–1462.

Mello, M. P., Vieira, C. A., Rudorff, B. F., Aplin, P., Santos, R. D., and Aguiar, D. A. (2013), "STARS: A new method for multitemporal remote sensing," *IEEE Transactions on Geoscience and Remote Sensing*, 51, 1897–1913.

Montgomery, D. C. (2004), *Introduction to statistical quality control*, New York, NY: John Wiley & Sons, 5th ed.

Müller, H.-G. and Siegmund, D. (1994), "Change-point problems," Hayward, CA: Institute of Mathematical Statistics.

Muthen, B. (2001), "Latent variable mixture modeling," *New Developments and Techniques in Structural Equation Modeling*, 1–33.

Neal, R. M. (2000), "Markov chain sampling methods for Dirichlet process mixture models," *Journal of Computational and Graphical Statistics*, 9, 249–265.

Ning, X. and Tsung, F. (2012), "A density-based statistical process control scheme for high-dimensional and mixed-type observations," *IIE Transactions*, 44, 301–311.

Nylund, K. L., Asparouhov, T., and Muthén, B. O. (2007), "Deciding on the number of classes in latent class analysis and growth mixture modeling: A Monte Carlo simulation study," *Structural Equation Modeling*, 14, 535–569.

O'brien, S. P. (2010), "Crisis early warning and decision support: Contemporary approaches and thoughts on future research," *International Studies Review*, 12, 87–104.

Page, E. (1954), "Continuous inspection schemes," *Biometrika*, 41, 100–115.

Parikh, N. and Boyd, S. P. (2014), "Proximal algorithms," *Foundations and Trends in Optimization*, 1, 127–239.

Peng, Y. and Reynolds Jr, M. R. (2014), "A GLR Control Chart for Monitoring the Process Mean with Sequential Sampling," *Sequential Analysis*, 33, 298–317.

Peng, Y., Xu, L., and Reynolds, M. R. (2015), "The design of the variable sampling interval generalized likelihood ratio Chart for monitoring the process mean," *Quality and Reliability Engineering International*, 31, 291–296.

Qiu, P. (2008), "Distribution-free multivariate process control based on log-linear modeling," *IIE Transactions*, 40, 664–677.

Radke, R. J., Andra, S., Al-Kofahi, O., and Roysam, B. (2005), "Image change detection algorithms: a systematic survey," *IEEE Transactions on Image Processing*, 14, 294–307.

Ramakrishnan, N., Butler, P., Muthiah, S., Self, N., Khandpur, R., Saraf, P., Wang, W., Cadena, J., Vullikanti, A., Korkmaz, G., et al. (2014), "'Beating the news' with EMBERS: forecasting civil unrest using open source indicators," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, pp. 1799–1808.

Rasmussen, C. E. (1997), "Evaluation of Gaussian processes and other methods for non-linear regression," Ph.D. thesis, National Library of Canada= Bibliothèque nationale du Canada.

Rasmussen, C. E. (2006), "Gaussian processes for machine learning," .

Reynolds, M. R. and Stoumbos, Z. G. (2010), "Robust CUSUM charts for monitoring the process mean and variance," *Quality and Reliability Engineering International*, 26, 453–473.

Reynolds Jr, M. R. and Lou, J. (2010), "An evaluation of a GLR control chart for monitoring the process mean," *Journal of Quality Technology*, 42, 287–310.

Reynolds Jr, M. R. and Lou, J. (2012), "A GLR control chart for monitoring the process variance," in *Frontiers in Statistical Quality Control 10*, Springer, pp. 3–17.

Reynolds Jr, M. R., Lou, J., Lee, J., and Wang, S. (2013), "The design of GLR control charts for monitoring the process mean and variance," *Journal of Quality Technology*, 45, 34–60.

Reynolds Jr, M. R. and Stoumbos, Z. G. (1999), "A CUSUM chart for monitoring a proportion when inspecting continuously," *Journal of Quality Technology*, 31, 87–108.

Reynolds Jr, M. R. and Stoumbos, Z. G. (2004a), "Control charts and the efficient allocation of sampling resources," *Technometrics*, 46, 200–214.

Reynolds Jr, M. R. and Stoumbos, Z. G. (2004b), "Should observations be grouped for effective process monitoring?" *Journal of Quality Technology*, 36, 343–366.

Roberts, G. O., Gelman, A., Gilks, W. R., et al. (1997), "Weak convergence and optimal scaling of random walk Metropolis algorithms," *The Annals of Applied Probability*, 7, 110–120.

Roberts, S. (2000), "Control chart tests based on geometric moving averages," *Technometrics*, 42, 97–101.

Saniga, E. M. (1989), "Economic statistical control-chart designs with an application to and R charts," *Technometrics*, 31, 313–320.

Scrucca, L. (2004), "qcc: an R package for quality control charting and statistical process control," *R News*, 4, 11–17.

Shewhart, W. A. (1931), *Economic control of quality of manufactured product*, New York, NY: Van Nostrand.

Shiryaev, A. N. (1963), "On optimum methods in quickest detection problems," *Theory of Probability & Its Applications*, 8, 22–46.

Singh, A. (1989), "Review article digital change detection techniques using remotely-sensed data," *International Journal of Remote Sensing*, 10, 989–1003.

Spokoiny, V. (2009), "Multiscale local change point detection with applications to value-at-risk," *The Annals of Statistics*, 1405–1436.

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

Süli, E. and Mayers, D. F. (2003), *An introduction to numerical analysis*, Cambridge University Press.

Testik, M. C., Runger, G. C., and Borror, C. M. (2003), "Robustness properties of multivariate EWMA control charts," *Quality and Reliability Engineering International*, 19, 31–38.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005), "Sparsity and smoothness via the fused lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67, 91–108.

Wahlberg, B., Boyd, S., Annergren, M., and Wang, Y. (2012), "An ADMM algorithm for a class of total variation regularized estimation problems," *IFAC Proceedings Volumes*, 45, 83–88.

Wang, X., Liu, Y., Ling, F., Liu, Y., and Fang, F. (2017), "Spatio-Temporal change detection of Ningbo coastline using landsat time-series images during 1976–2015," *ISPRS International Journal of Geo-Information*, 6, 68.

Weinberger, K. Q., Sha, F., Zhu, Q., and Saul, L. K. (2007), "Graph Laplacian regularization for large-scale semidefinite programming," *Advances in Neural Information Processing Systems*, 19, 1489.

Whiteley, N., Andrieu, C., and Doucet, A. (2010), "Efficient Bayesian inference for switching state-space models using discrete particle Markov chain Monte Carlo methods," Tech. Rep. 10:04, Bristol Statistics Research Report.

Wikipedia (2015), "September 2012 cacerolazo in Argentina — Wikipedia, The Free Encyclopedia," `https://en.wikipedia.org/w/index.php?title=September_2012_cacerolazo_in_Argentina&oldid=696215834`, [Online; accessed 21-December-2015].

Wikipedia (2016a), "2013 protests in Brazil — Wikipedia, The Free Encyclopedia," `https://en.wikipedia.org/w/index.php?title=2013_protests_in_Brazil&oldid=744677098`, [Online; accessed 16-October-2016].

Wikipedia (2016b), "2014-16 Venezuelan protests — Wikipedia, The Free Encyclopedia," `https://en.wikipedia.org/w/index.php?title=2014%E2%80%9316_Venezuelan_protests&oldid=741406311`, [Online; accessed 27-September-2016].

Williams, C. K. and Rasmussen, C. E. (2006), "Gaussian processes for machine learning," *the MIT Press*, 2, 4.

Williams, C. K. and Seeger, M. (2000), "Using the Nyström method to speed up kernel machines," in *Proceedings of the 13th International Conference on Neural Information Processing Systems*, MIT press, pp. 661–667.

Willsky, A. and Jones, H. (1976), "A generalized likelihood ratio approach to the detection and estimation of jumps in linear systems," *IEEE Transactions on Automatic Control*, 21, 108–112.

Wilson, A. and Adams, R. (2013), "Gaussian process kernels for pattern discovery and extrapolation," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1067–1075.

Wilson, A., Gilboa, E., Cunningham, J. P., and Nehorai, A. (2014), "Fast kernel learning for multidimensional pattern extrapolation," in *Advances in Neural Information Processing Systems*, pp. 3626–3634.

Wilson, A. and Nickisch, H. (2015), "Kernel interpolation for scalable structured Gaussian processes (KISS-GP)," in *International Conference on Machine Learning*, pp. 1775–1784.

Xue, Z., Wang, Z., and Ettl, M. (2015), "Pricing personalized bundles: A new approach and an empirical study," *Manufacturing & Service Operations Management*, 18, 51–68.

Yang, S., Wang, J., Fan, W., Zhang, X., Wonka, P., and Ye, J. (2013), "An efficient ADMM algorithm for multidimensional anisotropic total variation regularization problems," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, pp. 641–649.

Young, Y. T. and Kuo, L. (2001), "Bayesian binary segmentation procedure for a Poisson process with multiple changepoints," *Journal of Computational and Graphical Statistics*, 10, 772–785.

Yuan, M. and Lin, Y. (2006), "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68, 49–67.

Zhou, X., Shekhar, S., and Ali, R. Y. (2014), "Spatiotemporal change footprint pattern discovery: an inter-disciplinary survey," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4, 1–23.

Zou, H. (2006), "The adaptive lasso and its oracle properties," *Journal of the American Statistical Association*, 101, 1418–1429.