Two-stage Sequential Change Diagnosis Problems

By

XIAOCHUAN MA

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

Electrical and Computer Engineering

in the

OFFICE OF GRADUATE STUDIES

of the

UNIVERSITY OF CALIFORNIA

DAVIS

Approved:

____
LIFENG LAI, Chair

____
ZHI DING

____
KHALED ABDEL-GHAFFAR

Committee in Charge

2023

Xiaochuan Ma, Student ID: 916162692
Feb. 9th 2023
Electrical and Computer Engineering

Abstract

Sequential change-point analysis is a fundamental problem that arises in a variety of fields including network monitoring, power system, climate modeling, finance, image analysis, etc. Based on the sequential observations and additional information about observations, one of the main goals of change-point analysis is to detect the change point as quickly as possible. Beyond detecting the change, discovering what is the post-change status of the system is also important. In this dissertation, we study two sequential change point analysis problems. One is the two-stage sequential change point diagnosis (SCD) problem. The other is the data-driven quickest change point detection (QCD) problem.

In the first part, we study the two-stage SCD problem in the Bayesian setting. In the SCD problem, the data distribution will change at an unknown time, from distribution f_0 to one of the I candidate distributions. We need to detect the change point as quickly as possible and identify the distribution after the change as accurately as possible. In existing work on SCD problems, one must detect the change and identify the distribution after the change at the same time. In practice, however, after we detect the change, we may still have the opportunity to observe extra data samples with low unit cost, which may help us to make a more accurate identification decision. Motivated by this, we formulate a two-stage SCD problem. In this problem, we have two stopping times. The first stopping time is the time to raise an alarm once a change has been detected. After that, we can keep collecting more observations that have a low unit cost. The second stopping time is the time when we are ready to make the identification decision. Therefore, in our problem formulation, change detection and distribution identification become two different stages of the whole SCD procedure. The goal of a two-stage SCD rule is to minimize the total cost including delay, false alarm, and misdiagnosis probabilities. To solve the two-stage SCD problem, we first convert the problem into a two-ordered optimal stopping time problem. Using tools from optimal multiple stopping time theory, we obtain the optimal SCD rule. Moreover, to address the high computational complexity issue of the optimal SCD rule, we further propose a computationally efficient threshold-based two-stage SCD rule. By analyzing the asymptotic behaviors of the delay, false alarm, and misdiagnosis costs, we show that the proposed threshold SCD rule is asymptotically optimal as the per-unit delay costs go to zero. Furthermore, we extend the two-stage SCD problem to a sensor array setting where there is a sensor array with L sensors monitoring the environment. Once a change happens in the environment, the change will propagate across the sensor array gradually. After detecting the change, we are allowed to continue observing more samples so that we can identify the distribution after the change more accurately. Similar to the single sensor case, we characterize the structure of the optimal diagnosis rule. But this rule has considerably high complexity. Therefore, we further propose a threshold rule SCD rule for the multi-sensor setting. In addition, we also prove that this threshold rule is asymptotically optimal as the per-unit delay costs go to zero.

In the second part, we study the Bayesian QCD problem and provide data-driven solutions for this problem. The optimal solutions to the QCD problem under different settings have been extensively studied. Most of these solutions require a priori information about the QCD model and i.i.d. data samples. However, in many real-world applications, these requirements may not be satisfied. In these situations, the optimal QCD rules are not available. This dissertation proposes two data-driven approaches for the online Bayesian QCD problem, including the deep Q-network (DQN) method and the Neural Monte Carlo (NMC) based data-driven change-point detection rule. The NMC-based method is guaranteed to converge. More importantly, these two methods work not only for i.i.d. data samples but also for non-i.i.d. data. Numerical results illustrate that the two proposed methods can detect the change point accurately and timely.

Acknowledgement

This dissertation is not only a record of my research but also a milestone in my life and work at UC Davis. The past 5 years were some of the most enjoyable times in my past life. Looking back over the past 5 years, I can only see happiness and joy in my memory. The only reason for that is there are remarkable individuals who gave me great opportunities, guidance, help, and support.

First of all, I would like to appreciate my Ph.D. supervisors: Prof. Lifeng Lai and Shuguang Cui. Prof. Lai, in particular, has been a fantastic primary supervisor. Under his guidance, he always gave me enough freedom to explore different ideas and provided important suggestions when I met difficulties. I could not finish my research projects without all his help in selecting research topics, suggesting technical solutions, and proofreading my derivations and manuscripts. I am also grateful to Prof. Cui for providing continuous support on my research project.

Besides my supervisors, I would like to thank the rest of my dissertation committee: Prof. Zhi Ding and Prof. Khaled Abdel-Ghaffar, for their insightful comments and encouragement, but also for the questions which incented me to widen my research from various perspectives. I am also thankful to Prof. Hao Chen and Prof. Weijian Yang for their service on my qualifying exam. In addition, I want to thank Prof. Jianhua Zhang, my M.S. advisor. Without her support, I am not sure I would be at UC Davis and complete my Ph.D. study.

I wish to show my appreciation to everyone in my research group, Yulu Jin, Xinyi Ni, Parisa Oftadeh, Puning Zhao, Fuwei Li, Minhui Huang, Xinyang Cao, Guanlin Liu and Chenye Yang, for their great support and technical advice. I would like to thank all my friends in Davis, especially Han Zhang, Songyang Zhang, Zhelun Zhang, Qinwen Deng, Kangning Zhang, Kaiming Fu and Hudson Shih. They made my life at Davis so wonderful. I also wish to extend my special thanks to my friends in the UC Davis Table Tennis Club. I really enjoyed all the time I spent with them in the club.

In the end, I am grateful to my family, especially my parents, Yali Liu and Furong Ma. Without their everlasting love, guidance and support, I would be nothing. Also, a special thank you to my cousin, Fanghui Jia. He is always a role model I want to follow and gave me continuous encouragement along the way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this chapter, we introduce the two-stage change-point diagnosis (SCD) problem and the data-driven quickest change point detection (QCD) problem studied in this dissertation. First, we introduce the general background of the QCD problem and the SCD problem. Then, we discuss the proposed change-point analysis problems and summarize our contributions.

1.1 Backgrounds of Change-Point Analysis Problem

This dissertation focuses on two change-point analysis problems, i.e., the SCD problem and the QCD problem. The SCD problem is an extension of the QCD problem. Therefore, in this section, we first introduce the QCD problem and then describe the SCD problem.

Detecting and analyzing abrupt changes in the statistical behavior of an observed time series is a classical problem. Its provenance dates at least to the work in the 1930s on the problem of monitoring the quality of manufacturing processes [1]. Nowadays, this problem has been applied in a wide variety of fields, including environment and public health [2–5], image analysis [6, 7], finance [8–10], power system [11, 12], medical diagnosis [13–16], navigation [17, 18], network security [19–24], remote sensing [25–27], video editing [28], etc. In many of these applications, such as medical diagnosis and image analysis, the change-point analysis involved follows the off-line style, i.e., analyze the change-point problem given the whole sequence. The most common off-line change-point analysis tasks include identifying the presence of a change, estimating the occurrence time of the change, or further analyzing the stochastic properties of the pre-change and post-change observations, etc.

On the other hand, change analysis applications in remote sensing, power system, and finance mainly focus on the online setting, i.e. detecting the change point in real time as the data samples arrive sequentially. As shown in Fig. 1.1, in a process $\{x_t\}_{t\geq 1}$, the distribution of data samples changes from f_0 to f_1 at an unknown time λ. Concretely, the distribution of data samples is f_0 when $t < \lambda$. Otherwise, the distribution of data samples is f_1 when $t \geq \lambda$. The time we detect the change is τ. In the online change analysis setting, performance metrics such as the probability of the false alarm and the delay between the true change time and the detection time are important. This type of problem is called the quickest change-point detection (QCD) problem. These quantities will be defined rigorously in the sequel. For example in health monitoring or environmental monitoring, the most important task is to detect or analyze the event as soon as possible so that necessary actions can be taken quickly to avoid or reduce losses. However, a detector raising alarms all the time is obviously not desirable, although it can always raise alarms soon after the change. Therefore, the occurrence of false alarms should also be controlled in the online change-point analysis. In most cases, there is a tradeoff between these two metrics. Hence one of the key topics in online change-point detection and analysis problems is how to balance the costs of false alarm and delay.

Under an online setting, the change-point detection and analysis problems can be further classified into two categories based on whether we model the change-point as a random variable or not. If the change-point λ is modeled as a random variable with distribution P_λ, then the problem is a Bayesian change-point detection and analysis problem. If the change-point λ is assumed to be fixed but unknown value, the problem is a non-Bayesian change-point detection and analysis problem.

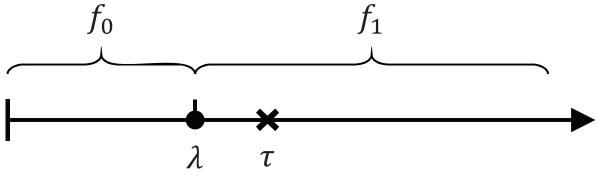


Figure 1.1: Time ordering of a QCD process. The change point is λ and the detection time is τ.

1.1.1 Quickest Change-Point Detection Problem

The Bayesian change-point detection problem is first proposed by Kolmogorov and Shiryaev [29]. In this problem, the distribution of the observations experiences an abrupt change at an unknown time λ. Under the Bayesian setting, λ is assumed to be a random variable with distribution P_λ. A very common choice of P_λ is the Geometric distribution. The Geometric distributed λ is not only mathematically convenient but also reasonable to many real-world applications. In the online change-point detection problem, we care about two costs, false alarm and delay. The goal is to detect the change point quickly and reduce the occurrence of false alarms. Under the Bayesian setting, the metric for false alarm cost is the false alarm probability and the metric for delay cost is the mean detection delay. Therefore, a detection rule which can optimize the trade-off between the false alarm probability and the mean detection delay is desired. Let c denote the unit delay cost, then the cost function for the Bayesian QCD problem can be defined as $C(\tau) = \mathbb{P}(\tau < \lambda) + c\mathbb{E}[(\tau - \lambda)_+]$. The goal of the Bayesian QCD problem is to design a stopping rule that minimizes this cost.

In [30], Lorden proposed a non-Bayesian formulation of the change-point detection problem. In this problem, no prior information about the change-point λ is known. In the non-Bayesian setting, the mean delay is replaced by a worst-case conditional delay, where the conditioning is with respect to the change point, and the worst case is taken over all possible values of the change point and all realizations of the measurements leading up to the change point. The constraint on the occurrence of false alarms is a lower bound on the allowable mean time between false alarms. Therefore, the non-Bayesian online change-point detection problem is to find a detection rule which minimizes the worst-case conditional delay while satisfying the constraint on the mean time between false alarms. Concretely, we require the mean detection delay cost

$$C^*_{delay} = \sup_{\lambda>0} ess \sup \mathbb{E}^{(\lambda)} \left[(\tau - \lambda)_+ | x_1, x_2, \ldots, x_{\lambda-1}\right]$$

should be as small as possible under the following constraint

$$E_0[\tau] \geq \overline{T}$$

which means the mean time between false alarms must be lower bounded. Here $\mathbb{E}^{(\lambda)}$ is the expectation given the change point is λ and $\mathbb{E}_0$ denotes the expectation given no change will happen.

A widely used scheme for the non-Bayesian change-point detection problem is the CUSUM test. For observation $\{x_n\}_{n>0}$, let $L_n = f_1(x_n)/f_0(x_n)$, where f_0 and f_1 are the pre-change and post-change distribution of the observations. Then define $S_n = L_n \max\{1, S_{n-1}\}, S_0 = 0$. The CUSUM detection rule is raising alarm once S_k is larger than a specific threshold.

Recently, there are many extensions of the two basic setups described above [31–40]. [31] proposed an asymptotically optimal solution for the non-Bayesian QCD problem in a multi-stream setting. [32] and [33] applied nearest neighbor methods to solve the non-Bayesian QCD problem and achieved good performance on high dimensional and non-Euclidean data. The Bayesian QCD problem with unknown post-change distribution is studied in [34]. In both Bayesian and non-Bayesian setups, [35] studied the QCD problem where the underlying linear model of the data changes at an unknown time. The non-stationary change point, in which the statistical behavior after the change is non-stationary, is analyzed in [36, 37]. With an additional constraint on the average number of observations taken before the change, the data efficient QCD problem is investigated in [38–40]. Of particular relevance to this dissertation is many recent works on QCD problems in multi-sensor setting [41–46].

[41] considers a temporal diffusion network model to capture the temporal dynamic structure of multiple changepoints. In [42], the problem of sequentially detecting a moving anomaly is studied, in which the anomaly affects different parts of a sensor network over time. [43] studies the Bayesian change analysis problem in a linear sensor array where the change can first happen in any sensor and then propagate to the neighboring sensors. The goal is not only to detect the change quickly but also to identify the sensor that the change pattern first reaches. [44] provides the optimal solution of a Bayesian distributed QCD problem under a quasi-classical information structure. [45] studies the QCD problem in a sensor array where the communication bandwidth is limited. The authors consider the problem under both Bayesian and non-Bayesian settings and developed asymptotically optimal solutions for two specific scenarios. In [46], the authors assume the change propagates across the sensors, and the propagation can be modeled as a Markov process. With perfect information about the observations and a priori knowledge of the statistics of the change process, the authors proposed a dynamic-programming based optimal solution of the Bayesian QCD. Faced with the high computation complexity of the optimal solution, the authors also proposed a low-complexity threshold QCD rule which is asymptotically optimal when the false alarm probability regime goes to zero. In Chapter 3 of this dissertation, we will consider the more challenging SCD problem with more general change-point propagation models.

1.1.2 Sequential Change Diagnosis Problem

Sequential change diagnosis (SCD) problem is the joint problem of online detection of a sudden change in the distribution of a random sequence and identification of the post-change distribution. In particular, the SCD problem can be viewed as a combination of the quickest change-point detection (QCD) problem and sequential multiple hypothesis testing (SMHT) problem. In QCD problems, the goal is to detect the presence of change in the distribution quickly [35, 37, 47–57]. In SMHT problems, the distribution does not change. The focus is to identify the data distribution from I candidate distributions [58–63]. In the SCD problem, as shown in Fig. 1.2, the data distribution will change at an unknown time λ, from distribution f_0 to one of the I candidate distributions, f_θ. The post-change state $\theta \in \{1, 2, \ldots, I\}$. We need to detect the change point λ timely and identify the post-change distribution f_θ accurately. If the change detection and identification must happen at the same time, the SCD problem is called the one-stage SCD problem. A one-stage SCD rule includes a stopping time τ and a decision d. In [64] the SCD problem is proposed for the first time. A criterion of optimality for the non-Bayesian SCD problem is formulated in [49] to minimize the mean detection/isolation delay while guaranteeing the mean time before a false alarm or false isolation is at some acceptable level. With this criteria, [64] gives an asymptotic lower bound of the mean detection/isolation delay. [65] generalizes earlier work on SCD and provides more tractable and appropriate performance criteria for both Bayesian and non-Bayesian cases. Afterward, the optimal solution and asymptotically optimal solution of the Bayesian SCD problem are derived in [66] and [67], respectively. The criteria of optimality of the Bayesian SCD problem is to minimize the Bayesian cost: $C(\tau) = \mathbb{P}(\tau < \lambda) + \mathbb{P}(d \neq \theta) + c\mathbb{E}[(\tau - \lambda)_+]$, where τ is the time when the change detection rule declare a change has happened and c is the unit delay cost. The expectation here is with respect to the distribution of λ. In [29] and [30], the a priori information about the SCD process including the prior distribution of the change point, the post-change distribution, the prior probability of all post-change states, and all possible post-change distributions are known. Therefore, the posterior probabilities can be updated at every time step. After proving the relationship between the Bayesian cost and the posterior probabilities, [28] gives the optimal solution of the SCD problem: when the posterior probability vector enters some particular region of the probability vector space, we should raise an alarm and choose the post-change state corresponding to the highest posterior probability. However, to obtain the decision region, dynamic programming should be repeated until convergence. This leads to the high complexity of the optimal solution. To overcome this issue, [30] proposed a threshold-based SCD rule which can be implemented easily. The threshold-based SCD rule

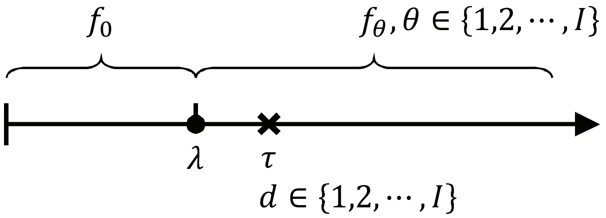


Figure 1.2: Time ordering of a one-stage SCD process. The change point is λ and the detection time is τ. d is the identification decision.

is also proved to be asymptotically optimal as $c \to 0$.

1.2 Two-stage SCD Problem

1.2.1 Two-stage SCD Problem in Single Sensor Case

All SCD problems in the literature are one-stage SCD problems, i.e., the change and identification of the change must happen at the same time. In practice, however, after we detect the change, we may still have the opportunity to observe extra data samples with low unit cost, which may help us to make a more accurate identification decision. For example, in the structural health monitoring (SHM) system [68] of a building, sensors are used to monitor the material or geometric properties of the building. When sudden damage happens to the building, the SHM system should detect the damage quickly and identify the type of damage accurately. Typically, identifying the type of damage requires more data than detecting the damage. In other words, we need more time to collect enough data for damage identification than damage detection. However, the detection task is very urgent because people in a damaged building can be in great danger. Therefore, in a good SHM system, the identification of the damage should be allowed to be completed after the damage detection. In this way, the people can be evacuated from the building immediately once the damage is detected. After that, the SHM system can keep collecting more data and make an accurate damage identification. In the detection stage, the unit delay cost is high because even one time unit of delay can cost lives. On the other hand, the unit delay cost in the identification

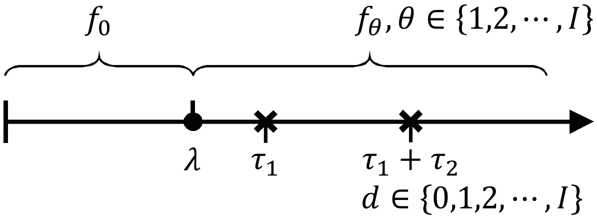

f_0
$f_\theta, \theta \in \{1,2,\cdots,I\}$
λ
τ_1
$\tau_1 + \tau_2$
$d \in \{0,1,2,\cdots,I\}$

Figure 1.3: Time ordering of a two-stage SCD process. The change point is λ. The detection time is τ_1 and the identification time is $\tau_1 + \tau_2$. d is the identification decision.

stage is much lower than in the detection stage since people have already been evacuated. As another example, a factory conducts quality tests on a manufacturing process that includes multiple processing components. When a sudden fault occurs in one of the processing components, quality testers need to detect the fault quickly and identify the faulted processing component accurately. In many cases, the identification task needs more product samples than the detection task. However, the detection task is more urgent than the identification task, since the potential loss (compensation for damages, product recall, or damage to the brand, etc.) could be very high if the faulted products go to market, especially for the products related to people's life and property security. Therefore, a smart quality testing system should allow fault detection to happen earlier than fault identification. In this case, the product samples produced after the detection will only be used for fault identification and will not be sold. The quality testers can use more samples to make an accurate fault identification. Although the factory still needs to pay the production cost of the samples produced after the detection, these samples can be used to identify the fault and will not cause any further potential costs. Therefore, the unit cost of the identification stage is lower than the detection stage. In addition to these two examples, this two-stage situation exists in many real-world applications, such as diagnosis of intrusions in computer networks [69], navigation system integrity monitoring [70] etc.

Motivated by this, we formulate a two-stage Bayesian SCD problem. In this problem, we have two stopping times, τ_1 and $\tau_1 + \tau_2$, as shown in Fig. 1.3. The first stopping time τ_1 is the time to raise an alarm once a change has been detected. After that, we can keep collecting more observations that have a low unit cost. The second stopping time $\tau_1 + \tau_2$ is the time when we are ready to make the identification decision. Therefore, in our problem formulation, change detection and distribution identification become two different stages of the whole SCD procedure. By taking advantage of low-cost samples after the change is detected, it is possible to improve the identification accuracy and hence achieve a lower total cost. It's worth noting that the detection and identification stages are not independent, as the end state of the detection stage is the start state of the identification stage. Hence the proposed problem is not a simple combination of a QCD problem and an SMHT problem. In this dissertation, we study the two-stage Bayesian SCD problem and extend it to a multi-sensor array setting.

The goal of a two-stage SCD rule is to minimize the total cost including delay, false alarm, and misdiagnosis probabilities. To solve the two-stage SCD problem, we first analyze the posterior probability at each time step. Based on the prior distribution of the change point, the post-change distribution, the prior probability of all post-change states, and all possible post-change distributions, we derive the update rule of the posterior probability of the change point and the post-change state. Then we express the mean delay, the false alarm, and the misdiagnosis probability with the posterior probability. [71] showed that the ordered multiple stopping time problem can be reduced to a sequence of optimal single stopping time problems defined by backward induction. Therefore, we use the same method and reduce the two-stage stopping problem to two optimal single stopping time problems. Then two Bellman equations can be acquired for the detection and identification stages, respectively. By applying dynamic programming to solve the two Bellman equations, we can obtain the optimal identification region and the optimal detection region on the posterior probability vector space. The optimal SCD rule is: (1) Raise an alarm when the posterior probability vector enters the optimal detection region in the probability vector space; (2) Make an identification decision when the posterior probability vector enters any of the optimal identification regions, and the identification decision is the post-change state corresponding to that identification region is entered.

Since the optimal rule should be obtained using dynamic programming, the complexity of the optimal solution of the two-stage SCD problem is high. To address the high computational complexity issue of the optimal SCD rule, we further propose a computationally efficient threshold-based two-stage SCD rule. The threshold-based SCD rule is:(1) Raise an alarm when the posterior probability of change has happened and reaches a selected threshold; (2) Make identification decision when the posterior probability of any post-change state reaches a selected threshold and that post-change state is the identification decision. By analyzing the asymptotic behaviors of the delay, false alarm, and misdiagnosis costs as the unit delay costs go to zero, we show that these three costs can be approximated using the thresholds in the threshold SCD rule. By expressing the total Bayesian cost with these thresholds and minimizing the cost, we can find the formulas of the optimal thresholds. Finally, we also proved that the proposed threshold SCD rule is asymptotically optimal as the per-unit delay costs go to zero. The results obtained in this study have been published in [72, 73].

1.3 Two-stage SCD Problem in Multi-sensor Array

To further improve the performance of the SCD rule, there are growing interests in distributed decision-making systems. In the multi-sensor setting, the information collected by multiple sensors is sent to the fusion center, where the detection decision is made. Therefore, the multi-sensor case can improve the performance when compared with the single-sensor case. There are many applications of distributed decision-making systems in the real world, e.g. (1) Structural health monitoring system of buildings, bridges or highway networks [74]; (2) intrusion detection in computer networks and security systems [19, 23]; (3) monitoring catastrophic faults to critical infrastructures such as water and gas pipelines, power systems, supply chains, etc.; (4) wireless resource access and allocation problems [75]; (5) Seismic monitoring and detection [76]. There are existing works that studied the change-point detection problem in multi-sensor setting [34, 43, 46, 49, 77–79]. In this dissertation, we also consider a two-stage SCD problem in the multi-sensor scenario. For this problem, we characterize the structure of the optimal diagnosis rule. The optimal stopping rule is obtained by converting the two-stage SCD problem into a two-ordered optimal stopping time problem, which can be solved using dynamic programming (DP). However, the dimension of the state space grows exponentially with the number of sensors and candidate post-change distributions. Thus the complexity to implement the DP solution is extremely high. To address this issue, we propose a low complexity threshold SCD rule. Furthermore, we analyze the performance of the proposed multi-sensor threshold SCD rule in two different linear array cases (Fig. 1.4) depending on whether the sensor first affected by the change is known or not. Concretely, for the general case in which the sensor first being affected by the change is randomly chosen and unknown, we prove the threshold rule is asymptotically optimal under some technical conditions. On the other hand, for the special case in which the sensor first affected by the change is fixed and known, we prove that the threshold rule is asymptotically optimal without additional technical conditions. Moreover, we extend the low-cost SCD rule to a more general 2D sensor array. In this 2D sensor array case, the change can happen to any sensor and then gradually propagate to the surrounding sensors. For this 2D sensor array case, we also prove the asymptotic optimality of the multi-sensor SCD rule. In addition, we investigate how increasing the number of sensors can improve the asymptotic performance of the multi-sensor threshold SCD rule. The results obtained in this study have been published in [80, 81].

1.4 Data driven QCD problem

As introduced in Section 1.1, the change-point analysis problem was extensively studied and many powerful methods have been proposed for different problem settings. However, two limitations make these methods hard to be used in many real-world applications. First, full

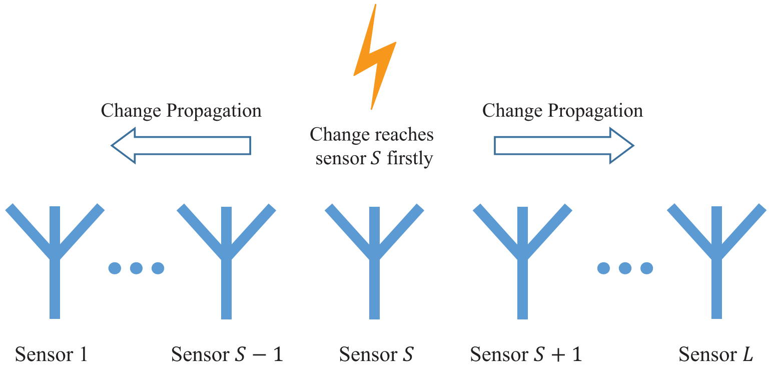


Figure 1.4: Change propagation model

knowledge of the sample distributions and their latent statistical structure is required to run these methods. Although there are some works that assume the distributions are unknown, they still assume the statistical structure of the observations is known. For example, in [82], the distributions of the data samples are unknown but they are assumed to belong to the multivariate exponential family. However, in many real-world applications, this knowledge about the latent stochastic QCD process is also unknown. A common situation is that the only given information is the historical ground truth data. Secondly, many of the existing methods assume the observed data samples are independent and identically distributed (i.i.d.), which is not always true in many real-world applications.

As a promising approach to address these issues, the data-driven QCD method becomes more and more of interest in recent years. At the same time, machine learning provides many efficient algorithms to solve data-driven problems. Therefore, machine learning algorithms have already been applied to QCD problems. For example, [32, 83–86] propose different data-driven algorithms for non-Bayesian QCD problems. On the other hand, as the Bayesian QCD process can be viewed as a partially observable Markov decision process (POMDP), reinforcement learning-based methods can also be applied to solve the data-driven Bayesian QCD problems. In [87], the tabular Q-learning is applied to solve the change-point detection in the power system. This paper regards the Bayesian cost as a negative reward of a POMDP and uses reinforcement learning methods to maximize the total reward. However, [87] still requires knowledge of the pre-change distribution, and the performance of the tabular Q-learning method will be degraded when the state space is large.

In this dissertation, we propose two data-driven approaches for the Bayesian QCD problem. In both two approaches, instead of using quantization to convert the problem to a learning problem with a finite number of states, we use the function approximation approach to directly address the issue related to continuous states.

In the first approach, we apply the deep Q-network (DQN) to solve the online Bayesian QCD problem. In the QCD problem, one needs to take one of two possible actions, keep observing more data or raise an alarm, at each time step. The distribution of the data samples is determined by the hidden change state. Therefore, in this paper, we formulate the online Bayesian QCD process as a partially observable Markov decision process (POMDP). There are data-driven methods in reinforcement learning which can efficiently solve the POMDP problem and some of them have already been used in the QCD problem. In this dissertation, instead of using quantization to convert the problem to a learning problem with a finite number of states, we use the function approximation approach to directly address the issue related to continuous states. In particular, we apply the deep Q-network (DQN) to solve the online Bayesian QCD problem. The DQN approximates the Q-value of continuous input using a neural network. After training with the historical data, the Q-network can approximate the Q-values for the two actions. Based on the Q-value approximations, one can determine if it is time to declare a change without knowing the a priori information of the QCD process. Numerical results show that, after training with a reasonable amount of historical data, the proposed DQN-based detection rule can achieve good performance for online Bayesian QCD problems under different settings.

In the QCD problem, one needs to make a decision at each time step based on the data samples observed so far. The distribution of the data samples is determined by the hidden change state. Therefore, in this paper, we formulate the online Bayesian QCD process as a partially observable Markov decision process (POMDP). There are data-driven methods in reinforcement learning which can efficiently solve the POMDP problem and some of them have already been used in the QCD problem. In [87], the authors applied the tabular Q-learning to change-point detection in the power system. At any given time, [87] used data samples within a certain window as the state. As the data samples are continuous values, to apply the tabular Q-learning, the data samples in [87] are quantized to finite discrete levels. However, in a common online Bayesian QCD problem, quantizing the data may compromise the accuracy of the QCD rule, especially when the window size is large so that one has to limit the number of quantization levels. In this paper, instead of using quantization to convert the problem to a learning problem with a finite number of states, we use the function approximation approach to directly address the issue related to continuous states. In particular, we apply the deep Q-network (DQN) to solve the online Bayesian QCD problem. The DQN approximates the Q-value of continuous input using a neural network. After training with the historical data, the Q-network can approximate the Q-values for the two actions. Based on the Q-value approximations, one can determine if it is time to declare a change without knowing the a priori information of the QCD process. Numerical results show that, after training with a reasonable amount of historical data, the proposed DQN-based detection rule can achieve good performance for online Bayesian QCD problems under different settings.

As for the second approach, we propose a Neural Monte Carlo (NMC) based method to solve the data-driven Bayesian QCD problem. In the optimal solution of the Bayesian QCD problem for i.i.d. data samples, an alarm will be raised once the posterior false alarm probability is lower than a threshold. On the other hand, the posterior false alarm probability can be regarded as a value function and learned from the historical data set. Concretely, at any time t we get a reward if we raise an alarm. If the change happens after t, the reward is 1; otherwise, the reward is 0. In this case, the false alarm probability at time t is equivalent to the value function of raising an alarm at time t. Inspired by these facts, we propose a reinforcement learning based method to solve the data-driven Bayesian QCD problem. First, we apply a randomized neural network to approximate the posterior false alarm probability. This neural network takes the historical data observations as input and outputs the approximation of the posterior false alarm probability. Since the posterior false alarm probability is mainly determined by the recently collected data samples rather than earlier data samples, the input of the neural network is the data samples within the most recent sliding window. With this neural network, the posterior false alarm probability can be monitored as new data samples come up. In particular, all the weights in this neural network except the linear output layer are untrainable. Therefore, training with the Gradient Monte Carlo algorithm [88], the neural network is guaranteed to converge. Afterward, following the idea of the optimal solution of the Bayesian QCD problem for i.i.d. data samples, the proposed NMC-based QCD rule raises an alarm once the approximation of the posterior false alarm probability meets a given threshold. The optimal threshold is chosen based on the performance on the validation data set. Besides, as a solution to the data-driven QCD problem, this method does not require prior knowledge about pre-change and post-change distributions. The only assumption is that the change point is a geometric random variable, which is satisfied in many real-world phenomena, such as failure times. More importantly, the proposed NMC-base QCD rule also works for non-i.i.d. data samples. The observation model of the different non-i.i.d. QCD problems could be different. Data sequences generated by a hidden Markov model (HMM) is a special case of non-i.i.d. data. In this dissertation, we take the HMM QCD problem as an example of non-i.i.d. QCD problems to explain how to apply the NMC-base QCD rule to non-i.i.d. observation data. Finally, numerical experiments are carried out and the results show that this NMC-based QCD rule has good performance in different Bayesian QCD problem settings.

The results obtained in this part have been published in [89, 90].

Chapter 2

Two-Stage Bayesian Sequential Change Diagnosis

In this chapter, we focus on the single sensor two-stage Bayesian SCD problem. Firstly, we provide our problem formulation and study the evolution of the posterior probability, and convert the two-stage SCD problem into two optimal single stopping time problems. Then we derive the optimal rules for the two optimal single stopping time problems. After that, we introduce the threshold two-stage SCD rule and prove the asymptotic optimality of the threshold two-stage SCD rule. Finally, simulation results are provided to illustrate the performance of the two proposed SCD rules.

2.1 Problem Formulation

Consider a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ that hosts a stochastic process $\{X_n\}_{n\geq 1}$. The range of X_n is $\mathcal{X}$. Let $\lambda : \Omega \to \{0, 1, \ldots\}$ be the time when the distribution of X_n changes and $\theta : \Omega \to \mathcal{I} \triangleq \{1, \ldots, I\}$ be the state after change. The state after θ change corresponds to one post-change distribution f_θ. We also denote $\mathcal{I}_0 = \mathcal{I} \cup \{0\}$. In particular, the distribution of X_n is f_0 when $n < \lambda$, and is f_θ when $n \geq \lambda$. λ and θ are independent random variables defined with the distributions

$$\mathbb{P}\{\lambda = t\} = \begin{cases} \rho_0, & \text{if} \quad t = 0 \\ (1-\rho_0)(1-\rho)^{t-1}\rho, & \text{if} \quad t \neq 0 \end{cases}$$

and $v_i = \mathbb{P}\{\theta = i\} > 0,\ i \in \mathcal{I}$. Here, $\rho_0, \rho \in (0,1)$, $v_i \in [0,1]$ and $\sum_{i\in\mathcal{I}} v_i = 1$. They are given constants. Given λ and θ, random variables $\{X_n\}_{n\geq 1}$ are independent. In addition, $\mathbb{F} = (\mathcal{F}_n)_{n\geq 0}$ is the filtration generated by the stochastic process $\{X_n\}_{n\geq 1}$; namely, $\mathcal{F}_0 = \{\emptyset, \Omega\}$ and $\mathcal{F}_n = \sigma(X_1, X_2 \dots X_n) = \{\emptyset, \{X_1, X_2 \dots X_n\} \times \{\mathcal{X}\}^\infty\}$. To simplify the notation, we express the conditional probabilities as:

$$\begin{cases} \mathbb{P}_i\{\cdot\} = \mathbb{P}\{\cdot|\theta = i\}, \\ \mathbb{P}_i^{(t)}\{\cdot\} = \mathbb{P}\{\cdot|\theta = i, \lambda = t\}, t \geq 0. \end{cases}$$

Correspondingly, $\mathbb{E}_i$ and $\mathbb{E}_i^{(t)}$ are the expectations under $\mathbb{P}_i$ and $\mathbb{P}_i^{(t)}$.

Our goal is to quickly raise an alarm when the change occurs and further accurately identify the state θ. Towards this goal, we employ a two-stage SCD rule $\delta = (\tau_1, \tau_2, d)$ that includes two stopping times τ_1 and $\tau_1 + \tau_2$ and a decision rule d. Here, τ_1 is the time when we raise an alarm that a change has occurred. In our model, after τ_1, we can keep collecting more low-cost observations to make a more accurate identification. Correspondingly, $\tau_1 + \tau_2$ is the time when we make the identification decision d.

Let $\Delta := \{(\tau_1, \tau_2, d) | \tau_1, \tau_1 + \tau_2 \in \mathbb{F}, \tau_2 \geq 0, d \in \mathcal{I}_0\}$ be the set of all possible two-stage SCD rules. Here, $\tau \in \mathbb{F}$ means that τ is a stopping time associated with $\mathbb{F}$. The time ordering of a two-stage SCD process is shown in Fig. 1.3. We should note that if a wrong decision is made at τ_1, i.e., $\tau_1 < \lambda$, then $d = 0$ is the correct identification as long as this identification is made before λ, i.e., $\tau_1 + \tau_2 < \lambda$.

The possible costs of an SCD rule include costs of delay, false alarm and misdiagnosis. The delay consists of two parts, $(\tau_1 - \lambda)_+$ and τ_2, which correspond to the change detection stage and the distribution identification stage respectively. The expected costs of them are $\mathbb{E}[c_1(\tau_1 - \lambda)_+]$ and $\mathbb{E}[c_2\tau_2]$, where c_1 and c_2 are per-unit delay costs associated with each stage. We assume that the ratio between c_1 and c_2 is a constant $r = c_1/c_2$. A false alarm occurs when a change alarm is raised before λ. The expected false alarm cost is $\mathbb{E}[a\mathbf{1}_{\{\tau_1<\lambda\}}]$, where a is the penalty factor of false alarm and $\mathbf{1}_{\{.\}}$ is the indicator function.

Misdiagnosis happens when a wrong distribution identification is made, i.e., $d \neq \theta$. The expected misdiagnosis cost is

$$\mathbb{E}\Big[\sum_{i\in\mathcal{I}} b_{ij}\mathbf{1}_{\{\infty>\tau_1+\tau_2>\lambda,\theta=i,d=j\}} + b_{0j}\mathbf{1}_{\{\tau_1+\tau_2<\lambda,d=j\}}\Big]$$

for $d = j$, where b_{ij} is the penalty factor for wrong decision $d = j$ when $\theta = i$ and $b_{0,j}$ is the penalty factor of the false alarm of the distribution identification stage. We set $b_{ij} = 0$ when $i = j$. Thus the Bayesian cost function for a two-stage SCD rule $\delta \in \Delta$ is

$$\begin{aligned} C(\delta) = c_1\mathbb{E}\left[(\tau_1 - \lambda)_+\right] + c_2\mathbb{E}[\tau_2] + a\mathbb{E}[\mathbf{1}_{\{\tau_1<\lambda\}}]+ \\ \sum_{j=0}^{I}\mathbb{E}\Big[\sum_{i=1}^{I} b_{ij}\mathbf{1}_{\{\infty>\tau_1+\tau_2>\lambda,\theta=i,d=j\}} + b_{0j}\mathbf{1}_{\{\tau_1+\tau_2<\lambda,d=j\}}\Big]. \end{aligned} \quad (2.1)$$

In a closely related one-stage SCD problem discussed in [66] and [67], the change detection and distribution identification must occur at the same time, and hence there is only one stopping time. We generalize the problem setup in [66] by allowing identification to occur later than change detection, with the hope of improving the decision accuracy using the extra samples with lower cost. If $c_1 \leq c_2$, there is no low cost samples and the two-stage SCD rule will become the one-stage rule in [66]. Therefore, in this proposal, we assume $c_1 > c_2$. Under this condition, we can improve the identification accuracy with a low delay cost in the distribution identification stage.

2.2 Posterior Analysis

Let $\Pi_n = (\Pi_n^{(0)}, \ldots, \Pi_n^{(I)})_{n\geq 0} \in \mathcal{Z}$ be the posterior probability process defined as

$$\begin{cases} \Pi_n^{(i)} := \mathbb{P}\{\lambda \leq n, \theta = i|\mathcal{F}_n\}, i \in \mathcal{I}, \\ \Pi_n^{(0)} := \mathbb{P}\{\lambda > n|\mathcal{F}_n\}, \end{cases}$$

where $\mathcal{Z} \triangleq \{\Pi \in [0,1]^{I+1}| \sum_{i\in\mathcal{I}_0} \Pi^{(i)} = 1\}$.

It is easy to check that $\{\Pi_n\}_{n\geq 0}$ is a Markov process satisfying

$$\Pi_n^{(i)} = \frac{D_i(\Pi_{n-1}, X_n)}{\sum_{j\in\mathcal{I}_0} D_j(\Pi_{n-1}, X_n)} \tag{2.2}$$

where

$$D_i(\Pi, x) := \begin{cases} (1-\rho)\Pi^{(0)} f_0(x) & i = 0 \\ (\Pi^{(i)} + \Pi^{(0)}\rho v_i) f_i(x) & i \in \mathcal{I}. \end{cases}$$

The initial state, Π_0, is set as $\Pi_0^{(0)} = 1 - \rho_0$ and $\Pi_0^{(i)} = \rho_0 v_i$ for $i \in \mathcal{I}$. In addition, We have the following assumption on these distributions.

Assumption 2.1. For every $i \in \mathcal{I}_0$ and $j \in \mathcal{I}_0 \backslash \{i\}$, we have

(i) $0 < f_i(x)/f_j(x) < \infty$ a.s.;

(ii) $\int_{\{x: f_i(x)\neq f_j(x)\}} f_i(x)(dx) > 0$.

Assumption 2.1 implies $0 < \Pi_n^{(i)} < 1$ for every finite $n \geq 1$ and $i \in \mathcal{I}_0$. The log-likelihood-ratio (LLR) processes are defined as

$$\Lambda_n(i,j) := \log \frac{\Pi_n^{(i)}}{\Pi_n^{(j)}}. \tag{2.3}$$

Proposition 2.1. With Π_n, we can express (2.1) as

$$C(\delta) = \mathbb{E}\left[\sum_{n=0}^{\tau_1 - 1} c_1 \left(1 - \Pi_n^{(0)}\right) + c_2\tau_2 + \mathbf{1}_{\{\tau_1<\infty\}} a \Pi_{\tau_1}^{(0)} + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}} \sum_{j=0}^{I} \mathbf{1}_{\{d=j\}} \sum_{i=0}^{I} b_{ij} \Pi_{\tau_1+\tau_2}^{(i)}\right].$$

Proof. Since $\{\tau_1 > n\} \in \mathcal{F}_n$ for every $n \geq 0$, then

$$\mathbb{E}[(\tau_1 - \lambda)_+] = \mathbb{E}\left[\sum_{n=0}^{\infty} \mathbf{1}_{\{\lambda \leq n < \tau_1\}}\right] = \sum_{n=0}^{\infty} \mathbb{E}\left[\mathbf{1}_{\{n<\tau_1\}} \mathbb{P}(\lambda \leq n | \mathcal{F}_n)\right] = \mathbb{E}\left[\sum_{n=0}^{\tau_1 - 1} (1 - \Pi_n^{(0)})\right].$$

Next, since $\{\tau_1 = n\} \in \mathcal{F}_n$,

$$\begin{aligned} \mathbb{E}\left[\mathbf{1}_{\{\tau_1<\lambda\}}\right] &= \sum_{n=0}^{\infty} \mathbb{E}\left[\mathbf{1}_{\{n<\lambda\}} \mathbf{1}_{\{\tau_1=n\}}\right] = \sum_{n=0}^{\infty} \mathbb{E}\left[\Pi_n^{(0)} \mathbf{1}_{\{\tau_1=n\}}\right] \\ = \lim_{N\to\infty} \mathbb{E}\left[\sum_{n=0}^{N} \Pi_n^{(0)} \mathbf{1}_{\{\tau_1=n\}}\right] &= \lim_{N\to\infty} \mathbb{E}\left[\Pi_{\tau_1}^{(0)} \mathbf{1}_{\{\tau_1 \leq N\}}\right] = \mathbb{E}\left[\Pi_{\tau_1}^{(0)} \mathbf{1}_{\{\tau_1<\infty\}}\right] \end{aligned}$$

because of the monotone convergence theorem and that $\lim_{N\to\infty} \mathbf{1}_{\{\tau_1 \leq N\}} = \cup_{n=1}^{\infty} \mathbf{1}_{\{\tau_1 \leq n\}} = \mathbf{1}_{\{\tau_1<\infty\}}$. Here, we do not consider the case $\tau_1 = \infty$ as $\Pi_{\infty}^{(0)} = 0$.

Similar to the derivation of $\mathbb{E}\left[\mathbf{1}_{\{\tau_1<\lambda\}}\right]$, for any $j \in \mathcal{I}$,

$$\mathbb{E}\left[\mathbf{1}_{\{\tau_1+\tau_2<\lambda, d=j\}}\right] = \mathbb{E}\left[\Pi_{\tau_1+\tau_2}^{(0)} \mathbf{1}_{\{\tau_1+\tau_2<\infty, d=j\}}\right].$$

Similarly, for any $i \in \mathcal{I}$ and $j \in \mathcal{I} \cup \{0\}$,

$$\mathbb{E}\left[\mathbf{1}_{\{\theta=i, d=j, \lambda \leq \tau_1+\tau_2<\infty\}}\right] = \mathbb{E}\left[\mathbf{1}_{\{\tau_1+\tau_2<\infty, d=j\}} \Pi_{\tau_1+\tau_2}^{(i)}\right].$$

Plugging these four expressions in equation (2.1) completes the proof. □

Define $B_j(\Pi) = \sum_{i\in\mathcal{I}_0} \Pi^{(i)} b_{ij}$, which is the misdiagnosis cost associated with the decision $d = j$. We have

$$
\begin{aligned}
C(\delta) &= \mathbb{E}\left[\sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right) + c_2\tau_2 + \mathbf{1}_{\{\tau_1<\infty\}} a\Pi_{\tau_1}^{(0)} + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}} \sum_{j\in\mathcal{I}_0} \mathbf{1}_{\{d=j\}} B_j(\Pi_{\tau_1+\tau_2})\right] \\
&\geq \mathbb{E}\left[\underbrace{\sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right) + \mathbf{1}_{\{\tau_1<\infty\}} a\Pi_{\tau_1}^{(0)}}_{\text{part 1}} + \underbrace{c_2\tau_2 + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}} B\left(\Pi_{\tau_1+\tau_2}\right)}_{\text{part 2}}\right] \qquad (2.4) \\
&= C(\tau_1, \tau_2, d^*),
\end{aligned}
$$

where $B(\Pi) = \min_{j\in\mathcal{I}_0} B_j(\Pi)$, the smallest misdiagnosis cost. From (2.4), we can see that the optimal decision d^* is the choice that achieves $B(\Pi)$. Then we only need to find the optimal stopping times τ_1 and τ_2, which means that the SCD problem becomes an optimal ordered two-stopping problem. [71] showed that the ordered multiple stopping time problem can be reduced to a sequence of optimal single stopping time problems defined by backward induction. Here we use the same method and reduce the two-stage stopping problem to two optimal single stopping time problems. According to (2.4), the total cost can be divided into two parts. The first part is the expected cost of the change detection stage, and the second part corresponds to the distribution identification stage. The first part depends on τ_1 while the second part depends both on τ_1 and τ_2. We write the cost functions of the change detection stage and distribution identification stage as

$$
C_1(\tau_1) = \sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right) + \mathbf{1}_{\{\tau_1<\infty\}} a\Pi_{\tau_1}^{(0)}
$$

and

$$
C_2(\Pi_{\tau_1}, \tau_2) = c_2\tau_2 + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}} B\left(\Pi_{\tau_1+\tau_2}\right).
$$

C_2 is a function of Π_{τ_1} and τ_2 because Π_{τ_1} and the observations from τ_1 to $\tau_1+\tau_2$ are sufficient to calculate $\Pi_{\tau_1+\tau_2}$. Then we have the minimal expected cost for the SCD process,

$$\begin{aligned} C(\tau_1^*, \tau_2^*, d^*) &= \min_{\tau_1, \tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\left[C_1(\tau_1) + C_2(\Pi_{\tau_1}, \tau_2)\right] = \min_{\tau_1, \tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\Big[C_1(\tau_1) + \mathbb{E}\left[C_2(\tau_2)|\Pi_{\tau_1}\right]\Big] \\ &= \min_{\tau_1 \in \mathbb{F}} \mathbb{E}\left[C_1(\tau_1) + \min_{\tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\left[C_2(\tau_2)|\Pi_{\tau_1}\right]\right]. \end{aligned} \tag{2.5}$$

By (2.5), the two-stage stopping time problem becomes two optimal single stopping time problems. The first one is for the identification stage, its goal is finding the optimal τ_2 which minimizes $\mathbb{E}[C_2(\tau_2)|\Pi_{\tau_1}]$ for any given τ_1 and Π_{τ_1}. The second single stopping time problem is to find the best stopping rule for the detection stage, i.e., selecting the optimal τ_1 to minimize the expected cost of the whole SCD process, $C(\tau_1, \tau_2, d^*)$. From the last line of (2.5), it is easy to see that we can find an optimal τ_1 to minimize the expected cost for the whole SCD process if the optimal rule for τ_2 is known. Therefore, we will solve the SCD problem in a reversed order, i.e., find the optimal rule for the identification stage first, then select the optimal stopping time for the detection stage.

2.3 Optimal Solution

In this section, we characterize the optimal solution to the two-stage SCD problem. We will first focus on the finite-horizon case, and then extend the solution to the infinite-horizon case.

To solve the two-stage SCD problem, we first restrict attention to the finite-horizon case. In particular, in the finite-horizon case, we can spend at most T_1 amount of time in the detection stage, i.e., $\tau_1 \leq T_1$, and we can spend at most T_2 amount of time in the identification stage, i.e., $\tau_2 \leq T_2$. Here, T_1 and T_2 are fixed positive integers.

We first consider the distribution identification stage. In this stage, τ_1 and Π_{τ_1} are already known. After we get the optimal τ_2^* and minimum expected cost, $C_2(\Pi_{\tau_1}, \tau_2^*)$ for any τ_1 and Π_{τ_1}, we will further introduce the optimal stopping rule for the change detection stage.

Now we consider the optimal single stopping time problem under a DP framework. Let $S_n^{(2)}$ denote the state of the system at time $n \in [\tau_1, T_2 + \tau_1]$. $S_n^{(2)}$ can take $\theta \in \mathcal{I}$, 0 and E (End). Here, $S_n^{(2)} = \theta$ means that the change has happened before n and the distribution after the change is f_θ. $S_n^{(2)} = 0$ means that no change has happened before n, which implies a false alarm was made at time τ_1. Once the result of distribution identification is declared, the state of system becomes E. The state evolves as $S_n^{(2)} = g_2(S_{n-1}^{(2)}, \lambda, \mathbf{1}_{\{\tau_1+\tau_2 \leq n\}})$. Here the transition function g_2 is

$$g_2(s, \lambda, \mathbf{1}_{\{\tau_1+\tau_2\leq n\}}) = \begin{cases} 0, & \text{if } \lambda > n, s \neq E, \tau_1 + \tau_2 > n, \\ \theta, & \text{if } \lambda \leq n, s \neq E, \tau_1 + \tau_2 > n, \\ E, & \text{if } s = E \text{ or } \tau_1 + \tau_2 \leq n. \end{cases}$$

The initial state $S_{\tau_1}^{(2)} = 0$ if $\lambda > \tau_1$, otherwise $S_{\tau_1}^{(2)} = \theta$. In addition, the observations in this DP framework are the data samples $\{X_n\}_{n\geq 1}$.

Under this DP framework, we can see that $\Pi_n^{(i)} = P(S_n^{(2)} = i|\mathcal{F}_n)$. Then the expected cost of the distribution identification stage can be expressed as $C_2(\Pi_n, n) = c_2(n - \tau_1) + \mathbf{1}_{\{n-\tau_1<\infty\}}B(\Pi_n)$. Therefore, Π_n is the sufficient statistics for the DP process. Furthermore, we can express the minimum cost-to-go function at time n for this DP problem as

$$V_n^{T_2+\tau_1}(\Pi_n) = B(\Pi_n), \text{ if } n = T_2 + \tau_1, \tag{2.6}$$

$$V_n^{T_2+\tau_1}(\Pi_n) = \min\Big(B(\Pi_n), c_2 + G_n^{T_2+\tau_1}(\Pi_n)\Big), \text{ if } n < T_2 + \tau_1, \tag{2.7}$$

where

$$G_n^{T_2+\tau_1}(\Pi_n) = \mathbb{E}[V_{n+1}^{T_2+\tau_1}(\Pi_{n+1})|\mathcal{F}_n] = \int \left[V_{n+1}^{T_2+\tau_1}(\Pi_{n+1}(\Pi_n, x)) \sum_{i\in\mathcal{I}_0} f_i(x)\Pi_n^{(i)}\right] dx \tag{2.8}$$

The first item of the minimization in equations (2.7) is the misdiagnosis cost for stopping at time n, while the second item corresponds to the cost of proceeding to time $n+1$. In this way, we know that the minimum expected cost for the finite-horizon DP problem is $V_{\tau_1}^{T_2+\tau_1}(\Pi_{\tau_1})$. Therefore, in the identification stage of finite-horizon two-stage SCD problem, the optimal stopping rule is stopping immediately when $B(\Pi_n) \leq c_2 + G_n^{T_2+\tau_1}(\Pi_n)$ or $n = T_2 + \tau_1$. This optimal rule tells us we should stop only when the expected cost for making identification is less or equal to the expected cost of observing more data.

After knowing the optimal stopping rule of the distribution identification stage and the minimum expected cost $V_{\tau_1}^{T_2+\tau_1}(\Pi_{\tau_1})$ for any given τ_1 and Π_{τ_1}, selecting an optimal τ_1 to minimize the total Bayesian cost becomes a single stopping time problem. The method to solve this problem is similar to the distribution identification stage.

Let $S_n^{(1)}$ denote the state of the system of the change detection stage at time $n \in [0, T_1]$. $S_n^{(1)}$ can take value 1 (post-change), 0 (pre-change) and E (End). Once a change alarm is raised, the state of system becomes E. The state evolves as $S_n^{(1)} = g_1(S_{n-1}^{(1)}, \lambda, \mathbf{1}_{\{\tau_1 \leq n\}})$ with $S_0^{(1)} = 0$, where the transition function g_1 is

$$g_1(s, \lambda, \mathbf{1}_{\{\tau_1 \leq n\}}) = \begin{cases} 0, & \text{if } \lambda > n, s \neq E, \tau_1 > n, \\ 1, & \text{if } \lambda \leq n, s \neq E, \tau_1 > n, \\ E, & \text{if } s = E \text{ or } \tau_1 \leq n. \end{cases}$$

In addition, the observations of this DP framework are the data samples $\{X_n\}_{n \geq 1}$. Under this DP framework, we can see that $\Pi_n^{(0)} = P(S_n^{(1)} = 0|\mathcal{F}_n)$ and $1 - \Pi_n^{(0)} = P(S_n^{(1)} = 1|\mathcal{F}_n)$. Then the expected cost of the whole SCD process can be expressed in terms of $\{\Pi_k\}_{k \leq n}$ as

$$C(n, \tau_2, d^*) = V_n^{T_2+n}(\Pi_n) + \sum_{k=0}^{n-1} c_1 \left(1 - \Pi_k^{(0)}\right) + \mathbf{1}_{\{n<\infty\}} a \Pi_n^{(0)}.$$

Therefore, $\{\Pi_k\}_{k \leq n}$ is the sufficient statistics for the DP process. Furthermore, we can express the minimum cost-to-go function at time n for this DP problem as

$$W_n^{T_1}(\Pi_n) = a\Pi_n^{(0)} + V_n^{T_2+n}(\Pi_n), \text{ if } n = T_1, \tag{2.9}$$

$$W_n^{T_1}(\Pi_n) \;=\; \min\Big(a\Pi_n^{(0)} \;+\; V_n^{T_2+n}(\Pi_n), c_1(1 \;-\; \Pi_n^{(0)}) \;+\; U_n^{T_1}(\Pi_n)\Big), \;\text{ if }\; n \;<\; T_1, \tag{2.10}$$

where

$$U_n^{T_1}(\Pi_n) = \mathbb{E}[W_{n+1}^{T_1}(\Pi_{n+1})|\mathcal{F}_n] = \int \left[W_{n+1}^{T_1}(\Pi_{n+1}(\Pi_n, x)) \sum_{i\in\mathcal{I}_0} f_i(x)\Pi_n^{(i)}\right] dx. \tag{2.11}$$

The first item of the minimization in equation (2.10) is the cost for stopping at time n, while the second item corresponds to the cost of proceeding to time $n+1$. In this way, we know that the minimum expected cost for the finite-horizon DP problem is $W_0^{T_1}(\Pi_0)$. Therefore, in the detection stage of finite-horizon two-stage SCD problem, the optimal stopping rule is stopping immediately when $a\Pi_n^{(0)} + V_n^{T_2+n}(\Pi_n) \leq c_1(1-\Pi_n^{(0)}) + U_n^{T_1}(\Pi_n)$ or $n = T_1$.

After establishing the DP frameworks for the two stages of the finite-horizon SCD problem, we can extend the frameworks to the infinite-horizon case, i.e., letting T_1 and T_2 go to infinity.

Theorem 2.1. For any $\Pi \in \mathcal{Z}$, the infinite-horizon cost-to-go function for the DP process of the identification stage is

$$V(\Pi) = \lim_{T_2\to\infty} V_n^{T_2+\tau_1}(\Pi) = \min\Big(B(\Pi), c_2 + G_V(\Pi)\Big), \tag{2.12}$$

where

$$G_V(\Pi) = \mathbb{E}[V(\tilde{\Pi})|\mathcal{F}] = \int \left[V(\tilde{\Pi}(\Pi, x)) \sum_{i\in\mathcal{I}_0} f_i(x)\Pi^{(i)}\right] dx. \tag{2.13}$$

Here, $\tilde{\Pi}$ denotes the posterior probability of the next time slot.

Proof. Please see Appendix A.1. □

From optimality equation (2.12), we know that the optimal rule for this single optimal stopping time problem is

$$\tau_2^* = \inf_{n \geq \tau_1} \{B(\Pi_n) < c_2 + G_V(\Pi_n)\} - \tau_1. \tag{2.14}$$

The optimal stopping rule (2.14) tells us that when $B(\Pi_n) < c_2 + G_V(\Pi_n)$, the optimal option is making identification immediately. Otherwise, observing more data samples is a better choice.

Based on (2.10), we can study the infinite-horizon DP process of change detection stage by letting $T_1 \to \infty$.

Theorem 2.2. For any $\Pi \in \mathscr{Z}$, the infinite-horizon cost-to-go function for the detection stage is

$$W(\Pi) = \lim_{T_1 \to \infty} W_n^{T_1}(\Pi) \ = \min\Big(a\Pi^{(0)} + V(\Pi), c_1(1 - \Pi^{(0)}) + U_W(\Pi)\Big), \tag{2.15}$$

where

$$U_W(\Pi) = \mathbb{E}[W(\tilde{\Pi})|\mathcal{F}] = \int \left[W(\tilde{\Pi}(\Pi, x)) \sum_{i \in \mathcal{I}_0} f_i(x)\Pi^{(i)}\right] dx. \tag{2.16}$$

Proof. The proof of this theorem is very similar to the proof of Theorem 2.1 and thus omitted. □

From optimality equation (2.15), we can see that the optimal rule for this problem is

$$\tau_1^* = \inf_{n \geq 0} \{a\Pi_n^{(0)} + V(\Pi_n) < c_1(1 - \Pi_n^{(0)}) + U_W(\Pi_n)\}. \tag{2.17}$$

The optimal stopping rule (2.17) tells us that when $a\Pi_n^{(0)}+V(\Pi_n) < c_1(1-\Pi_n^{(0)})+U_W(\Pi_n)$, the optimal option is to raise change alarm immediately. Otherwise, it is better to wait and observe more samples.

2.4 Low Complexity Two-stage SCD Rule

Similar to other DP-based solutions, the computational complexity of the optimal solution obtained in Section 3.3 is high, especially when I is large. In this section, we design a low complexity threshold-based two-stage SCD rule. Furthermore, we analyze the performance of this low complexity rule and show that this rule is asymptotically optimal.

2.4.1 Threshold Two-stage SCD Rule

Here, we describe our low complexity two-stage SCD rule. Our low complexity rule is a threshold rule. In particular, the proposed rule is charcterized by a set of thresholds $\{A, \vec{\boldsymbol{B}} = (B_0, B_1, B_2, ..., B_M)\}$, in which A and every elements in $\vec{\boldsymbol{B}}$ are strictly positive constants. With these thresholds, the proposed threshold rule $\delta_T = (\tau_A, \tau_{\vec{\boldsymbol{B}}}, d_{\vec{\boldsymbol{B}}})$ is defined as

$$\begin{cases} \tau_A := \inf\{n \geq 1, \Pi_n^{(0)} < 1/(1+A)\}, \\ \tau_{\vec{\boldsymbol{B}}} := \min\limits_{i\in\mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}, \\ \tau_{\vec{\boldsymbol{B}}}^{(i)} := \inf\{n \geq 1, \Pi_n^{(i)} > 1/(1+B_i)\} - \tau_A, \\ d_{\vec{\boldsymbol{B}}} := \arg\min\limits_{i\in\mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}. \end{cases} \tag{2.18}$$

In this threshold rule, the first stopping time τ_A is the first time $\Pi_n^{(0)}$ falls below the threshold $1/(1+A)$. After τ_A, the rule turns to check the posterior probabilities $\Pi_n^{(i)}$ for all $i \in \mathcal{I}_0$. It will stop immediately if any threshold $1/(1+B_i)$ is exceeded. The identification decision depends on which threshold is passed. In order to guarantee that this rule is in the two-stage SCD rule space Δ, it must satisfy $\tau_{\vec{\boldsymbol{B}}} \geq 0$. This condition can be satisfied by choosing appropriate A and $\vec{\boldsymbol{B}}$. So we assume that A and $\vec{\boldsymbol{B}}$ applied in this SCD rule satisfy $\tau_{\vec{\boldsymbol{B}}} \geq 0$. We will discuss how to select such values in Section 2.4.3.

For $i \in \mathcal{I}_0$ and $n \geq 1$, define the logarithm of the odds-ratio process as

$$\Phi_n^{(i)} := \log \frac{\Pi_n^{(i)}}{1 - \Pi_n^{(i)}} = -\log \left[\sum_{j \in \mathcal{I}_0 \setminus \{i\}} \exp(-\Lambda_n(i,j)) \right]. \tag{2.19}$$

Using $\Phi_n^{(i)}$, δ_T can be expressed as:

$$\begin{cases} \tau_A = \inf \left\{ n \geq 1, \dfrac{1 - \Pi_n^{(0)}}{\Pi_n^{(0)}} > A \right\} \\ \quad = \inf\{n \geq 1, \Phi_n^{(0)} < -\log A\}, \\ \tau_{\vec{\boldsymbol{B}}} = \min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}, \\ \tau_{\vec{\boldsymbol{B}}}^{(i)} = \inf \left\{ n \geq 1, \dfrac{1 - \Pi_n^{(i)}}{\Pi_n^{(i)}} < B_i \right\} - \tau_A \\ \quad = \inf\{n \geq 1, \Phi_n^{(i)} > -\log B_i\} - \tau_A, \\ d_{\vec{\boldsymbol{B}}} = \arg\min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}. \end{cases} \tag{2.20}$$

The complexity of the threshold rule (2.18) is very low. After obtaining a new sample, we only need to update the posterior probabilities using the recursive formula (2.2), and then compare them with the thresholds. In the following, we will show that this rule is asymptotically optimal as c_1 and c_2 go to zero.

2.4.2 Asymptotic Analysis

We now analyze the performance of the proposed threshold rule as c_1 and c_2 go to zero, for which A should go to infinity and elements of $\vec{\boldsymbol{B}}$ should go to zero.

We first analyze the delays. From (2.18), we can easily see that the delays increase as $A \to \infty$ and $B_i \to 0$ for all $i \in \mathcal{I}$, as shown in the following proposition.

Proposition 2.2. For $i \in \mathcal{I}$, we have $\mathbb{P}_i - a.s.$ $\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A \to \infty$ as $B_i \to 0$, and $\mathbb{P}_i - a.s.$ $\tau_A \to \infty$ as $A \to \infty$.

Proof. For any $n \geq 1$,

$$\mathbb{P}_i(\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A \leq n) = \mathbb{P}_i\left(\bigcup_{k=1}^{n}\left\{\Pi_k^{(i)} > \frac{1}{1+B_i}\right\}\right) \leq \sum_{k=1}^{n}\mathbb{P}_i\left(\Pi_k^{(i)} > \frac{1}{1+B_i}\right).$$

So $\limsup\limits_{B_i \to 0} \mathbb{P}_i\left(\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A \leq n\right) \leq \sum_{k=1}^{n} \mathbb{P}_i\left(\Pi_k^{(i)} > 1\right) = 0$. Finally, we have

$$\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A \xrightarrow[B_i \to 0]{\mathbb{P}_i - a.s.} \infty.$$

Similarly, for any $n \geq 1$,

$$\mathbb{P}_i(\tau_A \leq n) = \mathbb{P}_i\left(\bigcup_{k=1}^{n}\left\{\Pi_k^{(0)} < \frac{1}{1+A}\right\}\right) \leq \sum_{k=1}^{n}\mathbb{P}_i\left(\Pi_k^{(0)} < \frac{1}{1+A}\right).$$

So $\limsup\limits_{B_i \to 0} \mathbb{P}_i\left(\tau_A \leq n\right) \leq \sum_{k=1}^{n} \mathbb{P}_i\left(\Pi_k^{(0)} < 0\right) = 0$. Finally, we have $\tau_A \xrightarrow[A \to \infty]{\mathbb{P}_i - a.s.} \infty$. □

To further analyze how fast these delays increase, we study the behavior of the LLR process defined in (2.3).

We first give some definitions. For every $i \in \mathcal{I}$ and $j \in \mathcal{I}_0 \backslash \{i\}$, let

$$l(i,j) := \begin{cases} q(i,0) + |\log(1-\rho)| & j \in \mathcal{I}_0 \backslash (\Gamma_i \cup \{i\}), \\ q(i,j) & j \in \Gamma_i, \end{cases}$$

where $q(i,j)$ is the Kullback-Leibler divergence from f_j to f_i, and

$$\Gamma_i = \{j \in \mathcal{I} \backslash \{i\} | q(i,j) < q(i,0) + |\log(1-\rho)|\}.$$

The next proposition illustrates the behavior of the LLR process when stopping times go to infinity.

Proposition 2.3. For $i \in \mathcal{I}$, we have

$$\frac{\Lambda_n(i,j)}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i-a.s.} l(i,j) \tag{2.21}$$

and

$$\frac{\Lambda_n(0,i)}{n} = -\frac{\Lambda_n(i,0)}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i-a.s.} -l(i,0). \tag{2.22}$$

Proof. The proof of (2.21) is already given in [67]. By (2.21), we can easily have equation (2.22). □

According to (2.19), $\Phi_n^{(i)}$ is a function of the LLR process. Therefore, when $A \to \infty$ and $B_i \to 0$ for all $i \in \mathcal{I}$, the behavior of $\Phi_n^{(i)}$ can also be characterized.

Proposition 2.4. Let $l(i) = l(i, j(i))$, $j(i) = \underset{j\in\mathcal{I}_0\setminus\{i\}}{\arg\min}\, l(i,j)$. Then, for every $i \in \mathcal{I}$, we have

$$\frac{\Phi_n^{(i)}}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i-a.s.} l(i). \tag{2.23}$$

In addition, we also have

$$\frac{\Phi_n^{(0)}}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i-a.s.} -l(i,0). \tag{2.24}$$

Proof. Please see Appendix A.2. □

Using Proposition 2.4, we can prove the following lemma which can be used to calculate the expectation of the delay cost.

Proposition 2.5. For every $i \in \mathcal{I}$ we have

$$\frac{(\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A - \lambda)_+}{-\log B_i} \xrightarrow[B_i\to 0]{\mathbb{P}_i-a.s.} \frac{1}{l(i)}. \tag{2.25}$$

In addition,

$$\frac{(\tau_A - \lambda)_+}{-\log A} \xrightarrow[A\to\infty]{\mathbb{P}_i-a.s.} \frac{1}{-l(i,0)}. \tag{2.26}$$

Proof. Please see Appendix A.3. □

Here we don't consider $\tau_{\vec{\boldsymbol{B}}}^{(0)} + \tau_A$. This is because the decision $d = 0$ is not an optimal choice in any case if c_1 and c_2 go to zero. We will prove this in Section 2.4.3.

We now analyze the the false alarm and misdiagnosis probabilities. From the proposed SCD rule, we can see that the false alarm probability in the first stage is bounded by $1/(1+A)$. When $A \to \infty$, the false alarm probability in the first stage is very close to $1/(1 + A)$ since $\tau_A \to \infty$. As for the misdiagnosis probability, we consider several upper bounds. Firstly, following Proposition 2.4 of [67], we can prove the following lemma.

Lemma 2.1. For any SCD rule $\delta = (\tau_1, \tau_2, d) \in \Delta$, let

$$\begin{cases} R_{ji}(\delta) = \mathbb{P}\left\{\theta = j, d = i, \lambda \leq \tau_1 + \tau_2 < \infty\right\}, i \in \mathcal{I}, j \in \mathcal{I}\backslash\{i\}, \\ R_{0i}(\delta) = \mathbb{P}\left\{d = i, \lambda > \tau_1 + \tau_2\right\}, i \in \mathcal{I}. \end{cases}$$

Then for $j \in \mathcal{I}_0\backslash\{i\}$, we have

$$R_{ji}(\delta) = v_i \mathbb{E}_i \left[\mathbf{1}_{\{d=i, \lambda \leq \tau_1+\tau_2<\infty\}} e^{-\Lambda_\tau(i,j)}\right].$$

Using Lemma 2.1, we have the following upper bound for the misdiagnosis probability.

Proposition 2.6. For every $i \in \mathcal{I}$, we have

$$\sum_{j \in \mathcal{I}_0\backslash\{i\}} R_{ji}(\delta_T) \leq v_i B_i.$$

Proof. By Lemma 2.1, we have

$$R_{ji}(\delta_T) = v_i \mathbb{E}_i \left[\mathbf{1}_{\{d_{\vec{\boldsymbol{B}}}=i, \lambda<\tau<\infty\}} e^{-\Lambda_\tau(i,j)}\right].$$

Then we have

$$\sum_{j \in \mathcal{I}_0 \backslash \{i\}} R_{ji}(\delta_T) = \sum_{j \in \mathcal{I}_0 \backslash \{i\}} v_i \mathbb{E}_i \left[\mathbf{1}_{\{d_{\vec{\boldsymbol{B}}}=i, \lambda < \tau < \infty\}} e^{-\Lambda_\tau(i,j)} \right] = v_i \mathbb{E}_i \left[\mathbf{1}_{\{d_{\vec{\boldsymbol{B}}}=i, \lambda < \tau < \infty\}} \sum_{j \in \mathcal{I}_0 \backslash \{i\}} e^{-\Lambda_\tau(i,j)} \right]$$
$$\leq v_i B_i.$$

The last inequality is due to

$$\sum_{j \in M_0 \backslash \{i\}} e^{-\Lambda_\tau(i,j)} = e^{-\Phi_\tau^{(i)}} < B_i.$$

□

Using Proposition 2.6, we have the following proposition.

Proposition 2.7. For every $i \in \mathcal{I}$ and $j \in \mathcal{I}_0 \backslash \{i\}$, we have

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{I}_0 \backslash \{i\}} b_{ji} R_{ji}(\delta_T) \leq \sum_{i \in \mathcal{I}} v_i B_i \overline{b_i},$$

where $\overline{b_i} := \max_{j \in M_0 \backslash \{i\}} b_{ji}$.

Therefore, we know that the misdiagnosis probabilities for $d = i \in \mathcal{I}$ goes to zero as $B_i \to 0$. Now, we need to study the misdiagnosis probability for the case $d = 0$. The misdiagnosis probability in this case is $1 - \Pi^{(0)}_{\tau_A + \tau_{\vec{\boldsymbol{B}}}}$. The following proposition shows that this misdiagnosis probability does not go to zero.

Proposition 2.8. For any $\lambda \geq 0$, there always exists $0 < x < 1$, such that the posterior probability $\Pi_n^{(0)} < x$ is always true.

Proof. Please see Appendix A.4. □

By Propositions 2.7 and 2.8, we know that the misdiagnosis probability for the case $d = 0$ is much larger than misdiagnosis probability for the case $d \in \mathcal{I}$ if $B_i \to 0$.

2.4.3 Threshold Selection

We now discuss how to select the thresholds A and $\vec{\boldsymbol{B}}$. By Proposition 2.2, we know that $\tau_A \to \infty$ as $A \to \infty$. This implies that $\tau_A > \lambda$ almost surely as $A \to \infty$. So we have $\mathbb{E}(\tau_A - \lambda)_+ = \mathbb{E}(\tau_A - \lambda)$ as $A \to \infty$. If the condition $\tau_{\vec{\boldsymbol{B}}}^{(i)} \geq 0$, i.e.,

$$\inf \left\{ n \geq 1, \tfrac{1-\Pi_n^{(i)}}{\Pi_n^{(i)}} < B_i \right\} \geq \tau_A \tag{2.27}$$

is satisfied for all $i \in \mathcal{I}$, we can calculate the delay cost as

$$c_1 \mathbb{E}[(\tau_A - \lambda)_+] + c_2 \mathbb{E}(\tau_{\vec{\boldsymbol{B}}}) = (c_1 - c_2)\mathbb{E}[(\tau_A - \lambda)_+] + c_2 \mathbb{E}[(\tau_A + \tau_{\vec{\boldsymbol{B}}} - \lambda)_+]. \tag{2.28}$$

We will discuss how to find A and $\vec{\boldsymbol{B}}$ which can guarantee that (2.27) is satisfied in the sequel.

Now, by Proposition 2.5 we have $\mathbb{E}_i[(\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A - \lambda)_+]$ for all $i \in \mathcal{I}$. However, we need $\mathbb{E}_i[(\tau_{\vec{\boldsymbol{B}}} + \tau_A - \lambda)_+]$ for all $i \in \mathcal{I}$ to calculate the expectation of delay. So we consider the following proposition.

Proposition 2.9. For every $i \in \mathcal{I}$, we have

$$\mathbb{E}_i\left[(\tau_{\vec{\boldsymbol{B}}} + \tau_A - \lambda)_+\right] \xrightarrow{\mathbb{P}_i - a.s.} \frac{-\log B_i}{l(i)}$$

if $B_i \to 0$ for all $i \in \mathcal{I}$.

Proof. Please see Appendix A.5. □

Now, under the following three conditions:

(a) $\tau_{\vec{\boldsymbol{B}}}^{(i)} \geq 0$, i.e., inequality (2.27) is satisfied;

(b) $A \to \infty$, $B_i \to 0$ for all $i \in \mathcal{I}$ as c_1 and c_2 go to 0;

(c) $d = 0$ is not the optimal decision in any cases as c_1 and c_2 go to 0;

we can calculate the Bayesian cost and the thresholds. After getting the thresholds, we will verify that the chosen thresholds do satisfy these conditions.

By Proposition 2.7, we know that there exists a set of constant k_i such that $k_i < \bar{b}_i$ and the misdiagnosis probability

$$\sum_{i\in\mathcal{I}} \sum_{j\in\mathcal{I}_0\backslash\{i\}} b_{ji} R_{ji}(\delta_T) = \sum_{i\in\mathcal{I}} v_i B_i k_i.$$

Similarly, the false alarm cost can be approximated by $k_a/(1+A)$ with a constant k_a in $(0, a)$. By Propositions 2.9, the delay cost can be calculated. Therefore, if $c_2 \to 0$ and the ratio constant r is fixed, the Bayesian cost can be calculated as

$$C^{(c_2)}(\delta_T) = \underbrace{c_2 \sum_{i\in\mathcal{I}} v_i \left(\frac{-\log(B_i)}{l(i)}\right) + \sum_{i\in\mathcal{I}} v_i B_i k_i}_{part1} + \underbrace{c_2(\frac{1}{r}-1) \sum_{i\in\mathcal{I}} \frac{v_i \log A}{l(i,0)} + \frac{k_a}{1+A}}_{part2}. \tag{2.29}$$

A simple calculation shows that to minimize (2.29), we should set the thresholds as

$$\begin{cases} A_{opt} \approx \frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}} \frac{v_i}{l(i,0)}} - 2, \\ B_{i,opt} = \frac{c_2}{k_i l(i)}, i \in \mathcal{I}. \end{cases} \tag{2.30}$$

Plugging in A_{opt} and $\vec{\boldsymbol{B}}_{opt}$, we have the corresponding rule δ_T^* and its Bayesian cost

$$\begin{aligned} C^{(c_2)}(\delta_T^*) = {} & c_2 \sum_{i\in\mathcal{I}} \frac{-v_i}{l(i)} \log(\frac{c_2}{k_i l(i)}) + \sum_{i\in\mathcal{I}} \frac{v_i c_2}{k_i l(i)} k_i \\ & + c_2(\frac{1}{r}-1) \sum_{i\in\mathcal{I}} \frac{v_i}{l(i,0)} \log\left(\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}} \frac{v_i}{l(i,0)}} - 2\right) + k_a \frac{1}{\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}} \frac{v_i}{l(i,0)}} - 1}. \end{aligned} \tag{2.31}$$

Now we need to check if the three conditions are satisfied. First, we check condition (a). By the threshold rule (2.20), we know that $\tau_{A_{opt}}$ is the first time $\sum_{i\in\mathcal{I}} \Pi_n^{(i)} = 1 - \Pi_n^{(0)}$ exceeds the threshold $1 - 1/(1 + A_{opt})$. Also, $\tau^{(i)}_{\vec{\boldsymbol{B}}_{opt}} + \tau_{A_{opt}}$ is the first time for $\Pi_n^{(i)}$ exceeds the threshold $1/(1+B_{i,opt})$. So if

$$1 - \frac{1}{1+A_{opt}} < \frac{1}{1+B_{i,opt}} \tag{2.32}$$

for all $i \in \mathcal{I}$, it is guaranteed that the threshold $\vec{\boldsymbol{B}}$ can not be reached before threshold A, namely, $\tau_{\vec{\boldsymbol{B}}} \geq 0$. After plugging the explicit expressions of the optimal thresholds (2.30) in inequality (3.18) and basic calculation, we know that a sufficient condition of $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is

$$0 < r \leq \min_{i \in \mathcal{I}} \frac{1}{1 + \frac{k_a}{k_i l(i) \sum\limits_{i \in \mathcal{I}} \frac{v_i}{l(i,0)}}}. \tag{2.33}$$

If the value of r satisfies (2.33), condition (a) is satisfied. However, for the case (2.33) is not satisfied, we need to change the threshold accordingly as

$$\begin{cases} A' = A_{opt}, \\ B'_i = B_{i,opt} \frac{k_i}{\eta}, i \in \mathcal{I} \end{cases} \tag{2.34}$$

where η is a constant such that

$$r = \min_{i \in \mathcal{I}} \frac{1}{1 + \frac{k_a}{\eta l(i) \sum\limits_{i \in \mathcal{I}} \frac{v_i}{l(i,0)}}}.$$

We can see that with A' and $\vec{\boldsymbol{B}}'_{opt}$, condition (a) is satisfied. Hence the Bayesian cost of the rule $\delta'_T = (\tau_{A'}, \tau_{\vec{\boldsymbol{B}}'}, d')$ is

$$C^{(c_2)}(\delta'_T) = C^{(c_2)}(\delta^*_T) - c_2 \sum_{i \in \mathcal{I}} \log\left(\frac{k_i}{\eta}\right) \frac{v_i}{l(i)} + \sum_{i \in \mathcal{I}} v_i B_{i,opt} \left(\frac{k_i^2}{\eta} - k_i\right). \tag{2.35}$$

Since k_i, $l(i)$ and η are constants, the last two terms in (2.35) decay much faster than $C^{(c_2)}(\delta^*_T)$ as $c_2 \to 0$. This implies that the difference between the cost calculated by (2.31) and (2.35) is negligible as $c_2 \to 0$. So condition (a) is satisfied. Then we can see that the

Bayesian cost in (2.31) for any $0 < r < 1$ goes to 0 as $c_2 \to 0$. However, by Proposition 2.8, there is always a constant cost $x > 0$ if the decision $d = 0$ is made. Hence, choosing $d = 0$ will always end up with a higher Bayesian cost, as long as $c_2 \to 0$. So condition (c) is true, hence B_0 is set to be 0 to disable $d = 0$. In addition, it's easy to see that condition (b) is true by (2.30) and (2.34).

In summary, we select thresholds in the following manner: if r satisfies (2.33), we set the thresholds according to (2.30); otherwise, we choose the thresholds as (2.34). Besides, $B_0 = 0$.

Finally, we consider the values of k_a and $\{k_i\}_{i \in \mathcal{I}}$. As we can see from equations (2.31) and (2.35), the cost of false alarm and misdiagnosis costs decay much faster than the delay cost as $c_2 \to 0$. Therefore, as long as $\{k_i\}$ and k_a are constants, taking different values for them will not change the asymptotic behavior of the Bayesian cost. Typically, we set k_a to be the penalty factor a. For k_i, [67] introduced a method to calculate a higher order approximation of k_i:

$$k_i = b_{j(i)i} \mathbb{E}_i [e^{-Z_i}], i \in \mathcal{I}. \tag{2.36}$$

Here Z_i is a random variable with distribution

$$\mathbb{P}_i(Z_i \leq z) = \frac{\int_0^z \mathbb{P}_i^{(0)} \left\{ \sum_{l=0}^{T_i^{(0)}} \log \frac{f_i(X_l)}{f_{j(i)}(X_l)} > s \right\} ds}{\mathbb{E}_i^{(0)} \left[\sum_{l=0}^{T_i^{(0)}} \log \frac{f_i(X_l)}{f_{j(i)}(X_l)} \right]},$$

where $0 < z < \infty$ and

$$T_i^{(0)} := \inf \left\{ n \geq 0 : \sum_{l=0}^{n} \log \left(\frac{f_i(X_l)}{f_{j(i)}(X_l)} \right) > 0 \right\}. \tag{2.37}$$

2.4.4 Asymptotic Optimality

We now show that the threshold two-stage SCD rule is asymptotically optimal as $c_2 \to 0$. In particular, we will show that, for any $\delta = (\tau_1, \tau_2, d) \in \Delta$, we have

$$\frac{C^{(c_2)}(\delta)}{C^{(c_2)}(\delta_T)} \geq 1, \tag{2.38}$$

in which $\delta_T = (\tau_{A_T}, \tau_{\vec{\boldsymbol{B}}_T}, d_T)$ with thresholds A_T and $\vec{\boldsymbol{B}}_T$ computed using (2.30) or (2.34) according to the value of r. We already know that the difference between Bayesian costs calculated by (2.31) and (2.35) is negligible as $c_2 \to 0$. So we only need to consider the cost function calculated by (2.31), i.e., the case in which r satisfies (2.33) and hence A_T and $\vec{\boldsymbol{B}}_T$ is set as (2.30).

First, we study the delay cost of an SCD rule $\delta = (\tau_1, \tau_2, d) \in \Delta$. For convenience of expression, we define $\Delta_1 := \{\tau_1 | \tau_1 \text{ is any stopping time associated to } \mathbb{F}\}$ as the collection of all possible one-time change detection rules for the first stage. We also denote collections of rules which has bounded false alarm and misdiagnosis probabilities for the two stages respectively as $\Delta_1(\overline{R_0}) := \{\tau_1 \in \Delta_1 | R_0(\tau_1) \leq \overline{R_0}\}$, and $\Delta(\overline{R}) := \{(\tau_1, \tau_2, d) \in \Delta | R_{ji}(\tau_1, \tau_2, d) \leq \overline{R_{ji}}, i \in \mathcal{I}, j \in \mathcal{I}_0 \backslash \{i\}\}$, where $\overline{R}_0$ and $\overline{R} = \{\overline{R_{ji}}\}_{i \in \mathcal{I}, j \in \mathcal{I}_0 \backslash \{i\} \geq 0}$ are the upper bounds of false alarm and misdiagnosis probabilities respectively. As we discuss in 2.4.3, $d = 0$ should not be considered for a rule that can outperform our threshold rule as $c_2 \to 0$. So a bound for $i = 0$ is unnecessary here.

From (2.31), we know that the Bayesian cost of the threshold SCD rule goes to zero as $c_2 \to 0$. If there exists a rule such that it has a lower cost than the threshold rule, the false alarm and misdiagnosis cost must go to zero. Therefore, we only need to consider the SCD rule $\delta = (\tau_1, \tau_2, d)$ such that $\delta \in \Delta(\overline{R})$ and $\tau_1 \in \Delta_1(\overline{R_0})$ where $\overline{R} \to 0$ and $\overline{R_0} \to 0$. Here $\overline{R} \to 0$ means that every constant in set $\overline{R}$ goes to zero. If false alarm and misdiagnosis probabilities go to zero, the delays τ_1 and τ_2 must go to infinity. Given λ is finite almost surely, the delay cost can be expanded as

$$c_1\mathbb{E}[(\tau_1-\lambda)_+]+c_2\mathbb{E}(\tau_2)=(c_1-c_2)\mathbb{E}[(\tau_1-\lambda)_+]+c_2\mathbb{E}[(\tau_1+\tau_2-\lambda)_+].$$

The following lemma provides the lower bounds of $\mathbb{E}[(\tau_1-\lambda)_+]$ and $\mathbb{E}[(\tau_1+\tau_2-\lambda)_+]$ respectively.

Lemma 2.2. *If $i\in\mathcal{I}$ and $\delta=(\tau_1,\tau_2,d)$, we have*

$$\begin{cases}\liminf\limits_{\overline{R}\to 0}\inf\limits_{\delta\in\Delta(\overline{R})}\frac{\mathbb{E}_i[(\tau_1+\tau_2-\lambda)_+]}{\left|\log(\overline{R_{j(i)i}}/v_i)\right|/l(i)}\geq 1\\ \liminf\limits_{\overline{R_0}\to 0}\inf\limits_{\tau_1\in\Delta_1(\overline{R_0})}\frac{\mathbb{E}_i[(\tau_1-\lambda)_+]}{\left|\log(\overline{R_0}/v_i)\right|/l(i,0)}\geq 1\end{cases}\tag{2.39}$$

Proof. Please see Appendix A.6. □

With the lower bound of the delay, we finally establish the asymptotic optimality of the threshold two-stage SCD rule.

Proposition 2.10. If $\delta_T=(\tau_{A_T},\tau_{\vec{\boldsymbol{B}}_T},d_T)$ is a threshold two-stage SCD rule with thresholds as (2.30), then for any given fixed r we have

$$\lim_{c_2\to 0}\frac{\inf_{\delta\in\Delta}C^{(c_2)}(\delta)}{C^{(c_2)}(\delta_T)}\geq 1.$$

Proof. Please see Appendix A.7. □

This proposition implies that for given r satisfies (2.33), the threshold SCD rule with threshold (2.30) is asymptotically optimal. Since the difference between Bayesian costs calculated by (2.31) and (2.35) is negligible as $c_2\to 0$, so the asymptotic optimality of the proposed threshold SCD rule holds generally for any $0<r<1$.

Table 2.1: Comparison of the optimal Bayesian two-stage costs with different c_1 and r

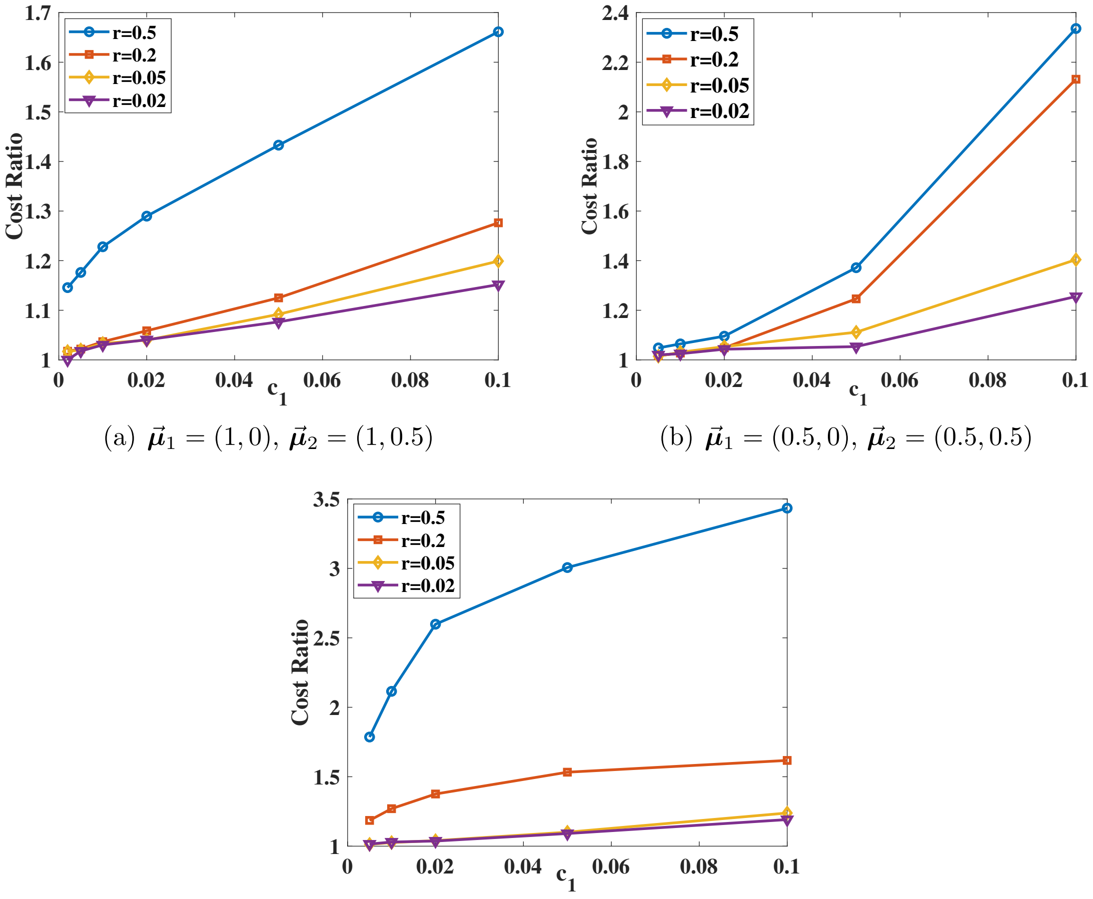
r c_1	0.02	0.05	0.2	0.5	1 (One-stage)
0.005	0.0720	0.0798	0.1009	0.1309	0.1580
0.02	0.2352	0.2511	0.3115	0.3695	0.4016
0.05	0.4763	0.5086	0.6123	0.6853	0.6980
0.2	0.9392	0.9892	1.0021	1.0023	1.0023
0.5	1.0059	1.0062	1.0058	1.0064	1.0067

2.5 Numerical Example

In this section, we provide numerical examples to illustrate the performance of the optimal and threshold SCD rules. In our simulation, the observed data samples are generated by a two-dimensional normal distribution, $\mathcal{N}(\boldsymbol{\vec{\mu}}, \boldsymbol{I}_2)$. The mean vector $\boldsymbol{\vec{\mu}}$ changes at the change point.

In the first example, we consider the case with two possible post-change mean vectors $\boldsymbol{\vec{\mu}}_1 = (1, 0)$ and $\boldsymbol{\vec{\mu}}_2 = (1, 0.5)$ and the pre-change mean vector $\boldsymbol{\vec{\mu}}_0 = (0, 0)$. In addition, we set $\rho_0 = 0$, $\rho = 0.01$, $(v_1, v_2) = (0.3, 0.7)$. All the penalty factors of the false alarm and misdiagnosis are set to be 1. The results are estimated by Monte-Carlo simulations. Table 2.1 presents the expected costs of the optimal two-stage SCD rule with different delay penalty factor settings, i.e., with different c_1 and r.

From Table 2.1, we can see that the performance of the optimal two-stage SCD rule becomes better as c_1 and r get smaller. In particular, with identical c_1, the optimal two-stage SCD rules with $r < 1$ generally outperform the rules with $r = 1$. Note that, the two-stage SCD problem will become a one-stage SCD problem when $r = 1$. Therefore, this result validates that the optimal two-stage SCD rule generally outperforms the optimal one-stage SCD rule when $c_2 < c_1$. Furthermore, with smaller c_1, the performance improvement brought by reducing r is more significant. The reason is, with a small c_1, we can use more data to improve the accuracy of change detection and identification without a significant increment of the delay cost. On the contrary, when c_1 is large enough, the performance can still be very poor even with a very small r. This result implies that when the per-unit delay



(a) $\vec{\boldsymbol{\mu}}_1 = (1, 0)$, $\vec{\boldsymbol{\mu}}_2 = (1, 0.5)$

(b) $\vec{\boldsymbol{\mu}}_1 = (0.5, 0)$, $\vec{\boldsymbol{\mu}}_2 = (0.5, 0.5)$

(c) $\vec{\boldsymbol{\mu}}_1 = (1, 0)$, $\vec{\boldsymbol{\mu}}_2 = (1, 0.25)$

Figure 2.1: The cost ratio between the optimal and threshold two-stage SCD rules

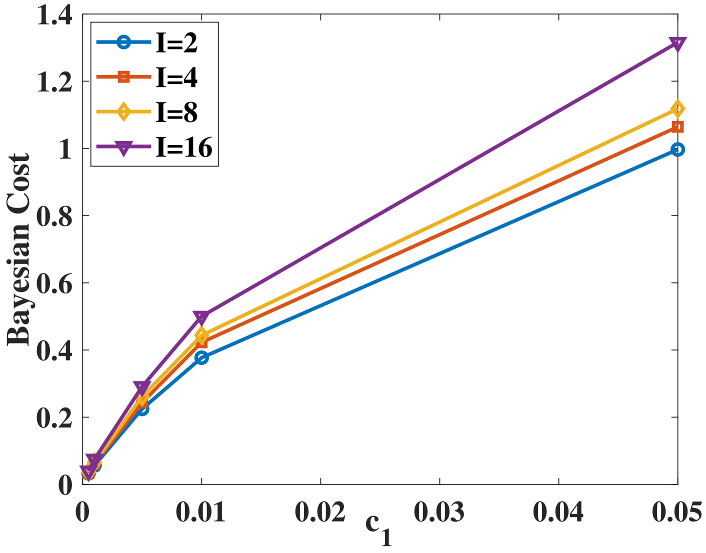


Figure 2.2: The Bayesian costs of the threshold two-satge SCD rules with different number of post-change distributions

cost is too large, the improvement on diagnosis accuracy becomes too expensive and also negligible.

Figure 4.4(a) illustrates the ratio between the costs of optimal and threshold SCD rules with different penalty factors, c_1 and r. The constants $\{k_i\}_{i\in\mathcal{I}}$ used to get the thresholds are approximated using (2.36) and k_a is set as 1. From this figure, we can see that the Bayesian cost of the threshold SCD rule converges to the cost of optimal SCD rule as $c_1 \to 0$. This result validates the asymptotic optimality of the threshold SCD rule. From the lines for different r values, we can see that the cost of the threshold SCD rule converges to the cost of optimal SCD rule faster and faster as r decreases. This implies that, with the same c_1, a smaller c_2 makes the cost of the threshold SCD rule more close to the cost of the optimal rule.

In the second example, we compare the performances of the threshold SCD rule in problems with different difficulty level. In particular, we investigate the performance of the threshold rule when the KL distances between f_0, f_1 and f_2 are reduced. Keeping all other parameters in example 1, we run two simulations: 1) In simulation 1, the post-change mean vectors are $\vec{\boldsymbol{\mu}}_1 = (0.5, 0)$ and $\vec{\boldsymbol{\mu}}_2 = (0.5, 0.5)$; 2) In simulation 2, the post-change mean vectors are $\vec{\boldsymbol{\mu}}_1 = (1, 0)$ and $\vec{\boldsymbol{\mu}}_2 = (1, 0.25)$. Results are shown in Figure 4.4(b) and 2.1(c).

From Figures 4.4(a), 4.4(b) and 2.1(c), the ratio between the costs of the optimal SCD rule and the threshold SCD rule is generally large when c_1 and r are not very small, especially when f_0, f_1 and f_2 are close. However, when c_1 and r are sufficiently small, the performance of threshold SCD rule becomes very close to the optimal SCD rule in all the examples, even if f_0, f_1 and f_2 are close. This indicates the difficulty of the change diagnosis task will not change the asymptotic optimality.

In the third example, we investigate the performances of the threshold SCD rule when there are more than two candidate post-change distributions. To this end, we implement four sets of simulations with 2, 4, 8, and 16 post-change distributions. In each set of simulations, all the distributions are still 2D Gaussian. The prior probabilities of the post-change situations are uniformly distributed, i.e., $(v_1, \dots, v_I) = (1/I, \dots, 1/I)$. The mean vector of the pre-change Gaussian distribution is $\vec{\boldsymbol{\mu}}_0 = (0, 0)$. The mean vectors of the post-change Gaussian distributions are uniformly distributed on the circle centering μ_0 with radius 0.5. For example, if $d = 4$, we can set $\vec{\boldsymbol{\mu}}_1 = (0.5, 0), \vec{\boldsymbol{\mu}}_2 = (0, 0.5), \vec{\boldsymbol{\mu}}_3 = (-0.5, 0), \vec{\boldsymbol{\mu}}_4 = (0, -0.5)$. The co-variance matrices of all distributions are identity matrices. In addition, ρ_0, ρ and penalty factors are same as example 1. The results of the simulations are presented in Figure 4.8. As we expected, with more post-change distributions around the same circle, the threshold SCD rule will have a larger Bayesian cost.

2.6 Conclusion

In this chapter, we have formulated the Bayesian two-stage sequential change diagnosis problem. We have converted the problem into two optimal single stopping time problems and obtained the optimality equations of them. After solving these equations using dynamic programming, we have obtained the optimal rule for the Bayesian two-stage SCD problem. However, the complexity of the proposed optimal solution is high due to the DP steps. To reduce the computational complexity, we have designed a threshold two-stage SCD rule and proved that this threshold rule is asymptotically optimal as the per-unit delay costs of the two stages go to zero.

Chapter 3

Bayesian Two-stage Sequential Change Diagnosis via Multi-sensor Array

In this chapter, we study the two-stage SCD problem in a multi-sensor array case. Firstly, we introduce the problem formulation with a linear sensor array. Then we introduce the structure of the optimal solution for the two-stage SCD problem. Afterward, we propose the low complexity threshold rule to the two-stage sensor array SCD problem and prove the asymptotic optimality of this method. Furthermore, we extend this method from a linear sensor array case to a more general 2D lattice sensor array case. Finally, simulation results are given to illustrate the performance of proposed SCD rule.

3.1 Problem Formulation

To facilitate the presentation and easiness of understanding, we will first present our work for the linear array case (as shown in Fig. 1.4). The more complicated 2D array scenario will be presented in Section 3.5.

In the linear array scenario, there is a linear array of L sensors monitoring the environment. The L sensors collect data at each time unit and then immediately send data to the fusion center for analysis. The observation of the system is a stochastic process hosted by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. At time k, the observation of the system is $\vec{\boldsymbol{X}}_k = (x_{k,1}, x_{k,2}, ..., x_{k,L})$, where $x_{k,l}$ is the data collected by the lth sensor at time k. Let $\lambda : \Omega \mapsto \{0, 1, \ldots\}$ be the time when an abrupt change happens in the sensing environment and $\theta : \Omega \mapsto \mathcal{I} := \{1, \ldots, I\}$ be the environment state after the change. The prior distribution of the change time is $\mathbb{P}(\lambda = k) = \rho(1-\rho)^k$. In addition, we denote $\mathcal{I} \cup \{0\}$ as $\mathcal{I}_0$. After time λ, the distribution of the data collected by each sensor may experience a change from f_0 to f_θ. f_θ can be one of the candidate distributions $\{f_i\}_{i \in \mathcal{I}}$. In addition, $\mathbb{F} = (\mathcal{F}_k)_{k \geq 0}$ is the filtration generated by the stochastic process $\{\vec{\boldsymbol{X}}_k\}_{k \geq 1}$.

3.1.1 Change Propagation Model

The change propagation model is illustrated in Fig. 1.4, the change will first happen to one sensor in the array and then propagate to other sensors. In the considered model, the change times of different sensors may be different. We denote the time change happen to sensor l as λ_l for all $1 \leq l \leq L$. Let S denote the index of the sensor that the change first reaches. The prior probability $\mathbb{P}(S = l) = \kappa_l$ is known. We denote $(\kappa_1, \kappa_2, \ldots, \kappa_L)$ as $\vec{\boldsymbol{K}}$. As shown in Fig. 1.4, the change first reaches sensor S at time $\lambda_S = \lambda$, then the change will propagate to sensors on both sides of sensor S following the directions $S \to S+1 \to \cdots \to L$ and $S \to S-1 \to \cdots \to 1$. The propagation of the change in the sensor array follows a geometric distribution, i.e., for $k_2 \geq 0$,

$$\begin{cases} \mathbb{P}\left[\lambda_{j-1} = k_1 + k_2 | \lambda_j = k_1, S = i\right] = \rho_1(1-\rho_1)^{k_2}, i > j \\ \mathbb{P}\left[\lambda_{j+1} = k_1 + k_2 | \lambda_j = k_1, S = i\right] = \rho_2(1-\rho_2)^{k_2}, i < j \end{cases}$$

where ρ_1 and ρ_2 are the probabilities of the change propagate from a sensor to its neighbor at each time step for the two directions.

3.1.2 Observation Model

In this section, we assume the observations of different times at every sensor are independent, conditioned on the change information. Concretely, if $k < \lambda_l$, $z_{k,l} \sim f_0$, otherwise $z_{k,l} \sim f_\theta$, where $\theta \in \mathcal{I}$. The prior probability of the state after change is defined as $v_i = \mathbb{P}\{\theta = i\}, i \in \mathcal{I}$. To simplify the notation, we express the conditional probabilities as:

$$\begin{cases} \mathbb{P}_i\{\cdot\} = \mathbb{P}\{\cdot|\theta = i\}, \\ \mathbb{P}_i^{(t)}\{\cdot\} = \mathbb{P}\{\cdot|\theta = i, \lambda = t\}, t \geq 0. \end{cases}$$

Correspondingly, $\mathbb{E}_i$ and $\mathbb{E}_i^{(t)}$ are the expectations under $\mathbb{P}_i$ and $\mathbb{P}_i^{(t)}$. Finally, assumption 2.1 also holds in this chapter.

3.1.3 Two-stage Multi-sensor SCD Problem

Our goal is to quickly raise an alarm after the change occurs and further accurately determine the state θ, based on all the data samples $\{\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k\}$. Towards this goal, we employ a two-stage SCD rule $\delta = (\tau_1, \tau_2, d)$ that includes two stopping times τ_1, $\tau_1 + \tau_2$, and an identification decision d. Here, τ_1 is the time for the change detection and $\tau_1 + \tau_2$ is the time for the identification. Let $\Delta := \{(\tau_1, \tau_2, d)|\tau_1 \geq 0, \tau_2 \geq 0, d \in \mathcal{I}_0\}$ be the set of all possible two-stage SCD rules. We should note that if a wrong decision is made at τ_1, i.e., $\tau_1 < \lambda$, then $d = 0$ is the correct identification as long as this identification is made before λ, i.e., $\tau_1 + \tau_2 < \lambda$. Besides, the parameters ρ, ρ_1, ρ_2, $\vec{\boldsymbol{K}}$ and $\{v_i\}_{i \in \mathcal{I}}$ are known.

The possible costs of an SCD rule include costs of delay, false alarm, and misdiagnosis. The delay consists of the delays in the change detection stage and the distribution identification stage, i.e. $(\tau_1 - \lambda)_+$ and τ_2. The expected delay costs of them are $\mathbb{E}[c_1(\tau_1 - \lambda)_+]$ and $\mathbb{E}[c_2\tau_2]$, where c_1 and c_2 are per-unit delay costs associated with each stage and $(z)_+ = \max(0, z)$ for any z. In addition, we define $r := c_2/c_1$ as the ratio between per-unit delay costs. A false alarm is the situation that a change alarm is raised before λ.

The expected false alarm cost is $\mathbb{E}[a\mathbf{1}_{\{\tau_1<\lambda\}}]$, where a is the penalty factor of false alarm and $\mathbf{1}_{\{\cdot\}}$ is the indicator function. Misdiagnosis occurs when a wrong identification is made, i.e., $d \neq \theta$. The expected misdiagnosis cost is $\mathbb{E}\Big[\sum_{i\in\mathcal{I}} b_{ij}\mathbf{1}_{\{\infty>\tau_1+\tau_2>\lambda,\theta=i,d=j\}} + b_{0j}\mathbf{1}_{\{\tau_1+\tau_2<\lambda,d=j\}}\Big]$ for $d = j$, where b_{ij} is the penalty factor for wrong decision $d = j$ when $\theta = i$ and $b_{0,j}$ is the penalty factor of the false alarm of the identification stage. We set $b_{ij} = 0$ when $i = j$. Hence the Bayesian cost function for a two-stage SCD rule $\delta \in \Delta$ is

$$\begin{aligned} &C(\delta) = c_1\mathbb{E}\left[(\tau_1-\lambda)_+\right] + c_2\mathbb{E}[\tau_2] + a\mathbb{E}[\mathbf{1}_{\{\tau_1<\lambda\}}] + \\ &\sum_{j=0}^{I}\mathbb{E}\Big[\sum_{i=1}^{I} b_{ij}\mathbf{1}_{\{\infty>\tau_1+\tau_2>\lambda,\theta=i,d=j\}} + b_{0j}\mathbf{1}_{\{\tau_1+\tau_2<\lambda,d=j\}}\Big]. \end{aligned} \tag{3.1}$$

The goal of the SCD problem is to find an SCD rule (τ_1, τ_2, d) that minimizes the expected cost $C(\delta)$.

3.2 Posterior Probability Analysis

Following the main idea of [73], we can solve a two-stage SCD problem using posterior probability process, $\Pi_k = (\Pi_k^{(0)}, \ldots, \Pi_k^{(I)})_{k\geq 0} \in \mathcal{Z}$, which is defined as

$$\begin{cases} \Pi_k^{(i)} := \mathbb{P}\{\lambda \leq k, \theta = i|\mathcal{F}_k\}, i \in \mathcal{I}, \\ \Pi_k^{(0)} := \mathbb{P}\{\lambda > k|\mathcal{F}_k\}, \end{cases}$$

and

$$\mathcal{Z} \triangleq \{\Pi \in [0,1]^{I+1} | \sum\nolimits_{i\in\mathcal{I}\cup\{0\}} \Pi^{(i)} = 1\}.$$

Using Bayesian rule, we know that, at any time $k \geq 1$, each component of Π_k can be computed as

$$\Pi_k^{(i)} = \frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,\ldots,\vec{\boldsymbol{X}}_k)}{\sum_{j\in\mathcal{I}_0}\alpha_k^{(j)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,\ldots,\vec{\boldsymbol{X}}_k)}, \tag{3.2}$$

in which

$$\left\{\begin{array}{l}
\alpha_k^{(0)} = (1-\rho)^{k+1} \prod\limits_{l=1}^{L} \prod\limits_{n=1}^{k} f_0(x_{n,l}) \\
\alpha_k^{(i)} = \sum\limits_{s=1}^{L} \kappa_s v_i \rho \sum\limits_{n_s=0}^{k} \left[(1-\rho)^{n_s} \left(\prod\limits_{n=1}^{n_s-1} f_0(x_{n,s}) \right) \cdot \left(\prod\limits_{n=max(n_s,1)}^{k} f_i(x_{n,s}) \right) \Psi_{s-1}^{(i)}(k,n_s) \Phi_{s+1}^{(i)}(k,n_s) \right] \\
\Psi_{l-1}^{(i)}(k,n_l) = (1-\rho_1)^{k-n_l+1} \prod\limits_{t=1}^{l-1} \prod\limits_{n=1}^{k} f_0(x_{n,t}) + \rho_1 \sum\limits_{n_{l-1}=n_l}^{k} \left[(1-\rho_1)^{n_{l-1}-n_l} \left(\prod\limits_{n=1}^{n_{l-1}-1} f_0(x_{n,l-1}) \right) \cdot \right. \\
\left. \left(\prod\limits_{n=n_{l-1}}^{k} f_i(x_{n,l-1}) \right) \Psi_{l-2}^{(i)}(k,n_{l-1}) \right], l > 1 \\
\Phi_{l+1}^{(i)}(k,n_l) = (1-\rho_2)^{k-n_l+1} \prod\limits_{t=l+1}^{L} \prod\limits_{n=1}^{k} f_0(x_{n,t}) + \rho_2 \sum\limits_{n_{l+1}=n_l}^{k} \left[(1-\rho_2)^{n_{l+1}-n_l} \left(\prod\limits_{n=1}^{n_{l+1}-1} f_0(x_{n,l+1}) \right) \cdot \right. \\
\left. \left(\prod\limits_{n=n_{l+1}}^{k} f_i(x_{n,l+1}) \right) \Phi_{l+2}^{(i)}(k,n_{l+1}) \right], l < L \\
\Phi_{L+1}^{(i)}(k,n_l) = \Psi_0^{(i)}(k,n_l) = 1
\end{array}\right. . \tag{3.3}$$

Assumption 2.1 implies $0 < \Pi_k^{(i)} < 1$ for every finite $k \geq 1$ and $i \in \mathcal{I}_0$. We define the log-likelihood-ratio (LLR) processes as

$$\Lambda_k(i,j) := \log \frac{\Pi_k^{(i)}}{\Pi_k^{(j)}} = \log \frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)}{\alpha_k^{(j)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k)}. \tag{3.4}$$

Directly calculating Π_k based on (3.2) requires us to remember all past samples, which require large storage and is not easy for implementation. Hence it is desirable to compute Π_k recursively once a new sample $\vec{\boldsymbol{X}}_k$ arrives. To achieve this, we further define the event

$$T_{i,k,s,l_1,l_2} = \{S = s, \lambda_{l_1-1} > k, \lambda_{l_1} \leq k, \lambda_{l_2+1} > k, \lambda_{l_2} \leq k, \theta = i\}$$

for $1 < s < L$, $l_1 \leq s$ and $l_2 \geq s$. Specially, $T_{i,k,1,1,l_2} = \{S = 1, \lambda_{l_2+1} > k, \lambda_{l_2} \leq k, \theta = i\}$ and $T_{i,k,L,l_1,L} = \{S = L, \lambda_{l_1-1} > k, \lambda_{l_1} \leq k, \theta = i\}$. From the definition, we know that event T_{i,k,s,l_1,l_2} denotes the event that the change with post-change distribution f_i firstly reaches sensor s and already propogates to sensors l_1 and l_2 at time k. In addition, we define the event that change has not happened yet as $T_{0,k} = \{\lambda > k\}$. In this change process setting, we can see that the underlying probability space Ω can be partitioned as

$$\Omega = \left(\bigcup_{s=1}^{L} \bigcup_{l_1=1}^{s} \bigcup_{l_2=s}^{L} \bigcup_{i \in \mathcal{I}} T_{i,k,s,l_1,l_2} \right) \bigcup T_{0,k}.$$

Then, we denote the posterior probability as $p_{i,k,s,l_1,l_2} := \mathbb{P}\{T_{i,k,s,l_1,l_2} | \mathcal{F}_k\}$ and $p_{0,k} = \mathbb{P}\{T_{0,k} | \mathcal{F}_k\}$. Using Bayesian rule, we can derive the updating rule for these posterior probabilities as

$$\begin{cases} p_{i,k,s,l_1,l_2} = \frac{N_{i,k,s,l_1,l_2}}{\sum\limits_{s=1}^{L} \sum\limits_{l_1=1}^{s} \sum\limits_{l_2=s}^{L} \sum\limits_{i \in I} N_{i,k,s,l_1,l_2} + N_{0,k}} \\ 1 < s < L, 1 \leq l_1 \leq s, s \leq l_2 \leq L, i \in \mathcal{I} \\ p_{0,k} = \frac{N_{0,k}}{\sum\limits_{s=1}^{L} \sum\limits_{l_1=1}^{s} \sum\limits_{l_2=s}^{L} \sum\limits_{i \in I} N_{i,k,s,l_1,l_2} + N_{0,k}} \end{cases}, \tag{3.5}$$

where N_{i,k,s,l_1,l_2} denotes the probability density

$$\begin{aligned} & d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{i,k,s,l_1,l_2}) \\ = & \left(\prod_{n=l_1}^{l_2} f_i(x_{k,n}) \right) \left(\prod_{n=1}^{l_1-1} f_0(x_{k,n}) \right) \left(\prod_{n=l_2+1}^{L} f_0(x_{k,n}) \right) \cdot \Bigg[p_{0,k-1} \kappa_s \rho (1-\rho_1)^{\mathbf{1}_{\{l_1 \neq 1\}}} (1-\rho_2)^{\mathbf{1}_{\{l_2 \neq L\}}} \rho_1^{s-l_1} \rho_2^{l_2-s} + \\ & \left(\sum_{n_1=l_1}^{s} \sum_{n_2=s}^{l_2} p_{i,k-1,s,n_1,n_2} (1-\rho_1)^{\mathbf{1}_{\{l_1 \neq 1\}}} (1-\rho_2)^{\mathbf{1}_{\{l_2 \neq L\}}} \rho_1^{n_1-l_1} \rho_2^{l_2-n_2} \right) \Bigg] \end{aligned} \tag{3.6}$$

and $N_{0,k}$ denotes the probability density

$$d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{0,k}) = p_{0,k-1}(1-\rho) \prod_{n=1}^{L} f_0(x_{k,n}). \tag{3.7}$$

For $k = 0$, we have $p_{0,0} = 1 - \rho$. For $l_1 \leq s \leq l_2$, we have

$$p_{i,0,s,l_1,l_2} = \kappa_s v_i \rho (1-\rho_1)^{\mathbf{1}_{\{l_1 \neq 1\}}} (1-\rho_2)^{\mathbf{1}_{\{l_2 \neq L\}}} \rho_1^{s-l_1} \rho_2^{l_2-s}.$$

Let $\mathbf{P}_k$ denote the 4-dimensional posterior probabilities tensor in which its elements are p_{i,k,s,l_1,l_2}. In $\mathbf{P}_k$, only elements satisfiing $l_1 \leq s$ and $l_2 \geq s$ can be non-zero values.

From (3.5) (3.6) and (3.7), we see that $\mathbf{P}_k$ can be computed from $\mathbf{P}_{k-1}$ and observation $\vec{\boldsymbol{X}}_k$ at time k. Hence, we have the recursive update formula for the posterior probabilities $\{\mathbf{P}_k, p_{0,k}\}$. More importantly, by the relationship between $\{\mathbf{P}_k, p_{0,k}\}$ and Π_k,

$$\begin{cases} \Pi_k^{(i)} = \sum\limits_{s=1}^{L} \sum\limits_{l_1=1}^{s} \sum\limits_{l_2=s}^{L} p_{i,k,s,l_1,l_2}, i \in \mathcal{I} \\ \Pi_k^{(0)} = p_{0,k} \end{cases}, \tag{3.8}$$

we can update Π_k recursively.

3.3 Optimal Multi-sensor Two-stage SCD rule

Given the updating rule of Π_k, (3.5) and (3.8), the optimal rule $(\tau_1^*, \tau_2^*, d^*)$ that minimizes (3.1) can be obtained by following similar steps as those in our recent work [73]. In particular, by converting the two-stage problem into two optimal single stopping time problems and solving them in reversed order, we can obtain the optimal SCD rule for the proposed two-stage sensor array SCD problem. Here, for completeness, we introduce the main steps of obtaining the optimal rule $(\tau_1^*, \tau_2^*, d^*)$.

To start, using Π_k, we can express the Bayesian cost (3.1) as

$$C(\delta) = \mathbb{E}\left[\sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right) + c_2\tau_2 + \mathbf{1}_{\{\tau_1<\infty\}} a\Pi_{\tau_1}^{(0)} + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}} \sum_{j=0}^{I} \mathbf{1}_{\{d=j\}} B_j(\Pi_{\tau_1+\tau_2})\right],$$

where $B_j(\Pi) = \sum_{i\in\mathcal{I}_0} b_{ij}\Pi^{(i)}$ is the misdiagnosis cost associated with the decision $d = j$. Therefore, $B(\Pi) = \min\limits_{j\in\mathcal{I}_0} B_j(\Pi)$ is the smallest misdiagnosis cost can be achieved at time k. As a result, the optimal identification decision is $d^* = \arg\min_{j\in\mathcal{I}_0} B_j(\Pi)$. Using this result, we have $C(\tau_1, \tau_2, d^*) = \mathbb{E}[C_1(\tau_1) + C_2(\Pi_{\tau_1}, \tau_2)]$, where

$$C_1(\tau_1) = \sum_{n=0}^{\tau_1-1} c_1\left(1-\Pi_n^{(0)}\right) + \mathbf{1}_{\{\tau_1<\infty\}} a\Pi_{\tau_1}^{(0)}$$

and $C_2(\Pi_{\tau_1}, \tau_2) = c_2\tau_2 + \mathbf{1}_{\{\tau_1+\tau_2<\infty\}}B\left(\Pi_{\tau_1+\tau_2}\right)$ are the cost functions of the change detection stage and distribution identification stage respectively. Then we have the minimal expected cost for the SCD process,

$$\begin{aligned} C(\tau_1^*, \tau_2^*, d^*) &= \min_{\tau_1, \tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\left[C_1(\tau_1) + C_2(\tau_1, \tau_2)\right] \\ &= \min_{\tau_1, \tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\Big[C_1(\tau_1) + \mathbb{E}\left[C_2(\tau_2)|\mathbf{P}_{\tau_1}, p_{0,\tau_1}\right]\Big] \\ &= \min_{\tau_1 \in \mathbb{F}} \mathbb{E}\left[C_1(\tau_1) + \min_{\tau_1+\tau_2 \in \mathbb{F}} \mathbb{E}\left[C_2(\tau_2)|\mathbf{P}_{\tau_1}, p_{0,\tau_1}\right]\right]. \end{aligned} \tag{3.9}$$

By (3.9), the two-stage stopping time problem becomes two ordered optimal single stopping time problems. The first one is for the identification stage, its goal is finding the optimal τ_2 which minimizes $\mathbb{E}[C_2(\tau_2)|\mathbf{P}_{\tau_1}, p_{0,\tau_1}]$ for any given τ_1, $\mathbf{P}_{\tau_1}$ and p_{0,τ_1}. The second single stopping time problem is to find the best stopping rule for the detection stage. From the last line of (3.9), we can find an optimal τ_1 to minimize the expected cost for the whole SCD process if the optimal rule for τ_2 is known. Therefore, we will firstly find the optimal rule for the identification stage, then select the optimal stopping time for the detection stage. DP is a good way to solve optimal single stopping time problems. With the expression C_1 and C_2, we can built the cost-to-go functions of the two optimal single stopping time problems. In particular, for the identification stage, let $\{\tilde{\boldsymbol{P}}, \tilde{p}_0\}$ be the posterior probabilities at time next to the time of $\{\boldsymbol{P}, p_0\}$. The infinite-horizon cost-to-go function for the DP process of the identification stage can be obtained by solving $V(\boldsymbol{P}, p_0) = \min(B(\boldsymbol{P}, p_0), c_2 + \mathbb{E}[V(\tilde{\boldsymbol{P}}, \tilde{p}_0)|\boldsymbol{P}, p_0])$. This implies that we should make an identification when the expected cost for keep observing exceeds the cost of making identification immediately. In addition, the optimal identification decision is $d = \arg\min_{j\in\mathcal{I}_0} B_j(\boldsymbol{P})$. Similarly, in the change detection stage, for any $\{\boldsymbol{P}, p_0\}$, the infinite-horizon cost-to-go function for the detection stage satisfies the following Bellman equation $W(\boldsymbol{P}, p_0) = \min(ap_0 + V(\boldsymbol{P}, p_0), c_1(1-p_0) + \mathbb{E}[W(\tilde{\boldsymbol{P}}, \tilde{p}_0)|\boldsymbol{P}, p_0])$. From this, we know that we should raise a change alarm when the expected cost of observing more data exceeds the cost of declaring a change has happened.

The cost-to-go functions $V(\boldsymbol{P})$ and $W(\boldsymbol{P})$ and the optimal stopping times can be calculated using DP. However, the size of the state space increases exponentially with L and I. With such a high complexity, the optimal solution is hard to implement.

3.4 Low-complexity Rule

Same as other DP-based methods, the complexity of the optimal solution is very high, even with an array with only two sensors and two post-change distributions. To address this issue, we propose a threshold SCD rule that is easy to implement. Moreover, we will prove this threshold SCD rule is asymptotically optimal as c_1 and c_2 go to zero. The main idea of the proof is similar to the proof for the single sensor case considered in [73]. However, the most important step of the proof, i.e., analyzing the convergence of the LLR process, becomes much more complicated in the sensor array case. In this section, we will introduce the main steps of the asymptotic optimality analysis and underline the proof details of the LLR convergence (Proposition 3.1).

3.4.1 Threshold SCD Rule

Here, we introduce the proposed low complexity two-stage SCD rule. The low complexity rule is a threshold rule. In particular, it is characterized by a set of thresholds $\{A, \vec{\boldsymbol{B}}\}$ where $\vec{\boldsymbol{B}} = (B_0, B_1, B_2, ..., B_I)$. A and all elements in $\vec{\boldsymbol{B}}$ are strictly positive constants. Using these thresholds, the proposed threshold rule $\delta_T = (\tau_A, \tau_{\vec{\boldsymbol{B}}}, d_{\vec{\boldsymbol{B}}})$ is defined as

$$\begin{cases} \tau_A := \inf\{k \geq 1, \Pi_k^{(0)} < 1/(1+A)\}, \\ \tau_{\vec{\boldsymbol{B}}} := \min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}, \\ \tau_{\vec{\boldsymbol{B}}}^{(i)} := \inf\{k \geq 1, \Pi_k^{(i)} > 1/(1+B_i)\} - \tau_A, \\ d_{\vec{\boldsymbol{B}}} := \arg\min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}. \end{cases} \tag{3.10}$$

In this threshold SCD rule, the first stopping time τ_A is the first time $\Pi_k^{(0)}$ falls below the threshold $1/(1+A)$. After τ_A, the rule turns to check the posterior probabilities $\Pi_k^{(i)}$ for all $i \in \mathcal{I}_0$. It will stop immediately if any threshold $1/(1+B_i)$ is exceeded. The identification decision depends on which threshold is passed. In order to guarantee that this rule is in the two-stage SCD rule space Δ, it must satisfy $\tau_{\vec{\boldsymbol{B}}} \geq 0$. This condition can be satisfied by choosing appropriate A and $\vec{\boldsymbol{B}}$, as will be introduced in Section 3.4.3.

For $i \in \mathcal{I}_0$ and $k \geq 1$, define the logarithm of the odds-ratio process as

$$\pi_k^{(i)} := \log \frac{\Pi_k^{(i)}}{1 - \Pi_k^{(i)}} = -\log \left[\sum_{j \in \mathcal{I}_0 \setminus \{i\}} \exp(-\Lambda_k(i,j)) \right].$$

Using $\pi_k^{(i)}$, δ_T can be expressed as:

$$\begin{cases} \tau_A = \inf\left\{k \geq 1, \dfrac{1-\Pi_k^{(0)}}{\Pi_k^{(0)}} > A\right\} = \inf\{k \geq 1, \pi_k^{(0)} < -\log A\}, \\ \tau_{\vec{\boldsymbol{B}}} = \min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}, \\ \tau_{\vec{\boldsymbol{B}}}^{(i)} = \inf\left\{k \geq 1, \dfrac{1-\Pi_k^{(i)}}{\Pi_k^{(i)}} < B_i\right\} - \tau_A = \inf\{k \geq 1, \pi_k^{(i)} > -\log B_i\} - \tau_A, \\ d_{\vec{\boldsymbol{B}}} = \arg\min\limits_{i \in \mathcal{I}_0} \tau_{\vec{\boldsymbol{B}}}^{(i)}. \end{cases} \tag{3.11}$$

The complexity of the threshold rule (3.10) is very low. After obtaining a new sample, we only need to update the posterior probabilities using the recursive formula (3.5), and then compare them with the thresholds. In the following parts, we will show that this rule is asymptotically optimal as c_1 and c_2 go to zero.

3.4.2 Convergence of LLR Process

By (3.2) and (3.4), we can see that

$$\Lambda_k(i,j) = \log \alpha_k^{(i)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k) - \log \alpha_k^{(j)}(\vec{\boldsymbol{X}}_1, \vec{\boldsymbol{X}}_2, \ldots, \vec{\boldsymbol{X}}_k).$$

For $i \in \mathcal{I}$ and time $k > 0$, we define

$$H_k^{(i)} = \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \left[\left(\prod_{n=1}^{n_s-1} \left(\frac{(1-\rho) f_0(x_{n,s})}{(1-\rho_1)(1-\rho_2) f_i(x_{n,s})} \right) \right) \cdot \psi_{s-1}^{(i)}(k, n_s) \phi_{s+1}^{(i)}(k, n_s) \right]$$

where

$$\begin{cases} \psi_l^{(i)}\left(k, n_{l+1}\right) = \prod_{n=1}^{k} \left[(1-\rho_1) \prod_{t=1}^{l} \frac{f_0(x_{n,t})}{f_i(x_{n,t})} \right] + \rho_1 \sum_{n_l = n_{l+1}}^{k} \prod_{n=1}^{n_l - 1} \frac{f_0(x_{n,l})}{f_i(x_{n,l})} \psi_{l-1}^{(i)}\left(k, n_l\right), l \geq 1 \\ \phi_l^{(i)}\left(k, n_{l-1}\right) = \prod_{n=1}^{k} \left[(1-\rho_2) \prod_{t=l}^{L} \frac{f_0(x_{n,t})}{f_i(x_{n,t})} \right] + \rho_2 \sum_{n_l = n_{l-1}}^{k} \prod_{n=1}^{n_l - 1} \frac{f_0(x_{n,l})}{f_i(x_{n,l})} \phi_{l+1}^{(i)}\left(k, n_l\right), l \leq L \end{cases}.$$

In addition, $\phi_{L+1}^{(i)}\left(k, n_L\right) = (1-\rho_2)^{n_L - 1}$ and $\psi_0^{(i)}\left(k, n_1\right) = (1-\rho_1)^{n_1 - 1}$. Therefore, we can express $\log \alpha_k^{(i)}$ as

$$\begin{cases} \log \alpha_k^{(i)} = \log[v_i \rho (1-\rho)] + \log \left(\prod_{l=1}^{L} \prod_{m=1}^{k} f_i(x_{m,l}) \right) + \log H_k^{(i)}, \text{for } i \in \mathcal{I} \\ \log \alpha_k^{(0)} = (k+1) \log(1-\rho) + \log \left(\prod_{l=1}^{L} \prod_{m=1}^{k} f_0(x_{m,l}) \right). \end{cases}$$

Let $q(j,i)$ be the KL divergence from f_i to f_j. We define the following condition for $i, j \in \mathcal{I}$.

Condition 1. $\log(1-\rho) + q(j,i) - q(j,0) \geq 0$ or $q(j,i) - q(j,0) \leq 0$.

The next proposition describes the limit of $\log H_k^{(i)}/k$ as $k \to \infty$.

Proposition 3.1. For any $i, j \in \mathcal{I}$, if Condition 1 is satisfied,

$$\frac{1}{k} \log H_k^{(i)} \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} h(i,j) \tag{3.12}$$

where $h(i,j) = \left(\log(1-\rho) + L(q(j,i) - q(j,0))\right)_+$.

Proof. Please see Appendix B.1. □

3.4.3 Asymptotic Optimality

Once we show the convergence of $\log H_k^{(i)}/k$, we can proceed to show the asymptotic optimality of the threshold rule. The main steps on this proof are: (1) Obtain approximations of the delay, false alarm probability and misdiagnosis probability, which leads to the expression of the Bayesian cost of the threshold rule, $C(\delta_T)$, w.r.t. A and $\boldsymbol{B}$; (2) Select the optimal A and $\boldsymbol{B}$ that can minimize $C(\delta_T)$; (3) Prove that $C(\delta_T, A_{opt}, \boldsymbol{B}_{opt})$ achieves the lower bound of the Bayesian cost for arbitrary two-stage SCD rule when c_1 and c_2 go to 0.

For any $i \in \mathcal{I}$, define

$$w(i,j) = \begin{cases} Lq(i,j) - h(j,i), j \in \mathcal{I} \\ Lq(i,0) - \log(1-\rho), j = 0 \end{cases}. \tag{3.13}$$

If the first affected sensor is unknown, and Condition 1 is satisfied for $i, j \in \mathcal{I}$, $h(i,j)$ can be calculated as in Proposition 3.1. As introduced in [73], the approximation of delay can be expressed as

$$\begin{cases} \mathbb{E}_i\left[(\tau_{\vec{\boldsymbol{B}}} + \tau_A - \lambda)_+\right] \xrightarrow[B_i \to 0]{\mathbb{P}_i - a.s.} \frac{-\log B_i}{w(i)}, \text{ for } i \in \mathcal{I} \\ \mathbb{E}_i\left[(\tau_A - \lambda)_+\right] \xrightarrow[A \to \infty]{\mathbb{P}_i - a.s.} \frac{\log A}{w(i,0)}, \text{ for } i \in \mathcal{I} \end{cases}, \tag{3.14}$$

where $w(i) = w(i, j(i))$, $j(i) = \underset{j \in \mathcal{I}_0 \backslash \{i\}}{\arg\min}\, w(i,j)$. In addition, the false alarm and misdiagnosis probability can be approximated as $\frac{k_a}{1+A}$ and $\sum\limits_{i \in \mathcal{I}} v_i B_i k_i$, respectively. Here $k_a = a$ and $k_i = \max_{j \in \mathcal{I}_0 \backslash \{i\}} b_{ji}$. Therefore, the Bayesian cost of the threshold rule can be approximated as

$$C^{(c_2)}(\delta_T) = c_2 \sum_{i \in \mathcal{I}} v_i \left(\frac{-\log(B_i)}{w(i)} \right) + \sum_{i \in \mathcal{I}} v_i B_i k_i + c_2 \left(\frac{1}{r} - 1 \right) \sum_{i \in \mathcal{I}} \frac{v_i \log A}{w(i,0)} + \frac{k_a}{1+A}. \tag{3.15}$$

By minimizing (3.15) w.r.t A and $\vec{\boldsymbol{B}}$, we get the optimal A and $\vec{\boldsymbol{B}}$ as

$$\begin{cases} A_{opt} \approx \frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}}\frac{v_i}{w(i,0)}} - 2, \\ B_{i,opt} = \frac{c_2}{k_i w(i)}, i \in \mathcal{I}. \end{cases} \tag{3.16}$$

The Bayesian cost for the optimal threshold SCD rule is

$$C^{(c_2)}(\delta_T^*) = c_2 \sum_{i\in\mathcal{I}} \frac{-v_i}{w(i)} \log\left(\frac{c_2}{k_i w(i)}\right) + \sum_{i\in\mathcal{I}} \frac{v_i c_2}{w(i)} + c_2\left(\frac{1}{r}-1\right) \sum_{i\in\mathcal{I}} \frac{v_i}{w(i,0)} \log\left(\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}}\frac{v_i}{w(i,0)}} - 2\right) + k_a \frac{1}{\frac{k_a}{c_2(\frac{1}{r}-1)\sum\limits_{i\in\mathcal{I}}\frac{v_i}{w(i,0)}} - 1}. \tag{3.17}$$

Now we need to check if the condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is satisfied. By the threshold rule (3.10), we know that $\tau_{A_{opt}}$ is the first time $\sum_{i\in\mathcal{I}} \Pi_n^{(i)} = 1 - \Pi_n^{(0)}$ exceeds the threshold $1 - 1/(1 + A_{opt})$. Also, $\tau^{(i)}_{\vec{\boldsymbol{B}}_{opt}} + \tau_{A_{opt}}$ is the first time for $\Pi_n^{(i)}$ exceeds the threshold $1/(1 + B_{i,opt})$. So if

$$1 - \frac{1}{1 + A_{opt}} < \frac{1}{1 + B_{i,opt}} \tag{3.18}$$

for all $i \in \mathcal{I}$, it is guaranteed that the threshold $\vec{\boldsymbol{B}}$ can not be reached before threshold A, namely, $\tau_{\vec{\boldsymbol{B}}} \geq 0$. After plugging the explicit expressions of the optimal thresholds (3.16) in inequality (3.18), we know that a sufficient condition of $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is

$$0 < r \leq \min_{i\in\mathcal{I}} \frac{1}{1 + \frac{k_a}{k_i w(i) \sum\limits_{i\in\mathcal{I}} \frac{v_i}{w(i,0)}}}. \tag{3.19}$$

If the value of r satisfies (3.19), condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ is satisfied. However, for the case (3.19) is not satisfied, we need to change the threshold accordingly as

$$\begin{cases} A' = A_{opt}, \\ B_i' = B_{i,opt}\frac{k_i}{\eta}, i \in \mathcal{I} \end{cases} \tag{3.20}$$

where η is a constant such that

$$r = \min_{i \in \mathcal{I}} \frac{1}{1 + \frac{k_a}{\eta w(i) \sum\limits_{i \in \mathcal{I}} \frac{v_i}{w(i,0)}}}.$$

We can see that with A' and $\vec{\boldsymbol{B}}'_{opt}$, condition $\tau_{\vec{\boldsymbol{B}}} \geq 0$ still is satisfied even if (3.19) is not satisfied. In this case, the Bayesian cost of the rule $\delta'_T = (\tau_{A'}, \tau_{\vec{\boldsymbol{B}}'}, d')$ is

$$\begin{aligned} C^{(c_2)}(\delta'_T) &= C^{(c_2)}(\delta^*_T) - c_2 \sum_{i \in \mathcal{I}} \log\left(\frac{k_i}{\eta}\right) \frac{v_i}{w(i)} \\ &+ \sum_{i \in \mathcal{I}} v_i B_{i,opt} \left(\frac{k_i^2}{\eta} - k_i\right). \end{aligned} \tag{3.21}$$

Since k_i, $w(i)$ and η are constants, the last two terms in (3.21) decay much faster than $C^{(c_2)}(\delta^*_T)$ as $c_2 \to 0$. This implies that the difference between the cost calculated by (3.17) and (3.21) is negligible as $c_2 \to 0$.

Finally, in the following proposition, we prove that (3.17) (also true for (3.21) if (3.19) is not satisfied) is the lowest Bayesian cost any two-stage SCD rule can achieve when c_1 and c_2 go to 0. In other words, the proposed threshold rule is asymptotically optimal.

Proposition 3.2. If $\delta_T = (\tau_{A_T}, \tau_{\vec{\boldsymbol{B}}_T}, d_T)$ is a threshold two-stage SCD rule with thresholds as (3.16), then for any given fixed $r := c_2/c_1$ we have

$$\lim_{c_2 \to 0} \frac{\inf_{\delta \in \Delta} C^{(c_2)}(\delta)}{C^{(c_2)}(\delta_T)} \geq 1.$$

The main steps to prove Proposition 3.2 are as follows: (1). Derive a lower bound of the Baysian cost for any possible SCD rule; (2) Prove the proposed threshold SCD rule can achieve the lower bound as c_1 and c_2 go to zero. For more details of the proof, please refer to [73]. Note that, since Proposition 3.2 is proved based on Proposition 3.1, Condition 1 is also necessary for Proposition 3.2.

From the results of asymptotic analysis in Proposition 3.1 and equation (3.14) and (3.16), we can see that the prior probabilities of first affected sensor $\{\kappa_s\}_{1\leq s\leq L}$ do not affect the asymptotic behaviors of the threshold rule. Therefore, in the case when $\{\kappa_s\}_{1\leq s\leq L}$ are unknown, we can just set $\kappa_s = 1/L$ for all $1 \leq s \leq L$. Even if the true prior probabilities are not $1/L$, it will not affect the asymptotic optimality of the threshold SCD rule. In addition, the Condition 1 is not a strong condition because it just rules out the case $0 < q(j,i) - q(j,0) < -\log(1-\rho)$ in which the change is very hard to detect and identify. Considering the change is typically rare, i.e. ρ is small and the range $[0, -\log(1-\rho)]$ is narrow, Condition 1 can be satisfied in most cases.

3.4.4 Special Case: When the First Affected Sensor is Known

As discussed above, when the first affected sensor S is an unknown random variable, Condition 1 is necessary for the asymptotic optimality of the multi-sensor threshold SCD rule. In this section, we will show that, when the first affected sensor is fixed and known, the multi-sensor threshold SCD rule is asymptotically optimal with no additional condition.

When the first affected sensor is fixed and known, one element of $\vec{\boldsymbol{K}}$ is 1 and all other elements are 0. Without loss of generality, we assume that the first affected sensor is the sth sensor, i.e., $\kappa_s = 1$. With this additional assumption, the computations in the previous section can be further simplified and we can prove stronger asymptotic optimality results. In particular, for any time $k \geq 1$, Π_k can be directly calculated as

$$\Pi_k^{(i)} = \frac{\alpha_k^{(i)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,...,\vec{\boldsymbol{X}}_k)}{\sum_{j\in\mathcal{I}_0}\alpha_k^{(j)}(\vec{\boldsymbol{X}}_1,\vec{\boldsymbol{X}}_2,...,\vec{\boldsymbol{X}}_k)} \tag{3.22}$$

where

$$\begin{cases} \alpha_k^{(0)} = (1-\rho)^{k+1}\prod_{l=1}^{L}\prod_{n=1}^{k} f_0(x_{n,l}) \\ \alpha_k^{(i)} = v_i\rho\sum_{n_s=0}^{k}\left[(1-\rho)^{n_s}\left(\prod_{n=1}^{n_s-1} f_0(x_{n,s})\right)\cdot\left(\prod_{n=max(n_s,1)}^{k} f_i(x_{n,s})\right)\Psi_{s-1}^{(i)}(k,n_s)\Phi_{s+1}^{(i)}(k,n_s)\right]. \end{cases} \tag{3.23}$$

For $i \in \mathcal{I}$, we define

$$H_k^{(i)} = \sum_{n_s=0}^{k} \left[\left(\prod_{n=1}^{n_s-1} \left(\tfrac{(1-\rho) f_0(x_{n,s})}{(1-\rho_1)(1-\rho_2) f_i(x_{n,s})} \right) \right) \cdot \psi_{s-1}^{(i)}(k, n_s) \phi_{s+1}^{(i)}(k, n_s) \right]. \tag{3.24}$$

Define

$$\eta_l(i,j) = \begin{cases} \log\left[\tfrac{1-\rho}{(1-\rho_1)^{1\{s\neq 1\}}(1-\rho_2)^{1\{s\neq L\}}}\right] + q(j,i) - q(j,0), l = s \\ \log(1-\rho_1) + q(j,i) - q(j,0), l = 1 \text{ and } s \neq 1 \\ \log(1-\rho_2) + q(j,i) - q(j,0), l = L \text{ and } s \neq L \\ q(j,i) - q(j,0), \text{otherwise.} \end{cases} . \tag{3.25}$$

For any $i, j \in \mathcal{I}$, according to the value of $\eta_l(i,j)$, we divide the sensor labels $1 \leq l \leq L$ into several consecutive groups (the labels in each group are consecutive). The grouping rule is described in Algorithm 1. After implementing Algorithm 1 for the case $i, j \in \mathcal{I}$, we will have $M(i,j) + N(i,j) + 1$ consecutive groups

$$\begin{aligned} &\{a_1^m(i,j), a_1^m(i,j)+1, \ldots, a_2^m(i,j)\}_{1 \leq m \leq M(i,j)}, \\ &\{a_2^{M(i,j)}(i,j)+1, a_2^{M(i,j)}(i,j)+2, \ldots, b_2^{N(i,j)}(i,j)-1\}, \\ &\{b_2^n(i,j), \ldots, b_1^n(i,j)-1, b_1^n(i,j)\}_{N(i,j) \geq n \geq 1}. \end{aligned}$$

The next proposition describes the limit of $\log H_k^{(i)}/k$ as $k \to \infty$.

Proposition 3.3. For any $i, j \in \mathcal{I}$,

$$\tfrac{\log H_k^{(i)}}{k} \xrightarrow[k\to\infty]{\mathbb{P}_j - a.s.} h(i,j) \tag{3.26}$$

where

$$h(i,j) = \sum_{l=1}^{a_2^{M(i,j)}(i,j)} \eta_l(i,j) + \sum_{l=b_2^{N(i,j)}(i,j)}^{L} \eta_l(i,j) + \left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \eta_l(i,j) \right)_+ . \tag{3.27}$$

Proof. Please see Appendix B.2. □

Algorithm 1: Grouping the sensors

```
Initialize $a_1^1(i,j) = 1$, $a_2^0(i,j) = 0$, $b_1^1(i,j) = L$, $b_2^0(i,j) = L+1$, $m = 1$, $n = 1$;
for $l$=1,2,...,s-2,s-1 do
    if $\sum_{k=a_1^m(i,j)}^{l} \eta_k(i,j) \geq 0$ then
        $a_2^m(i,j) = l$, $a_1^{m+1}(i,j) = l+1$;
        $m+ = 1$;
end
for $l$=L,L-1,...,s+2, s+1 do
    if $\sum_{k=b_1^m(i,j)}^{l} \eta_k(i,j) \geq 0$ then
        $b_2^n(i,j) = l$, $b_1^{n+1}(i,j) = l-1$;
        $n+ = 1$;
    $M(i,j) = m-1, N(i,j) = n-1$
end
```

Then following the same steps of Section 3.4.3, we can prove that the multi-sensor threshold SCD rule is asymptotically optimal as c_1 and c_2 go to zero. Plugging (3.27) in (3.13), (3.16) and (3.17), we will have the optimal threshold and the corresponding Bayesian cost. Different from the asymptotic optimality for the general case in Section 3.4.3, in this special case when the first affected sensor is known, the asymptotic optimality does not need any additional condition. This is because knowing first affected sensor makes the structure of $H(i,j)$ easier and thus we can prove Proposition 3 true in general. Moreover, if Condition 1 is true for any $i, j \in \mathcal{I}$, we can easily check that the $h(i,j)$ in Proposition 3.1 and equation (3.3) are equivalent following Algorithm 1. With equivalent $h(i,j)$, $w(i,j)$ and the limit of cost function in (3.17) will also be equivalent. This indicates that the performances of the general case and special case will tend to be the same as c_1 and c_2 go to zero.

3.5 Extension of the proposed SCD rules to 2D sensor array Case

In the Section 3.4, we studied the SCD problem in a linear sensor array. In this section, we extend our study to a 2D lattice array scenario.

3.5.1 Change Propagation Model on 2D Lattice

Consider an 2D lattice with vertices $\{V_{a,b}\}_{1\leq a\leq H,1\leq b\leq W}$, where $V_{a,b}$ denotes the vertex at the ith row and the jth column of the lattice. An edge exists between vertex pair $(V_{a,b}, V_{c,d})$ if $|a-c|+|b-d|=1$. A change could happen at any single vertex first and then start to diffuse outward via the edges. At time k, the sensors collect the data samples $\vec{\boldsymbol{X}}_k = (x_{k,1,1}, x_{k,1,1}, ..., x_{k,H,W})$. Let $S=(S_1,S_2)$ be index of the sensor where the change happens first, the prior probability $\mathcal{P}(S=(a,b))=\kappa_{a,b}$ is known. We denote $(\kappa_{1,1},\kappa_{1,2},\ldots,\kappa_{H,W})$ as $\boldsymbol{K}$. The change propagation process is characterized by the distance between the target sensor and the first affected sensor and follows a geometric distribution. let $\mathcal{V}(a,b,r)$ be vertex layer whose distance to $V_{a,b}$ is exactly r, i.e., $\mathcal{V}(a,b,r)=\{V_{m,n}||a-m|+|b-n|=r\}$. The change will first propagate from V_{S_1,S_2} to all the vertex in $\mathcal{V}(S_1,S_2,1)$ at time $\lambda_{\mathcal{V}(S_1,S_2,1)}$, then to all vertices in $\mathcal{V}(S_1,S_1,2)$ at time $\lambda_{\mathcal{V}(S_1,S_2,2)}$ and so on. The propagation of the change in the 2D lattice follows a geometric distribution as

$$\mathbb{P}\left[\lambda_{\mathcal{V}(a,b,r+1)} = k_1+k_2|\lambda_{\mathcal{V}(a,b,r)}=k_1, S=(a,b)\right] = \rho_1(1-\rho_1)^{k_2}, k_2\geq 0 \tag{3.28}$$

where ρ_1 is the probability of the change propagates outward the next layer. As an example of the 2D lattice sensor array, we illustrate the change propagation process in a 5×5 lattice sensor array in Fig. 3.1. On each vertex, a sensor is implemented to collect data. $x_{k,a,b}$ denotes the data sample collected by the sensor at $V_{a,b}$ at time k. For the convenience of expression, we denote

$$\begin{cases} \mathcal{C}(S_1,S_2,r)=\{(a,b)||a-S_1|+|b-S_2|=r, 1\leq a\leq H, 1\leq b\leq W\}, \\ \mathcal{I}(S_1,S_2,r)=\{(a,b)||a-S_1|+|b-S_2|\leq r, 1\leq a\leq H, 1\leq b\leq W\}, \\ \mathcal{O}(S_1,S_2,r)=\{(a,b)||a-S_1|+|b-S_2|> r, 1\leq a\leq H, 1\leq b\leq W\}, \\ R(S_1,S_2)=max_{1\leq a\leq H, 1\leq b\leq W}(|a-S_1|+|b-S_2|). \end{cases}$$

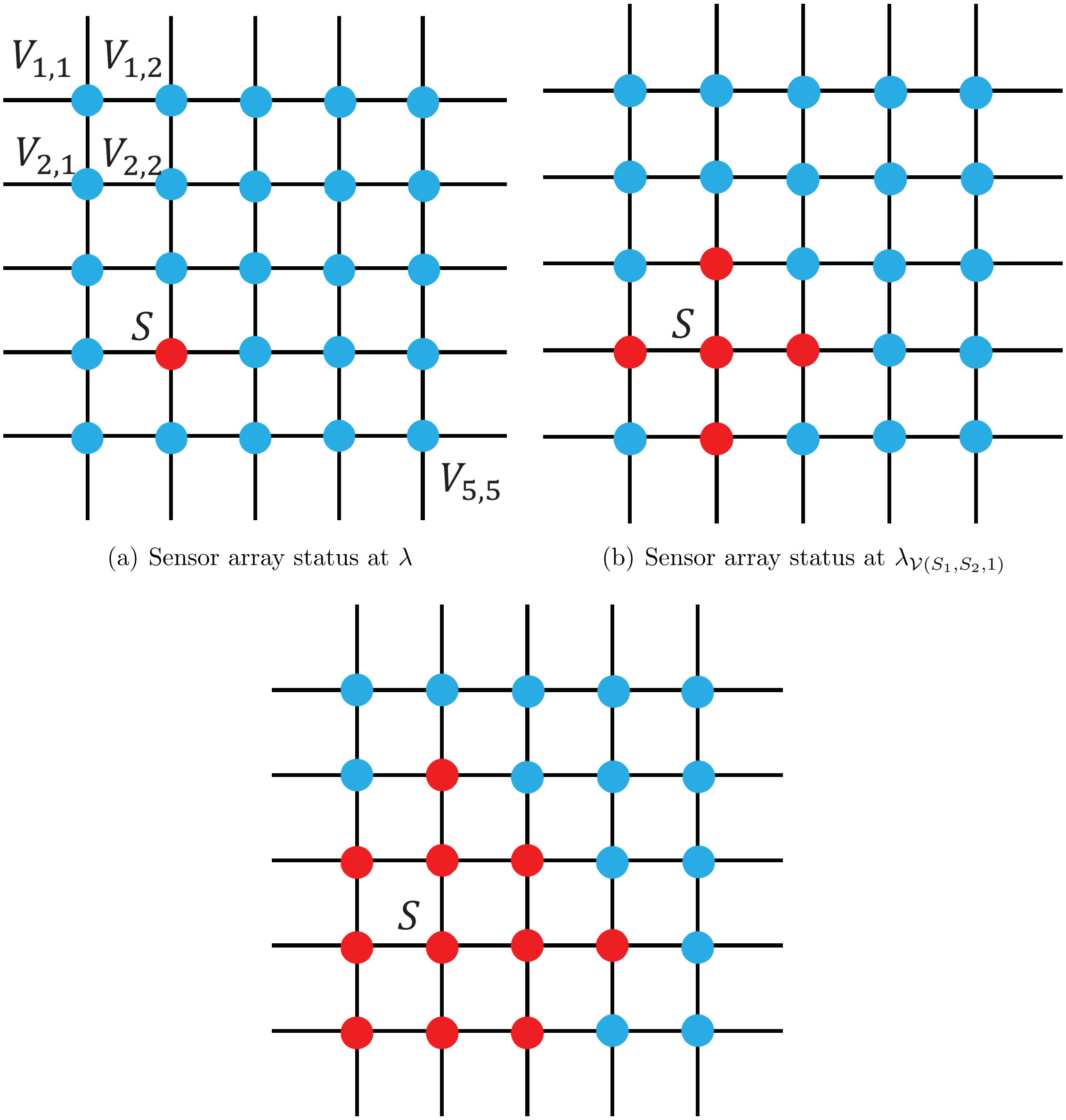
Now we have a new 2D lattice sensor array and a corresponding change propagation model. The other parts in the SCD problem formulation, such as the prior distribution of the change time λ, observation model and etc., are the same as in Section 3.1.

3.5.2 Posterior Probability Analysis

In the SCD problem with the 2D lattice sensor array, the posterior probability Π_k defined in (3.2) still plays a key role. However, the calculation of $\alpha_k^{(i)}$ in (3.3) will be replaced as

$$\begin{cases} \alpha_k^{(0)}=(1-\rho)^{k+1}\prod_{a=1}^{H}\prod_{b=1}^{W}\prod_{n=1}^{k} f_0(x_{n,a,b}) \\ \alpha_k^{(i)}=\sum_{S_1=1}^{H}\sum_{S_2=1}^{W}\kappa_{S_1,S_2}v_i\rho\sum_{n_0=0}^{k}\Bigg[(1-\rho)^{n_0}\left(\prod_{n=1}^{n_0-1} f_0(x_{n,S_1,S_2})\right)\cdot \\ \left(\prod_{n=max(n_0,1)}^{k} f_i(x_{n,S_1,S_2})\right)\Psi_1^{(i)}(k,n_0,S_1,S_2)\Bigg] \\ \Psi_{l+1}^{(i)}(k,n_l,S_1,S_2)=(1-\rho_1)^{k-n_l+1}\prod_{(a,b)\in\mathcal{O}(S_1,S_2,l)}\prod_{n=1}^{k} f_0(x_{n,a,b})+ \quad \cdot \\ \rho_1\sum_{n_{l+1}=n_l}^{k}\Bigg[(1-\rho_1)^{n_{l+1}-n_l}\prod_{(a,b)\in\mathcal{C}(S_1,S_2,l+1)}\Bigg(\left(\prod_{n=1}^{n_{l+1}-1} f_0(x_{n,a,b})\right)\cdot \\ \left(\prod_{n=n_{l+1}}^{k} f_i(x_{n,a,b})\right)\Bigg)\Psi_{l+2}^{(i)}(k,n_{l+1},S_1,S_2)\Bigg], R(S_1,S_2)>l\geq 0 \\ \Psi_{R(S_1,S_2)+1}^{(i)}(k,n_l)=1 \end{cases}$$

Similar to Section 3.2, we want to compute Π_k recursively once a new sample $\vec{\boldsymbol{X}}_k$ arrives



(a) Sensor array status at λ

(b) Sensor array status at $\lambda_{\mathcal{V}(S_1,S_2,1)}$

(c) Sensor array status at $\lambda_{\mathcal{V}(S_1,S_2,2)}$

Figure 3.1: Change propagation model of the 2D lattice sensor array

rather than remembering all historical data samples. To this end, we further define the event

$$T_{i,k,a,b,r} = \{S_1 = a, S_2 = b, \lambda_{\mathcal{V}(a,b,r+1)} > k, \lambda_{\mathcal{V}(a,b,r)} \leq k, \theta = i\}$$

for $1 \leq a \leq H, 1 \leq b \leq W, 1 \leq r \leq R(a,b), i \in \mathcal{I}$. From the definition, we know that event T_{i,k,s,l_1,l_2} denotes the event that the change with post-change distribution f_i firstly reaches vertex $V_{a,b}$ and already propogates to vertices $\mathcal{V}(a,b,r)$ at time k. In addition, we define the event that change has not happened yet as $T_{0,k} = \{\lambda > k\}$. In this change process setting, we can see that the underlying probability space Ω can be partitioned as

$$\Omega = \left(\bigcup_{a=1}^{H} \bigcup_{b=1}^{W} \bigcup_{r=0}^{R(a,b)} \bigcup_{i \in \mathcal{I}} T_{i,k,a,b,r} \right) \bigcup T_{0,k}.$$

Then, we denote the posterior probability as $p_{i,k,a,b,r} := \mathbb{P}\{T_{i,k,a,b,r} | \mathcal{F}_k\}$ and $p_{0,k} = \mathbb{P}\{T_{0,k} | \mathcal{F}_k\}$. Using Bayesian rule, we can derive the updating rule for these posterior probabilities as

$$\begin{cases} p_{i,k,a,b,r} = \frac{N_{i,k,a,b,r}}{\sum_{a=1}^{H} \sum_{b=1}^{W} \sum_{r=0}^{R(a,b)} \sum_{i \in I} N_{i,k,a,b,r} + N_{0,k}}, 1 \leq a \leq H, 1 \leq b \leq W, 0 \leq r \leq R(a,b), i \in \mathcal{I} \\ p_{0,k} = \frac{N_{0,k}}{\sum_{a=1}^{H} \sum_{b=1}^{W} \sum_{r=1}^{R(a,b)} \sum_{i \in I} N_{i,k,a,b,r} + N_{0,k}} \end{cases} \tag{3.29}$$

where $N_{i,k,a,b,r}$ denotes the probability density

$$\begin{gathered} d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{i,k,a,b,r}) = \left(\prod_{(m,n) \in \mathcal{I}(a,b,r)} f_i(x_{m,n,k}) \right) \left(\prod_{(a,b) \in \mathcal{O}(a,b,r)} f_0(x_{m,n,k}) \right) \cdot \\ \left(p_{0,k-1} \kappa_{a,b} \rho (1 - \rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \rho_1^r + \sum_{r_{k-1}=1}^{r} p_{i,k,a,b,r_{k-1}} \rho_1^{r - r_{k-1}} (1 - \rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \right) \end{gathered} \tag{3.30}$$

and $N_{0,k}$ denotes the probability density

$$d\mathbb{P}((\vec{\boldsymbol{X}}_1, \ldots, \vec{\boldsymbol{X}}_k), T_{0,k}) = p_{0,k-1}(1 - \rho) \prod_{a=1}^{H} \prod_{b=1}^{W} f_0(x_{k,a,b}). \tag{3.31}$$

For $k=0$, we have $p_{0,0} = 1-\rho$. For $1 \le a \le H, 1 \le b \le W, 0 \le r \le R(a,b), i \in \mathcal{I}$, we have

$$p_{i,k,a,b,r} = \kappa_{a,b} v_i \rho (1-\rho_1)^{\mathbf{1}_{\{r \neq R(a,b)\}}} \rho_1^r.$$

Let $\mathbf{P}_k$ denote the 4-dimensional posterior probabilities tensor in which its elements are $p_{i,k,a,b,r}$. In $\mathbf{P}_k$, only elements satisfying $1 \le r \le R(a,b)$ can be non-zero values. From (3.29) (3.30) and (3.31), we see that $\mathbf{P}_k$ can be computed from $\mathbf{P}_{k-1}$ and observation $\vec{\boldsymbol{X}}_k$ at time k. Hence, we have the recursive update formula for the posterior probabilities $\{\mathbf{P}_k, p_{0,k}\}$. More importantly, by the relationship between $\{\mathbf{P}_k, p_{0,k}\}$ and Π_k,

$$\begin{cases} \Pi_k^{(i)} = \sum\limits_{a=1}^{H} \sum\limits_{b=1}^{W} \sum\limits_{1=1}^{R(a,b)} p_{i,k,a,b,r}, i \in \mathcal{I} \\ \Pi_k^{(0)} = p_{0,k} \end{cases} \tag{3.32}$$

we can update Π_k recursively. Afterwards, we can follow the same steps described in Section 3.3 and obtain the optimal SCD rule of the 2D lattice case. Similar to the linear sensor array case, since the state space increase exponentially with H, W and I, the extreme high complexity make the optimal method hard to implement.

3.5.3 Low-complexity rule

The low-complexity threshold given in (3.11) works for te 2D lattice sensor array case and the asymptotic optimality also preserves. The only difference between the threshold rules of the linear sensor case and the 2D lattice case is the proof of the convergence of the LLR process. Therefore, we only provide the proof the convergence of the LLR process for the 2D lattice case.

For $i \in \mathcal{I}$ and time $k > 0$, we define

$$H_k^{(i)} = \sum_{S_1=1}^{H} \sum_{S_2=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k} \left[\left(\prod_{n=1}^{n_0-1} \left(\frac{(1-\rho) f_0(x_{n,S_1,S_2})}{(1-\rho_1) f_i(x_{n,S_1,S_2})} \right) \right) \cdot \psi_1^{(i)}(k, n_0, S_1, S_2) \right]$$

where

$$\begin{aligned}\psi^{(i)}_{r+1}\left(k,n_r,S_1,S_2\right) = \prod_{n=1}^{k}\left[(1-\rho_1)\prod_{(a,b)\in\mathcal{O}(S_1,S_2,r)}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right] + \\ \rho_1\sum_{n_{r+1}=n_r}^{k}\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)}\prod_{n=1}^{n_{r+1}-1}\frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\psi^{(i)}_{r+2}\left(k,n_{r+1},S_1,S_2\right), R(S_1,S_2) > r \geq 0.\end{aligned} \tag{3.33}$$

In addition, $\psi^{(i)}_{R(S_1,S_2)+1}\left(k,n_{R(S_1,S_2)},S_1,S_2\right) = (1-\rho_1)^{n_{R(S_1,S_2)}-1}$. Therefore, we can express $\log\alpha^{(i)}_k$ as

$$\begin{cases}\log\alpha^{(i)}_k = \log[v_i\rho(1-\rho)] + \log\left(\prod_{a=1}^{H}\prod_{b=1}^{W}\prod_{m=1}^{k}f_i(x_{m,a,b})\right) + \log H^{(i)}_k, \text{for } i\in\mathcal{I} \\ \log\alpha^{(0)}_k = (k+1)\log(1-\rho) + \log\left(\prod_{a=1}^{H}\prod_{b=1}^{W}\prod_{m=1}^{k}f_0(x_{m,a,b})\right).\end{cases}$$

The next proposition describes the limit of $\log H^{(i)}_k/k$ as $k\to\infty$.

Proposition 3.4. For any $i,j\in\mathcal{I}$, if Condition 1 is satisfied,

$$\frac{1}{k}\log H^{(i)}_k \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} h(i,j) \tag{3.34}$$

where $h(i,j) = \left(\log(1-\rho) + L(q(j,i)-q(j,0))\right)_+$ and $L = HW$.

Proof. Please see Appendix B.3. □

After proving the convergence of the LLR process, the asymptotic optimality of the threshold rule (3.11) in the 2D sensor array case can be proved following the same steps introduced in Section 3.4.

3.6 Benefits of Increasing Number of Sensors

In this section we will prove that adding more sensors to the sensor array will always improve the performance of the multi-sensor threshold SCD rule when c_1 and c_2 are sufficiently small.

From the Bayesian cost of the optimal threshold rule in (3.17), we can see that if constants $w(i)$ and $w(i,0)$ increase, the cost will decrease. Although we know that $C^{(c_2)}(\delta_T^*) \to 0$ as $c_1, c_2 \to 0$, greater constants $w(i)$ and $w(i,0)$ can make $C^{(c_2)}(\delta_T^*)$ converge to 0 faster. Next, we will analyze how $w(i)$ and $w(i,0)$ change as more sensors are added to different sensor array structures.

3.6.1 Case 1: The first affected sensor is unknown

When Condition 1 is satisfied for $i, j \in \mathcal{I}$, and the first affected sensor is randomly chosen and unknown (as in Section 3.4 and 3.5). By (3.13) and Proposition 3.1, we have

$$w(i,j) = \begin{cases} Lq(i,j), \text{if } \log(1-\rho) \geq q(i,0) - q(i,j), i \in \mathcal{I} \\ Lq(i,0) - \log(1-\rho), j = 0 \text{ or } q(i,j) \leq q(i,0) \end{cases}.$$

By Assumption 2.1 and the fact $q(i,j)$ is the KL divergence, $q(i,j)$ is positive for $i, j \in \mathcal{I}$. Therefore, $w(i)$ and $w(i,j)$ will increase with the number of sensors. This implies that, with more sensors in the sensor array, the performance of the multi-sensor threshold SCD rule will be improved when Condition 1 is satisfied for all $i, j \in \mathcal{I}$ in the general case.

3.6.2 Case 2: The first affected sensor is known

As we introduced in Section 3.4.4, when Condition 1 does not hold and the first affected sensor is fixed and known, the calculation of constant w is more complicated. The reason is that adding one more sensor to the array may change the grouping result of Algorithm 1. Without of generality, we assume the sensor is added to the right of the first affected sensor s, i.e., we added the $l = (L+1)th$ sensor to the array. Then $\eta_L(i,j)$ change from $\log(1-\rho_2) + q(j,i) - q(j,0)$ to $q(j,i) - q(j,0)$. The new added $\eta_{L+1}(i,j) = \log(1-\rho_2) + q(j,i) - q(j,0)$. Based on the value of $\eta_L(i,j)$, the increment of $h(i,j)$ could be different. However, it's easy to check that, the increment of $h(i,j)$ is upper bounded by $(q(j,i) - q(j,0))_+$. Based on this observation and (3.13), we can see that by adding one sensor, $w(i, j)$ will always increase. Therefore, the performance of the multi-sensor threshold SCD rule can always be improved by adding sensors to the sensor array.

It is worth noting that the benefit introduced in this section is for the asymptotic case, i.e. $c_1, c_2 \to 0$. In other words, adding more sensors will improve the performance when c_1 and c_2 are sufficiently small. However, such property many not hold when c_1 and c_2 is relative large.

3.7 Numerical results

Since the optimal SCD rule is too complex to implement in the multi-sensor case, obtaining the optimal solution is extremely time-consuming, even for a simple case with $L = 2$ and $I = 2$. Therefore, we will not carry out experiments to directly compare the performance of the optimal SCD rule and the threshold SCD rule. However, we still can validate that the multi-sensor threshold SCD rule has a considerable improvement over a single sensor threshold rule (all sensors except the first one are ignored) and a mismatched threshold rule (changes of all sensors are falsely assumed to happen at the same time). Particularly, we will investigate the performance of the multi-sensor threshold SCD rule in a general case (first affected sensor is a random variable) and a special case (first affected sensor is fixed and known). In this section, we provide 4 numerical examples to illustrate the performance of the threshold SCD rule. In all following examples, the results are estimated by Monte-Carlo simulations. Concretely, we generate data samples following the underlying SCD process and apply the SCD rules to the generated sequence. An episode ends when the SCD rule makes the final detection and identification decisions. Then we calculate the Bayesian cost and start another episode. The Bayesian cost $C(\tau_1, \tau_2, d)$ is approximated using the average value of 10,000 episodes of Monte-Carlo simulation.

In the first example, the observed data samples are generated by a two-dimensional normal distribution, $\mathcal{N}(\vec{\boldsymbol{\mu}}, \boldsymbol{I}_2)$. The mean vector $\vec{\boldsymbol{\mu}}$ changes at the change point. In the first example, we consider the case with two possible post-change mean vectors $\vec{\boldsymbol{\mu}}_1 = (0, 1)$ and $\vec{\boldsymbol{\mu}}_2 = (0, -1)$ and the pre-change mean vector $\vec{\boldsymbol{\mu}}_0 = (0, 0)$. In addition, we set $\rho_1 = 0.2$, $\rho_2 = 0.2$, $\rho = 0.01$, $(v_1, v_2) = (0.3, 0.7)$ and $c_2/c_1 = 0.1$. All the penalty factors of the false alarm and misdiagnosis are set to be 1. For this problem formulation, we study 7 different cases: (1). $L = 5$ with $\vec{\boldsymbol{K}} = [0.2, 0.2, 0.2, 0.2, 0.2]$ (General case); (2). $L = 5$ with $\vec{\boldsymbol{K}} = [0, 0, 1, 0, 0]$ (Special case); (3) $L = 5$ with $\vec{\boldsymbol{K}} = [0, 0, 1, 0, 0]$ (Mismatch case); (4). $L = 2$ with $\vec{\boldsymbol{K}} = [0.5, 0.5]$ (General case); (5). $L = 2$ with $\vec{\boldsymbol{K}} = [0, 1]$ (Special case); (6) $L = 2$ with $\vec{\boldsymbol{K}} = [0, 1]$ (Mismatch case); (7) Single sensor case. The result of these 7 cases are shown in Fig. 3.2. In addition, Table I presents the performance of the two-stage SCD rule with different sensor arrays. In Table I, we have the following columns: FAP (false alarm probability), MISDP (misdiagnosis probability), delay1 (expected delay time in the detection stage), delay2 (expected delay time in the identification stage), wrong decision costs (FAP + MISDP), total delay cost (c_1∗delay1+c_2∗delay2), Bayesian cost (FAP+ MISDP+total delay cost). From these results, we can see the general trends of the performance of the threshold rule are: (1) Special case¿ General case¿ Mismatch case and single sensor case; (2) $L = 5$ ¿ $L = 2$ for the general and the special case. The advantage of the special case over the general case is due to the additional information that the first sensor affected by the change is known in the special case. In conclusion, the results of this example indicate that with more sensors and the correct information about the problem formulation, the proposed multi-sensor threshold SCD rule can efficiently improve the performance.

In the second example, we illustrate our results using pre-change and post-change distributions that are more complex than the one used in the first example. Firstly, we define a 2-D distribution, $F_L(\mu_1, \mu_2)$. With $F_L(\mu_1, \mu_2)$, the two elements in each data sample are independent and follow the Laplace distributions, $L(\mu_1, 1/\sqrt{2})$ and $L(\mu_2, 1/\sqrt{2})$, respectively. In this example, we implement three experiments: (1) Change in the mean vector of $F_L(\mu_1, \mu_2)$. The pre-change distribution is $F_L(0, 0)$, the post-change distributions are $F_L(0, 1)$ and $F_L(0, -1)$;

	c_1	FAP	MISDP	Delay1	Delay2	Wrong decision cost	Total delay cost	Bayesian cost
Single sensor	0.1	0.1334	0.0045	7.6437	18.729	0.1379	0.9517	1.08956
	0.05	0.058	0.0023	10.0065	10.0966	0.0603	0.5508	0.6111
	0.02	0.0215	0.0013	12.4123	6.4296	0.0228	0.2611	0.2839
	0.01	0.0099	0.0006	13.926	5.2957	0.0105	0.1446	0.1551
	0.005	0.006	0.0004	15.3682	4.9344	0.0064	0.07931	0.08571
General case (L=2)	0.1	0.0887	0.0044	7.2048	11.0701	0.0931	0.8312	0.9243
	0.05	0.0459	0.0023	8.4144	6.9719	0.0482	0.4556	0.5038
	0.02	0.0161	0.0004	9.6692	4.0748	0.0165	0.2015	0.218
	0.01	0.0097	0.0002	10.5496	3.2829	0.0099	0.1088	0.1187
	0.005	0.0043	0.0002	11.387	2.7081	0.0045	0.05829	0.06279
General case (L=5)	0.1	0.0286	0.0016	8.3372	4.5829	0.0302	0.8795	0.9097
	0.05	0.0134	0.0006	9.0294	2.7989	0.014	0.4655	0.4795
	0.02	0.0049	0.0003	9.7337	1.9988	0.0052	0.1987	0.2039
	0.01	0.0025	0.0003	10.3545	1.7942	0.0028	0.1053	0.10814
	0.005	0.0015	0.0002	10.771	1.5232	0.0017	0.05461	0.05632
Special case (L=2)	0.1	0.0885	0.0041	6.9261	11.1448	0.0926	0.8041	0.8967
	0.05	0.0386	0.0022	8.1569	6.2093	0.0408	0.4389	0.4797
	0.02	0.0166	0.0006	9.4394	3.9092	0.0172	0.1966	0.2138
	0.01	0.0083	0.0004	10.2133	3.1732	0.0087	0.1053	0.114
	0.005	0.0038	0.0002	11.0598	2.6632	0.004	0.05663	0.06063
Special case (L=5)	0.1	0.0265	0.0018	7.0316	4.2327	0.0283	0.7455	0.7738
	0.05	0.0127	0.0005	7.6022	2.8422	0.0132	0.3943	0.4075
	0.02	0.005	0.0003	8.3083	1.7477	0.0053	0.1697	0.175
	0.01	0.0038	0.0002	8.7582	1.6554	0.004	0.08924	0.09323
	0.005	0.0008	0.0001	9.1803	1.2985	0.0009	0.04655	0.04745
Mismatch case (L=2)	0.1	0.045	0.0017	8.8483	6.957	0.0467	0.9544	1.0011
	0.05	0.0226	0.0009	9.8357	4.5418	0.0235	0.5145	0.538
	0.02	0.009	0.0005	10.9599	3.3959	0.0095	0.226	0.2355
	0.01	0.0035	0.0001	11.7296	2.5338	0.0036	0.1198	0.1234
	0.005	0.0017	0.0001	12.5006	2.3469	0.0018	0.06368	0.06548
Mismatch case (L=5)	0.1	0.0115	0.0003	10.0698	2.7791	0.0118	1.03477	1.04657
	0.05	0.0064	0.0003	10.5553	1.9905	0.0067	0.5377	0.5444
	0.02	0.0021	0	16.3119	1.7122	0.0021	0.3297	0.3318
	0.01	0.001	0	11.5815	1.2742	0.001	0.1171	0.1181
	0.005	0.0005	0	12.0188	1.229	0.0005	0.0607	0.0612

Table 3.1: Performance of the multi-sensor threshold SCD rule in 7 different cases for the change on the mean of 2-D Gaussian distribution

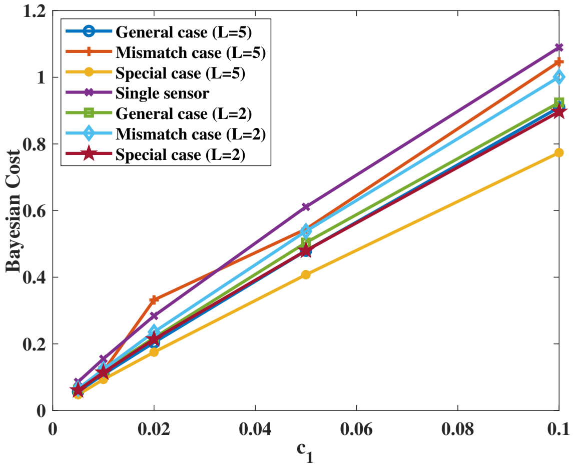
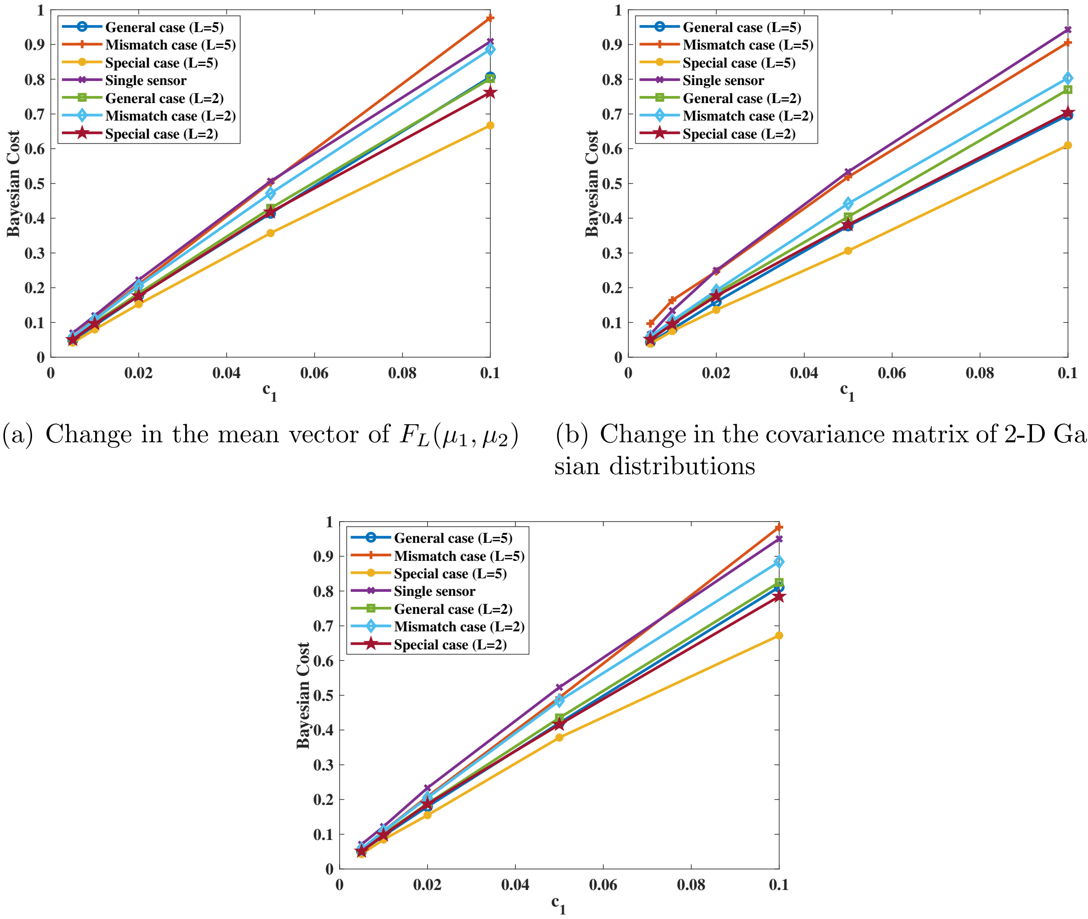


Figure 3.2: Performance of the multi-sensor threshold SCD rule in 7 different cases for the change on the mean of 2-D Gaussian distribution

(2) Change in the covariance matrix of 2-D Gaussian distribution. The pre-change distribution is 2-D Gaussian distribution, $\mathcal{N}(\vec{\mathbf{0}}, 0.5\boldsymbol{I}_2)$, the post-change distributions are $\mathcal{N}(\vec{\mathbf{0}}, \boldsymbol{I}_2)$ and $\mathcal{N}(\vec{\mathbf{0}}, 2\boldsymbol{I}_2)$; (3) Change in the type of the distribution. The pre-change distribution is a 2-D Gaussian distribution, $F_L(0,0)$, the post-change distributions are $\mathcal{N}((0,1), \boldsymbol{I}_2)$ and $\mathcal{N}((0,-1), \boldsymbol{I}_2)$. All the other parameters in this example are the same as the first example. The simulation results of the three settings are shown in Figure 3.3. These results are very similar to the results in the first example. It indicates that the proposed multi-sensor threshold SCD rule (general case and special case) works well for various settings of pre-change and post-change distributions.

In the first two examples, we know that the additional information about the first sensor affected by the change makes the special case has better performance than the general case. However, from the analysis in Section 3.4.4, the limit of the cost function of the two cases should be the same. In the third example, we implement an experiment to validate this analysis result. Assume $L = 5$, for the general case, we assume $\vec{\boldsymbol{K}} = [0.2, 0.2, 0.2, 0.2, 0.2]$. For the special case, we assume $\vec{\boldsymbol{K}} = [0, 0, 1, 0, 0]$. Following similar setting of the first example, we only change the mean vector to $\vec{\boldsymbol{\mu}}_1 = (0, 0.2)$ and $\vec{\boldsymbol{\mu}}_2 = (0, -0.2)$. It is easy to check that Condition 1 is satisfied for all $i, j \in \mathcal{I}$. The cost functions of the two cases and the



(a) Change in the mean vector of $F_L(\mu_1, \mu_2)$

(b) Change in the covariance matrix of 2-D Gaussian distributions

(c) Change in the type of 2-D distributions

Figure 3.3: Performance of the multi-sensor threshold SCD rule in 7 different cases for different types of change

ratio between them are given in Table 3.2. From that table, we can see that, with smaller c_1 (and smaller c_2 since c_2/c_1 is set to be 0.1), the ratio between the cost of the special case and the general case is getting closer to 1. From the experiments we did in the first three examples, we can see that the prior information about the first affected sensor can help to improve the performance of the multi-sensor threshold SCD rule, especially when c_1 and c_2 is not very small. However, this improvement will get smaller as c_1 and c_2 approach zero.

As we introduced in Section 3.4.3, the threshold SCD rule is asymptotically optimal when the Condition 1 is satisfied for all $i, j \in \mathcal{I}$. If the condition is not satisfied, currently we are not able to prove the asymptotic optimality of the threshold SCD rule for the general case. In the fourth example, we numerically study the performance of the multi-sensor SCD rule in the general case when Condition 1 is not satisfied. We still use the same 2-D Gaussian setting of the first example except for the mean vector. We set $\vec{\boldsymbol{\mu}}_1 = (0, 0.1)$ and $\vec{\boldsymbol{\mu}}_2 = (0, -0.1)$ in order to make the Condition 1 unsatisfied. In this setting, we compare the performance of the general case and the special case. The result is shown in Fig. 3.4. From this figure, we can see that the performance of the multi-sensor threshold SCD rule in the general case is very close to that in the special case. According to our analysis in Section 3.4, we know the multi-sensor threshold SCD rule is always asymptotically optimal in the special case. Therefore, we know that without the asymptotic optimal guarantee, the multi-sensor threshold SCD rule can still have good performance.

Finally, we provide a numerical experiment for the 2D sensor array described in Section 3.5. In this experiment, the propagation probability of the 2D lattice sensor array is $\rho_1 = 0.2$. The change can happen to any sensor in the array following a uniform distribution, i.e., $\mathcal{P}(S = (a, b)) = 1/(HW)$ for any $1 \leq a \leq H$ and $1 \leq b \leq W$. All other settings of this experiment are the same as the first experiment. The Bayesian costs of the multi-sensor threshold SCD rule with three different 2D lattice arrays are presented in Table. 4.2. The performance of the single sensor case is also given as a reference. From this table, we can see that the performance of the threshold SCD rule in the sensor array case is generally

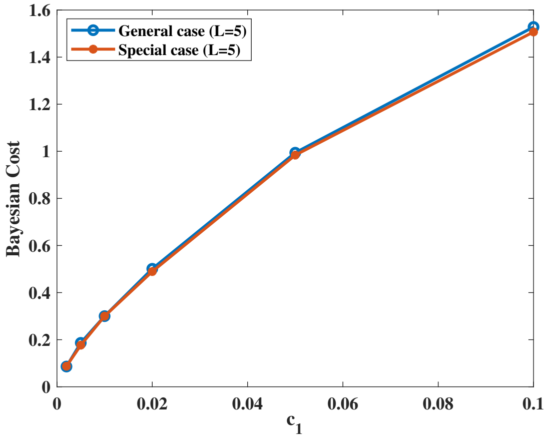


Figure 3.4: Performance of the multi-sensor threshold SCD rule in general case and special case when Condition 1 is not satisfied

Table 3.2: Performances of the two-stage multi-sensor threshold SCD rules with different c_1

c_1	General Case	Special Case	Bayesian Cost Ratio
10^{-2}	0.5291	0.4956	0.937
10^{-4}	1.03e-2	9.83e-3	0.955
10^{-6}	1.26e-4	1.23e-4	0.980
10^{-8}	1.69e-6	1.66e-6	0.988
10^{-10}	2.09e-8	2.08e-8	0.993

better than in the single sensor case. We also notice that the performance of a large sensor array can be worse than a smaller sensor array when the unit delay cost is relatively big. For example, the Bayesian costs of 10 × 10 and 5 × 5 sensor array are larger than that of the 2 × 2 sensor array when $c_1 = 0.1$. This result indicates that the Bayesian cost of the multi-sensor threshold SCD rule does not strictly decrease as the number of sensors increases when the unit delay cost is not very small. However, the results in Table. 4.2 also validate that, when the unit delay costs are sufficiently small, e.g. $c_1 = 1 \times 10^{-6}, 1 \times 10^{-8}$ or 1×10^{-10}, the performance of the multi-sensor threshold SCD rule with a large sensor array is always better than that with a smaller sensor array. This result is consistent with the conclusion we obtained in Section 3.6.

Table 3.3: Performances of the two-stage multi-sensor threshold SCD rules in 2D lattice sensor array case

	c_1=0.1	c_1=0.05	c_1=0.01	c_1=0.05	c_1=0.001	c_1=1e-6	c_1=1e-8	c_1=1e-10
Single Sensor	1.08956	0.6111	0.1551	0.08571	0.01958	3.2514e-05	4.1536e-07	5.08306e-09
2 × 2 Sensor Array	0.8477	0.4504	0.09892	0.05267	0.01121	1.46e-05	1.73129e-07	1.96511e-09
5 × 5 Sensor Array	0.8534	0.4606	0.09743	0.05018	0.01049	1.2134e-05	1.3309e-07	1.3984e-09
10 × 10 Sensor Array	0.9512	0.4803	0.1017	0.05129	0.01062	1.1824e-05	1.2765e-07	1.3422e-09

Chapter 4

Data Driven QCD problems

In this chapter, we will introduce the data-driven QCD problem. Firstly, we will introduce the formulation of the data-driven QCD problem under the i.i.d case and hidden Markov model (HMM) case. Secondly, we will introduce the structures of the optimal solutions for the Bayesian QCD problem in the i.i.d. case and HMM case. Then, we will introduce the deep Q-learning based QCD rule. Afterward, we will present the neural Monte Carlo based QCD rule for the i.i.d. case and HMM case, respectively. Finally, we will present experimental results to validate the performance of the proposed methods.

4.1 Problem formulation

In this section, we mainly study the QCD problem with two different observation models, i.e., the i.i.d. model and the Hidden Markov Model (HMM). As the observation model of the i.i.d. case is the same as that introduced in Section 2.1, here we only give the observation model for the HMM case. Afterward, we give the formulation of the data-driven Bayesian QCD problem.

4.1.1 HMM observation model

Let $\{Y_t, t \geq 0\}$ be a time-homogeneous Markov chain on a probability space $(\Omega_Y, \mathcal{F}_Y, \mathbb{P}_Y)$ with finite state space $\mathcal{Y} = \{1, 2, ..., I\}$, and transition matrix P in which $P(k, i) := \mathbb{P}(i|k)$ for $i, k \in \mathcal{Y}$. Suppose $\mathcal{Y}_1 = \{I_0 + 1, I_0 + 2, ..., I\}$ is a closed subset of $\mathcal{Y}$, where $I_0 < I$. The collection of remaining states $\mathcal{Y}_0 = \mathcal{Y} \backslash \mathcal{Y}_1 = \{1, 2, ..., I_0\}$ does not have any closed sets. In other words, in the transition matrix P, $P(y_t, y_{t+1}) = 0$ if $y_t \in \mathcal{Y}_1$ and $y_{t+1} \in \mathcal{Y}_0$. The change time $\lambda : \Omega \mapsto \{0, 1, \ldots\}$ is the first time the state $Y_t \in \mathcal{Y}_1$, i.e., λ is the time when the hidden states change from $\mathcal{Y}_0$ to $\mathcal{Y}_1$. In addition, the initial probability $P(Y_0 = i) = \eta_i$ where $\sum_{i \in \mathcal{Y}} \eta_i = 1$. Let $\overrightarrow{\eta} = (\eta_1, \eta_2, ..., \eta_I)$.

However, the sequence $\{Y_t, t \geq 0\}$ can not be directly observed. $\{X_t\}_{1 \leq t}$ is the directly observable process hosted by a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and the distribution of X_t depends on the hidden state Y_t. Let $f_y(\mathcal{X}), y \in \mathcal{Y}$ be the probability measures on a measurable space $(\mathcal{X}, \mathfrak{X})$, then

$$\begin{aligned}
&\mathbb{P}\left(Y_0 = y_0, Y_1 = y_1, \ldots, Y_t = y_t, X_1 = x_1, \cdots, X_t = x_t\right) \\
&= \overrightarrow{\eta}(y_0) \textstyle\prod_{n=1}^{t} P\left(y_{n-1}, y_n\right) f_{y_n}(x_n)
\end{aligned}$$

for $t \geq 1$, $y_0, y_1, ..., y_t \in \mathcal{Y}$.

4.1.2 Data-driven Bayesian quickest change detection problem

In Bayesian QCD problem, the change point, λ, follows a prior distribution P_λ. We assume P_λ is

$$\mathbb{P}\{\lambda = t\} = \begin{cases} \rho, & \text{if} \quad \mathrm{t = 0} \\ (1-\rho)^t \rho, & \text{if} \quad \mathrm{t \neq 0} \end{cases}.$$

For the HMM case, the transition matrix satisfies $\sum_{k \in \mathcal{Y}_1} P(i, k) = \rho$ for every $i \in \mathcal{Y}_0$ and the initial probabilities satisfy $\sum_{k \in \mathcal{Y}_1} \eta_k = \rho$.

Our goal is to detect the change point λ quickly and accurately, based on the observation sequence $\{X_t, t > 0\}$. Let τ be the time we raise an alarm. Then the false alarm happens if $\tau < \lambda$ and the delay is $(\lambda - \tau)_+$. Hence we define the expected cost of the change point detection problem as

$$C(\tau) = \mathbb{E}[\mathbf{1}_{\{\tau<\lambda\}} + c(\lambda - \tau)_+] \tag{4.1}$$

where $\mathbf{1}_{\{.\}}$ the indicator function and c is the unit cost of detection delay. Therefore, the best expected cost for the change point detection problem is

$$V_0 = \underset{\tau\in\mathcal{T}}{inf}\, \mathbb{E}[\mathbf{1}_{\{\tau<\lambda\}} + c(\lambda - \tau)_+] \tag{4.2}$$

where $\mathcal{T}$ is the space of stopping time $\tau : \Omega \to \{1, 2, \cdots, T\}$.

As will be discussed in Section 4.2, the optimal solution for online Bayesian QCD problem can be found when ρ, f_0 and f_1 or $\overrightarrow{\eta}$, P and $\mathbb{P}(x|y)$ are known. However, this knowledge is not always available in real-world problems. When the true underlying model is different from the model used to derive the optimal solution, the performance could be seriously affected. In practice, a common situation is that the only information we have is the historical data about the QCD process. In this section, we want to solve the online Bayesian QCD problem under the data-driven problem setting. Concretely, based on the historical dataset, our goal is to find a data-driven stopping rule which can achieve or get close to V_0.

4.2 The Optimal solution with Prior Knowledge of the QCD process

Before discussing the proposed data-driven solution, we introduce the optimal solution of the online Bayesian QCD problem. The optimal solution only works when prior knowledge of the QCD process is known. However, the structure of the optimal solution is important for the understanding of the proposed NMC-based QCD rule, which will be introduced in Sections 4.4 and 4.5 for the i.i.d. and HMM observation models respectively. Therefore, in this section, we provide a brief introduction of the optimal solution for Bayesian QCD problems for the i.i.d. case and the HMM case. For detailed proof of the optimal QCD rules in these two cases, please refer to [50, 91].

4.2.1 The i.i.d. case

For $t \geq 0$, let $\Pi_t = (\Pi_t^{(0)}, \Pi_t^{(1)}) \in \mathcal{Z}$ be the posterior probability process defined as $\Pi_t^{(1)} := \mathbb{P}\{\lambda \leq t | \mathcal{F}_t\}$ and $\Pi_t^{(0)} := \mathbb{P}\{\lambda > t | \mathcal{F}_t\}$ where $\mathcal{Z} \triangleq \{\Pi \in [0,1]^2 | \Pi^{(1)} + \Pi^{(0)} = 1\}$.

Following the derivation in [50], the expected cost in (4.1) can be expressed as

$$C(\tau) = \mathbb{E}\left[\sum\nolimits_{n=0}^{\tau-1} c\Pi_n^{(1)} + \Pi_\tau^{(0)}\right]. \tag{4.3}$$

Then we can define the cost-to-go function as a function of the posterior probability,

$$W(\Pi_t) = \min\left(\Pi_t^{(0)}, c\Pi_t^{(1)} + \mathbb{E}\left[W(\Pi_{t+1})|\mathcal{F}_t\right]\right). \tag{4.4}$$

The first item inside the minimization is the expected cost of raising an alarm immediately and the second item is the expected cost of observing more data samples. $W(\Pi_t)$ is the minimal expectation of the cost we still need to pay in the future based on the current state Π_t.

For the i.i.d case, when the pre-change distribution f_0, post-change distribution f_1, and the distribution of change ρ are known, we are able to update the posterior probability recursively following:

$$\Pi_t^{(0)} = \frac{(1-\rho)\Pi_{t-1}^{(0)} f_0(x_t)}{(1-\rho)\Pi_{t-1}^{(0)} f_0(x_t) + \left(\Pi_{t-1}^{(1)} + \Pi_{t-1}^{(0)}\rho\right) f_1(x_t)} \tag{4.5}$$

and $\Pi_t^{(1)} = 1 - \Pi_t^{(0)}$. The initial state, $\Pi_0 = (1-\rho, \rho)$. Based on this recursive updating rule,

$\mathbb{E}\left[W(\Pi_{t+1})|\mathcal{F}_t\right]$ can be calculated as

$$\mathbb{E}\left[W(\Pi_{t+1})|\mathcal{F}_t\right] = \int W(\Pi_{t+1}(\Pi_t, x)) \sum_{i\in\{0,1\}} f_i(x)\Pi_t^{(i)} dx.$$

Then, we can use DP to solve the Bellman equation (4.4) and obtain the cost-to-go $W(\Pi)$ for all $\Pi \in \mathscr{Z}$.

After solving (4.4) using dynamic programming, we have the cost-to-go function $W_t(\Pi_t)$ for $0 \le t \le T$. The optimal stopping rule is $\tau^* = \inf\left\{t|W_t(\Pi_t) = \Pi_t^{(0)}\right\}$. As discussed in [50], when P_λ is a Geometric distribution, then the optimal solution can be further simplified as

$$\tau_{opt} = \inf\{t \ge 0|\Pi_t^{(0)} \le \pi^*\} \tag{4.6}$$

where $\pi^* = \sup\{\pi \in [0,1]|\pi = W((\pi, 1-\pi))\}$. This rule indicates that we should raise an alarm once the expected cost of false alarm is smaller than the expected cost of observing more data samples.

4.2.2 The HMM case

For the HMM case, we can apply a similar solution as in the i.i.d. case. However, the posterior probability we used in the i.i.d. case, Π_t, can not be recursively updated. For this reason, we define the posterior probabilities $\tilde{\Pi}_t = (\tilde{\Pi}_t^{(1)}, \tilde{\Pi}_t^{(2)}, ..., \tilde{\Pi}_t^{(I)})_{t\ge0} \in \tilde{\mathscr{Z}}$, where $\tilde{\Pi}_t^{(i)} := \mathbb{P}\{y_t = i|\mathcal{F}_t\}$ for all $i \in \mathcal{Y}$ and $\tilde{\mathscr{Z}} = \{\tilde{\Pi} \in [0,1]^I | \sum_{i\in\mathcal{Y}} \tilde{\Pi}^{(i)} = 1\}$. With this definition, the posterior false alarm probability can be expressed as $\sum_{i\in\mathcal{Y}_0} \tilde{\Pi}^{(i)}$. Therefore, the expected cost in (4.3) can be expressed as

$$C(\tau) = E\left[c\sum\nolimits_{n=0}^{\tau-1}\sum\nolimits_{i\in\mathcal{Y}_1}\tilde{\Pi}_n^{(i)} + \sum\nolimits_{i\in\mathcal{Y}_0}\tilde{\Pi}_\tau^{(i)}\right]. \tag{4.7}$$

Then we can define the cost-to-go function for this DP problem as a function of the posterior probability,

$$W(\tilde{\Pi}_t) = \min \Bigg(\sum\nolimits_{i \in \mathcal{Y}_0} \tilde{\Pi}_t^{(i)}, \\ c \sum\nolimits_{i \in \mathcal{Y}_1} \tilde{\Pi}_t^{(i)} + \mathbb{E} \left[W(\tilde{\Pi}_{t+1}) | \mathcal{F}_t \right] \Bigg). \tag{4.8}$$

The first item inside the minimization is the expected cost of raising an alarm immediately and the second item is the expected cost of observing one more data sample. $W(\tilde{\Pi}_t)$ is the minimal expectation of the cost we still need to take in the future based on the current state Π_t.

For the HMM case, when the sample distributions $\{f_y\}_{y \in \mathcal{Y}}$, transition matrix P, and the distribution of change ρ are known, we are able to update the posterior probability $\tilde{\Pi}_t$ recursively. Concretely, $\tilde{\Pi}_t$ can be updated recursively as:

$$\tilde{\Pi}_t^{(i)} = \frac{\sum_{k \in \mathcal{Y}} \tilde{\Pi}_{t-1}^{(k)} P(k,i) f_i(x_t)}{\sum_{j \in \mathcal{Y}} \sum_{k \in \mathcal{Y}} \tilde{\Pi}_{t-1}^{(k)} P(k,j) f_j(x_t)}, \text{for } i \in \mathcal{Y}, \tag{4.9}$$

and $\tilde{\Pi}_0 = (\eta_1, \eta_2, ..., \eta_I)$. Based on this recursive updating rule, $\mathbb{E} \left[W(\tilde{\Pi}_{t+1}) | \mathcal{F}_t \right]$ can be calculated as

$$\mathbb{E} \left[W(\tilde{\Pi}_{t+1}) | \mathcal{F}_t \right] = \int W(\tilde{\Pi}_{t+1}(\tilde{\Pi}_t, x)) \sum\nolimits_{i \in \mathcal{Y}} f_i(x) \tilde{\Pi}_t^{(i)} dx.$$

Then, we can use DP method to solve the Bellman equation (4.8) and calculate the cost to go $W(\tilde{\Pi})$ for all $\tilde{\Pi} \in \tilde{\mathcal{Z}}$. The optimal stopping rule is

$$\tau^* = \inf \left\{ t | W_t(\tilde{\Pi}_t) = \sum\nolimits_{i \in \mathcal{Y}_0} \tilde{\Pi}_t^{(i)} \right\}. \tag{4.10}$$

This rule indicates that we should raise an alarm once the expected cost of false alarm is smaller than the expected cost of observing more data samples.

4.3 A Deep Q-learning based QCD rule for the i.i.d case

The online Bayesian QCD process can be formulated as a partially observable Markov decision process (POMDP). A POMDP is typically defined by a tuple $\{S, A, T_r, R, \Omega, O, \gamma\}$. S is the state space. A is the action space. T_r is the set of conditional transition probabilities between states. $R : S \times A \to \mathbb{R}$ is the reward function. Ω is the space of observations. O is a set of conditional observation probabilities. Finally, $\gamma \in [0, 1]$ is the discount factor. In the POMDP, the state is unknown to the agent. However, the agent can make decision based on the observations. At each time t, a new observation $o_t \in \Omega$ is obtained. The distribution of o_t is determined by the hidden state s_t. Based on the observations, the agent can take an action $a_t \in A$. Then the environment generates a reward $r_t = R(s_t, a_t)$ and a new state s_{t+1} following the conditional transition distribution $T_r(s_{t+1}|s_t, a_t)$. This procedure will be repeated until the terminal state is reached. The goal is to design a policy that maximizes the expected reward $\mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]$.

Now, we formulate the change point detection process as a POMDP. The state space of the QCD process $S := \{0, 1, E\}$ includes three states, i.e., pre-change state (0), post-change state (1) and terminal state (E). In the QCD problem, the state is hidden. However, we can make decision based on the observations. At each time t, a new data sample $X_t \in \Omega$ can be collected. If $s_t = 0$, then X_t follows the pre-change statistic process. If $s_t = 1$, then X_t follows the post-change statistic process. Based on the observations, the agent can take an action $a_t \in A = \{1, 0\}$. Here $a_t = 0$ means keep observing more data and $a_t = 1$ means raising an change alarm. Then the environment generates a reward $r_t = R(s_t, a_t)$ and

$$R(a_t, s_t) = \begin{cases} -c, & \text{If } \mathrm{a_t} = 0 \text{ and } \mathrm{s_t} = 1 \\ -1, & \text{If } \mathrm{a_t} = 1 \text{ and } \mathrm{s_t} = 0 \\ 0, & \text{otherwise} \end{cases} . \tag{4.11}$$

The new state in QCD process is determined as:

$$s_t = \begin{cases} 0, \ \text{If } s_{t-1} \neq E, \ a_{t-1} \neq 1 \ \text{ and } t < \tau \\ 1, \ \text{If } s_{t-1} \neq E, \ a_{t-1} \neq 1 \ \text{ and } t \geq \tau \\ E, \ \text{If } a_{t-1} = 1 \text{ or } s_{t-1} = E \text{ or } t > T \end{cases} . \tag{4.12}$$

By (4.1), we can see that the goal of QCD problem is to find a policy that minimizes $\mathbb{E}[\sum_{t=0}^{T} r_t]$.

When a historical dataset with finite samples is the only given resource, it is hard to extract the information required by the optimal solution. In this case, the QCD process is a black box to the decision-maker because only the input sequences and the true change-points are known. Model-free reinforcement learning is a good method for this situation. Q-learning is a classical and widely-used model-free reinforcement learning algorithm. Q-learning agents make decisions based on the learned Q-function. The optimal Q-function of online Bayesian QCD setting can be expressed as

$$Q^*(\mathbf{X}_t, a_t) = \begin{cases} \mathbb{E}[-\mathbf{1}_{\{t<\lambda\}} - c(\lambda - t)_+ | \mathbf{X}_t], \text{if } \mathrm{a_t} = 1 \\ \mathbb{E}[-c(t+1-\lambda)_+ + \max\left(Q^*(\mathbf{X}_{t+1}, 0), Q^*(\mathbf{X}_{t+1}, 1)\right) | \mathbf{X}_t], \text{if } \mathrm{a_t} = 0 \end{cases} , \tag{4.13}$$

where $\mathbf{X}_t = (X_1, \cdots, X_t)$ is the data sample available at time t and a_t is the action taken at time t. Corresponding to this optimal Q-function, the optimal decision is $a^* = \arg\max_{a \in \{0,1\}} Q^*(\mathbf{X}_t, a)$. This rule is equivalent to optimal rule described in Section 4.2. In [87], the authors discretized the observation space in order to apply tabular Q-learning. In this paper, considering the dimension of the observation space could be much larger than that in [87], we apply the DQN model [92].

DQN uses a neural network to approximate the Q function. Based on the observed data and the approximated Q function, the agent can take the action corresponding to the larger Q value. Since the dimension of the input layer of neural network is fixed, we can not input all observed samples $\mathbf{X}_t = (X_1, \cdots, X_t)$ to the neural network. Therefore, we set the input of the neural network as the observations in a sliding window with width w, i.e. $\mathbf{X}_{w,t} = (X_{t-w+1}, \cdots, X_t)$. In addition, the Q-function could be different at different time steps. Therefore, the input vector of the neural network is $\mathbf{I}_t = (X_{t-w+1}, \cdots, X_t, t)$. By inputting $\mathbf{I}_t$ into the neural network, the action value of $a = 0$ and $a = 1$, $Q(\mathbf{I}_t, 0, \theta)$ and $Q(\mathbf{I}_t, 1, \theta)$ are acquired. Here θ denotes the parameters of the neural network. With these two action values, the agent will take the action that has a larger action value. In other words, the DQN-based decision rule is $\tau = \inf\{t : 0 \leq t \leq T \text{ and } Q(\mathbf{I}_t, 1, \theta) > Q(\mathbf{I}_t, 0, \theta)\}$.

The dataset includes N episodes of the change process. For the ith episode, the data includes a sequence $\mathbf{X}_i = (X_{i,1}, \cdots, X_{i,T})$ and the true change point τ_i. To make the sequential data fit the sliding window, we need to preprocess the sequences in the dataset. For the first $w - 1$ samples in each sequence, they do not have enough previous samples to make a w long input sequence. To handle this issue, we add a $w - 1$ long prefix to every sequence in the dataset. Firstly, we collect all pre-change data samples in the training set and obtain a pool of pre-change samples. Then for each episode, $w - 1$ samples are randomly picked from the pool of pre-change samples and added in front of the data sequence. As a result, for the ith episode, the data sequence becomes $\mathbf{X}_i = (X_{i,2-w}, \cdots, X_{i,0}, X_{i,1}, \cdots, X_{i,T})$.

The training process of the DQN is described in Algorithm 1. We apply the techniques such as experience replay and target Q-network proposed by [92]. Experience replay randomly applies historical data to the current update step in order to reduce the variance of updates and achieves greater data efficiency. The target Q-network Q_2 is a copy of the main Q-network Q_1 but with older parameters. After every C steps, we update Q_2 once. Q_2 is used to calculate the expected continuing reward and can make the training process more stable. Although the approximated Q function may not be very close to the optimal Q function[93, 94], this error in action value may not significantly affect the performance of the DQN. As long as the order of two Q values is correct, the performance will be good. The experiments in the next section validate that the DQN-based QCD rule has a good performance. It is also worth noting that, although the prior distribution of the change

Algorithm 2: Training of the DQN

```
Initialize replay memory $\mathcal{D}$ to capacity $N_D$;
Initialize a DQN $Q_1$ with random weights $\theta_1$;
Initialize a target DQN $Q_2$ with the weights $\theta_2 = \theta_1$;
for $e$=1,...,$N$ do
    Initialize the input $\mathbf{I}_i = (X_{i,2-w}, \cdots, X_{i,1}, 1)$, and change point $\tau_e$;
    for $t$=1,...,$T$ do
        With probability $\epsilon$ select a random action $a_t$, otherwise select
          $a_t = \arg\max_{a\in\{0,1\}} Q_1(\mathbf{I}_t, a, \theta_1)$;
        if $t < \tau_e$ and $a_t = 1$ then
            $r_t = 1$;
        else if $t \geq \tau_e$ and $a_t = 0$ then
            $r_t = c$;
        else
            $r_t = 0$ ;
        end
        $\mathbf{I}_{t+1} = (X_{i,2-w+t}, \cdots, X_{i,1+t}, t+1)$;
        Store $(\mathbf{I}_t, r_t, a_t, \mathbf{I}_{t+1})$ in $\mathcal{D}$;
        Sample random minibatch of transitions $(\mathbf{I}_j, r_j, a_j, \mathbf{I}_{j+1})$ from $\mathcal{D}$;
        Set $y_j = \begin{cases} r_j, \text{if j = T} \\ r_j + \gamma \max_{a\in\{0,1\}} Q_2\left(I_{j+1}, a_j, \theta_2\right), \text{otherwise} \end{cases}$; Perform gradient descent
          on $(y_j - Q_1(\mathbf{I}_j, a_j, \theta_1))^2$ with respect to $\theta_1$. Every $C$ steps reset $\theta_2 = \theta_1$.
    end
end
```

point is assumed to be geometric in our problem setting, the DQN based approach also works for other types of prior distributions of the change point since it takes time steps into consideration.

4.4 A Neural Monte Carlo based QCD rule for the i.i.d case

As described in Section 4.2, a key step in the optimal QCD rules is to update posterior probabilities (Π_t or $\tilde{\Pi}_t$) recursively. This updating step can only be implemented when the a-priori information such as P, $\overrightarrow{\eta}$, ρ, f_0, f_1 and f_y are known. However, in many applications, it is common that a historical dataset with finite data samples is the only given resource. Concretely, only the observation sequences and the true change points in the dataset are known. We even do not know if the data samples are i.i.d., following an HMM, or some other non-i.i.d. process. In this case, it is hard to accurately extract these a-priori information from the data set. Therefore, we need a data-driven method that can help us estimate the posterior probabilities from the data. In this section, we will propose the Neural Monte Carlo (NMC) based solution for the i.i.d. case. In the next section, we will explain why this method also works for the HMM case and other non-i.i.d. cases.

From (4.6), we know that the posterior probability $\Pi_t^{(0)}$ and the threshold π^* are key parts of the optimal QCD rule for the i.i.d. case. In the data-driven setting, if we can approximate the posterior probability with $\hat{\Pi}_t^{(0)}$ for any time t, then we can select the optimal threshold $\hat{\pi}^*$ using line search and finally have a data-driven QCD rule similar to (4.6). To this end, we propose a Neural Monte Carlo (NMC) based solution for the data-driven QCD problem. The steps of the NMC-based QCD rule is given in Algorithm 1. Next, we will introduce these steps of this NMC-based QCD method.

4.4.1 A Neural Monte Carlo approximation model

If the cost of false alarm $c_F = 1$, $\Pi_t^{(0)}$ can be seen as the expected cost of raising an alarm given all data samples collected by t. In other words, the value function corresponding to the observations $\{x_1, x_2, ...x_t\}$ is $\Pi_t^{(0)}$. Therefore, the problem of estimating $\Pi_t^{(0)}$ can be seen as a value function approximation problem.

In the data set, we have the data sequences and the true change points of these sequences. With this data set, we can use the Monte Carlo method to approximate $\Pi_t^{(0)}$. Because continuous observation data samples are common in the QCD problem, we have a continuous input space. Therefore, we approximate $\Pi_t^{(0)}$ using a randomized neural network. In a randomized neural network, only the weights between the hidden layer and the output layer are trained while all other weights are frozen after being initialized. In particular, the last layer of the randomized neural network is a linear layer. Therefore, training the randomized neural network becomes a convex problem and has a convergence guarantee. More importantly, as proved in [95, 96], a randomized neural network can accurately approximate any continuous functions with a sufficiently wide hidden layer. Therefore, we apply a simple shallow neural network with one hidden layer. If we need a more powerful model for specific applications, a deep extension to this neural network is immediate by adding more non-linear layers to the model.

4.4.2 Data Preprocessing

Since the dimension of the input layer of the randomized neural network is fixed and the size of all observed samples $(X_1, \cdots, X_t)$ changes with different time t, we can not input $(X_1, \cdots, X_t)$ to the neural network. Therefore, we set the input of the neural network as the observations in a sliding window with width w, i.e., $\mathbf{X}_t = (X_{t-w+1}, \cdots, X_t)$. From equation (4.5), we can see that recently collected data samples are usually more important than earlier data samples in the calculation of $\Pi_t^{(0)}$ for the i.i.d case. In other words, with an appropriate value of w, the data samples in the sliding window $\mathbf{X}_t$ are sufficient to make a good estimation of $\Pi_t^{(0)}$. Typically, we select a large w if earlier data samples are important in the calculation of posterior probability. Otherwise, we can use a relatively narrow sliding window. For every time t and the corresponding input $\mathbf{X}_t$, the reward of raising alarm at t, R_t, is 1 if $t < \lambda$. On the other hand, if $t \geq \lambda$, $R_t = 0$.

The dataset includes N episodes of the change process. For the ith episode, the data includes a sequence of T samples, $\mathbf{S}_i = (X_{i,1}, \cdots, X_{i,T})$ and the true change point λ_i. To make the sequential data fit the sliding window, we need to preprocess the sequences in the dataset. For the first $w-1$ samples in each sequence, they do not have enough previous samples to make a w long input sequence. To handle this issue, we add a $w-1$ long prefix to every sequence in the dataset. Firstly, we collect all pre-change data samples in the training set and obtain a pool of pre-change samples. Then for each episode, $w-1$ samples are randomly picked from the pool of pre-change samples and added in front of the data sequence. As a result, for the ith episode, the data sequence becomes $\tilde{\mathbf{S}}_i = (X_{i,2-w}, \cdots, X_{i,0}, X_{i,1}, \cdots, X_{i,T})$. For each data sequence $\tilde{\mathbf{S}}_i$, we can generate T input data samples $\{\mathbf{X}_{i,1}, \mathbf{X}_{i,2}..., \mathbf{X}_{i,T}\}$ where $\mathbf{X}_{i,t} = (X_{i,t-w+1}, \cdots, X_{i,t})$. The corresponding reward samples are $\{R_{i,1}, R_{i,2}..., R_{i,T}\}$ where $R_{i,t} = 0$ if $t > \lambda_i$ and $R_{i,t} = 1$ if $t \leq \lambda_i$. By combining M episodes of data samples, $\{\tilde{\mathbf{S}}_i\}_{1 \leq i \leq M}$, we have the training data set $\{\mathbf{X}_{NMC}, \overrightarrow{R}_{NMC}\}$. The rest $N - M$ episodes, $\{\tilde{\mathbf{S}}_i\}_{M+1 \leq i \leq N}$, are used for building the validation set and test set. Typically, we set M as 70% of N.

Following the steps stated above, we have built a training dataset. However, in some situations, the training set could be imbalanced. A common imbalanced situation is that, when the length of the sequence T is very large, then there will be many more post-change data samples than pre-change data samples in the training set. With such an unbalanced training set, the accuracy of the posterior probability approximation model and the performance of the data-driven QCD rule can be compromised. In this case, we can use rebalance techniques, such as data selection or re-sampling, to process the training set $\{\mathbf{X}_{NMC}, \overrightarrow{R}_{NMC}\}$ if it is unbalanced. Concretely, we can rebalance the training data set by discarding the data samples after a threshold time $\tilde{T} < T$. Since most of the post-change data samples are at the later part of the sequence, dropping the later data samples can reduce the fraction of post-change data and make the training set balanced. In this method, the threshold time $\tilde{T}$ is treated as a hyper-parameter that can be tuned using the validation data set.

4.4.3 Training Process of the Neural Monte Carlo model

The randomized neural network is trained using Monte Carlo methods, as shown in Algorithm 1. Let $\boldsymbol{\theta}_0 \in \mathbb{R}^{(w+1)\times d}$ be the weights of the hidden layer, where d is the number of nodes in the hidden layer. Then the output of the hidden layer is $\mathbf{O}_{0,k} = \sigma(\boldsymbol{\theta}_0^T \mathbf{I}_{0,k})$, where $\mathbf{I}_{0,k} = [\mathbf{X}_k, 1]$ and $\mathbf{X}_k$ is an input data sample in the training set. Here σ is the activation function of the hidden layer. Elements in $\boldsymbol{\theta}_0$ are typically initialized by the standard normal distribution and will not be changed in the training process. Let $\boldsymbol{\theta}_1 \in \mathbb{R}^{(d+1)}$ be the weights of the output layer. Then the output of the neural network is $\boldsymbol{\theta}_1^T \mathbf{I}_{1,k}$ where $\mathbf{I}_{1,k} = [\mathbf{O}_{0,k}, 1]$. $\boldsymbol{\theta}_1$ is the weights we want to optimize in the training process. Since $\boldsymbol{\theta}_0$ is fixed, we can get a hidden output data set $\mathbf{I}_{1,NMC}$ from $\mathbf{X}_{NMC}$. In this case, this training problem becomes a linear value-function approximation problem. In [88], the gradient Monte Carlo algorithm is introduced to solve this linear value-function approximation problem and the updating rule of $\boldsymbol{\theta}_{1,t}$ is

$$\boldsymbol{\theta}_{1,t+1} = \boldsymbol{\theta}_{1,t} + \alpha(R_t - \boldsymbol{\theta}_{1,t}^T \mathbf{I}_{1,t})\mathbf{I}_{1,t} \tag{4.14}$$

for every step t, where α is the step size. Since the approximation model we apply is linear, the convergence of this training process is guaranteed.

From the updating rule (4.14), we can see that

$$\mathbb{E}[\boldsymbol{\theta}_{1,t+1}|\boldsymbol{\theta}_{1,t}] = \boldsymbol{\theta}_{1,t} + \alpha(\mathbb{E}[R_t \mathbf{I}_{1,t}] - \mathbb{E}[\mathbf{I}_{1,t}\mathbf{I}_{1,t}^T]\boldsymbol{\theta}_{1,t}).$$

Therefore, this algorithm will converge to $\boldsymbol{\theta}_{1,GMC}$ at which

$$\mathbb{E}[R_t \mathbf{I}_{1,t}] - \mathbb{E}[\mathbf{I}_{1,t}\mathbf{I}_{1,t}^T]\boldsymbol{\theta}_{1,GMC} = \mathbf{0}.$$

Since the data set $\{\mathbf{I}_{1,NMC}, \overrightarrow{R}_{NMC}\}$ is given in the data-driven QCD problem, we can estimate $\mathbb{E}[R_k \mathbf{I}_{1,k}]$ and $\mathbb{E}[\mathbf{I}_{1,k}\mathbf{I}_{1,k}^T]$ with the sample mean. Here $\mathbf{I}_{1,NMC} \in \mathbb{R}^{(d+1)\times MT}$, $\overrightarrow{R}_{NMC} \in \mathbb{R}^{MT}$. If there is enough data such that $\mathbb{E}[R_k \mathbf{I}_{1,k}]$ and $\mathbb{E}[\mathbf{I}_{1,k}\mathbf{I}_{1,k}^T]$ can be well estimated by their sample mean of the data set, the weights can be directly calculated as

$$\boldsymbol{\theta}_1 = (\mathbf{I}_{1,NMC}\mathbf{I}_{1,NMC}^T)^{-1}(\mathbf{I}_{1,NMC}\overrightarrow{R}_{NMC}). \tag{4.15}$$

Considering the case of $\mathbf{I}_{1,NMC}\mathbf{I}_{1,NMC}^T$ is not invertible or the matrix inversion requires computational complexity, the direct calculation (4.15) can only be used if $\mathbf{I}_{1,NMC}\mathbf{I}_{1,NMC}^T$ is invertible, and the number of hidden nodes is relatively small such that a $(d+1) \times (d+1)$ matrix can be inverted with a reasonable running time and memory. Therefore, as shown in Algorithm 1, if $\mathbf{I}_{1,NMC}\mathbf{I}_{1,NMC}^T$ is invertible and we have enough computational resource to invert a $(d+1)\times(d+1)$ matrix, we can directly calculate $\boldsymbol{\theta}_1$. Otherwise, we can update $\boldsymbol{\theta}_1$ following (4.14) iteratively.

Although the neural network is designed to approximate the posterior probability, the output may not necessarily fall in $[0,1]$ since the output layer is a linear layer. Therefore, after getting the output of the neural network, we clip the output to make sure it is a value in $[0,1]$ as

$$\hat{\Pi}^{(0)}(\boldsymbol{\theta}_1, \mathbf{X}_k) = \begin{cases} 1, & \text{if} \quad \boldsymbol{\theta}_1^T\mathbf{I}_{1,k} \geq 1 \\ 0, & \text{if} \quad \boldsymbol{\theta}_1^T\mathbf{I}_{1,k} \leq 0 \\ \boldsymbol{\theta}_1^T\mathbf{I}_{1,k}, & \quad \text{Otherwise} \end{cases} .$$

4.4.4 Threshold Selection

After training the neural network, we can estimate posterior probability $\hat{\Pi}_t^{(0)}$ for any time t. Next, we need to determine the threshold $\hat{\pi}_*$ for our NMC-based QCD rule. First, we apply the well-trained neural network to approximate the posterior probability $\hat{\Pi}_t^{(0)}$ for every sequence in the validation set. Second, we run Monte Carlo experiments on sequences of in the validation set and record the Bayesian costs for all candidate threshold $\pi \in \{1/K, 2/K, ..., (K-1)/K, 1\}$, using the threshold rule $\tau = \inf\{t \geq 0|\hat{\Pi}_t^{(0)} \leq \pi\}$. Here, K is the number of candidates in the line search method. Finally, based on the Bayesian cost records, we set $\hat{\pi}^*$ as the candidate π which is corresponding to the lowest Bayesian cost. Finally, we have the NMC-based QCD rule as

$$\tau = \inf\{t \geq 0|\hat{\Pi}_t^{(0)} \leq \hat{\pi}^*\}. \tag{4.16}$$

Algorithm 3: NMC-based SCD rule

```
Data preprocessing following Section 4.4.2. Get data set $\{\mathbf{X}_{NMC}, \overrightarrow{R}_{NMC}\}$;
Initialize a random neural network with hyper parameter $d$, $w$, $\tilde{T}$ and weights
  following standard gaussian distribution;
Get data set $\mathbf{I}_{1,NMC}$;
if a $(d+1) \times (d+1)$ matrix can be inverted with reasonable computational resource then
    $\boldsymbol{\theta}_1 = (\mathbf{I}_{1,NMC}\mathbf{I}_{1,NMC}^T)^{-1}(\mathbf{I}_{1,NMC}\overrightarrow{R}_{NMC})$;
else
    for $\mathbf{I}_k \in \mathbf{I}_{1,NMC}$ do
        $\boldsymbol{\theta}_{1,t+1} = \boldsymbol{\theta}_{1,t} + \alpha(R_t - \boldsymbol{\theta}_{1,t}^T\mathbf{I}_{1,t})\mathbf{I}_{1,t}$
    end
end
Select the threshold $\hat{\pi}^*$ by line search method.
Fine-tune the hyper-parameters $w$, $d$ and $\tilde{T}$.
Finally, the NMC-based QCD rule is $\tau = \inf\{t \geq 0 | \hat{\Pi}_t^{(0)} \leq \hat{\pi}^*\}$.
```

4.5 A Neural Monte Carlo based QCD rule for the HMM case

In Section 4.4, by replacing the two key parts in the optimal QCD rule, $\{\Pi_t^{(0)}, \pi^*\}$, with corresponding approximations $\{\hat{\Pi}_t^{(0)}, \hat{\pi}^*\}$, we proposed a NMC-based rule (4.16) for the data-driven Bayesian QCD problem in the i.i.d. case. It's natural to consider if we can do the same thing to the optimal rule in the HMM case. Unfortunately, it is challenging to extract sufficient information for the optimal HMM QCD rule from the observations. Concretely, it is hard to estimate the hidden process from observations, e.g., the number of hidden states, which state the post-change state is and which state the pre-change state is, etc.. Without these information, even the number of elements $\tilde{\Pi}_t$ should include is unknown. Hence it is challenging to directly extend the optimal SCD rule for the HMM to a data-driven version.

However, although we cannot estimate $\tilde{\Pi}_t$, we can still estimate the posterior probability $\Pi_t^{(0)} = \sum_{i \in \mathcal{Y}_0} \tilde{\Pi}_t^{(i)}$, similar to in the i.i.d. case. As the sufficient statistics $\tilde{\Pi}_t$ is unavailable, the posterior probability $\Pi_t^{(0)}$ becomes a reasonable alternative indicator that can help to

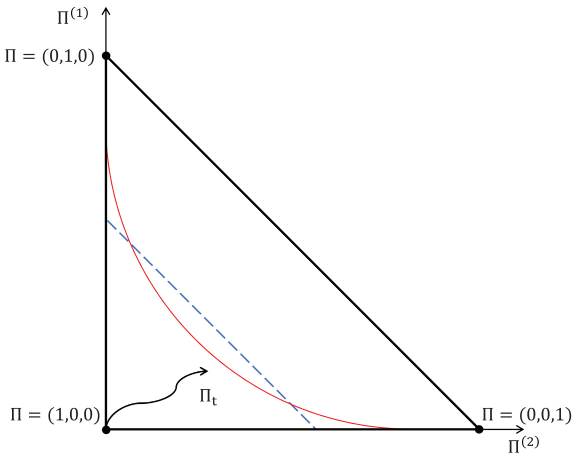


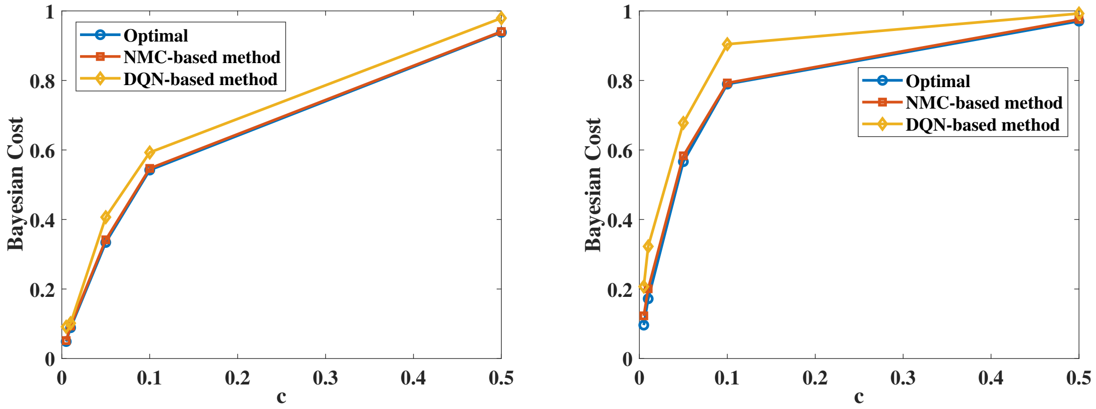
Figure 4.1: QCD boundaries of a simple QCD example in HMM case

detect the change. Concretely, we can apply the threshold rule as

$$\tau = \inf\left\{t \geq 0 | \Pi_t^{(0)} = \sum\nolimits_{i \in \mathcal{Y}_0} \tilde{\Pi}_t^{(i)} \leq \pi\right\}. \tag{4.17}$$

To illustrate this threshold rule, a simple example is given in Fig. 4.1. Assume we have a HMM QCD problem in which $\mathcal{Y} = \{1, 2, 3\}$, $\mathcal{Y}_0 = \{1\}$ and $\mathcal{Y}_1 = \{2, 3\}$. In the posterior probability space $\tilde{\mathcal{Z}} = \{\tilde{\Pi} \in [0, 1]^3 | \sum_{i \in \mathcal{Y}} \tilde{\Pi}^{(i)} = 1\}$, the optimal decision boundary given by (4.10) is the red curve. The straight blue line represents the decision boundary of the QCD rule (4.17). In general HMM QCD problems, the decision boundary of the QCD rule (4.17) is a plane in the space $\tilde{\mathcal{Z}}$ while the decision boundary of the optimal rule (4.10) is a surface in $\tilde{\mathcal{Z}}$. Since the information about the HMM is incomplete, we use a plane as the alternative of the surface. In the data-driven QCD problem, we apply the NMC-based QCD rule introduced in Section 4.4 to approximate the plane of the QCD rule (4.17). Finally, we obtain the same NMC-based QCD rule as (4.16).

Due to the difference between the HMM case and the i.i.d. case, we need to make some adjustments when we apply the NMC-based SCD rule to HMM QCD problems. In HMM case, the data samples in one sequence are not independent. Therefore, the elements in the prefix of the data sequence should not be independently selected as in the i.i.d. case. In



(a) $p_0 = 0.2$, $p_1 = 0.8$, $\rho = 0.01$

(b) $p_0 = 0.3$, $p_1 = 0.7$, $\rho = 0.01$

Figure 4.2: The Bayesian costs of the three QCD methods in the i.i.d. Bernoulli experiment

order to apply the NMC-based QCD rule to the HMM case, one change is needed for the data preprocessing step. In HMM case, we collect all pre-change subsequences from the data sequences $\{\mathbf{S}_i\}_{1\leq i\leq N}$ as the pool of pre-change subsequences. Note that, we only collect pre-change subsequence longer than $w-1$ samples. After that, for every sequence $\{\mathbf{S}_i\}$, $w-1$ continuous data samples are randomly selected from the pool of pre-change subsequences and added to $\mathbf{S}_i$ as the prefix. Afterwards, following the same steps as discussed in Section 4.4, we have the NMC-based QCD rule for the HMM case, which has the same expression as (4.16).

In the general non-i.i.d case, the posterior analysis and the detection boundary could be even more complicated than in the HMM case. That means getting the optimal solution for a QCD problem in general non-i.i.d. case becomes even harder. However, the posterior false alarm probability is still a reasonable indicator for the non-i.i.d. QCD problem and can be learned by following similar steps as those in the i.i.d. case and the HMM case. Therefore, the NMC-based rule could still be used for the QCD problem in different non-i.i.d. settings. This will be validated in the following section by simulation.

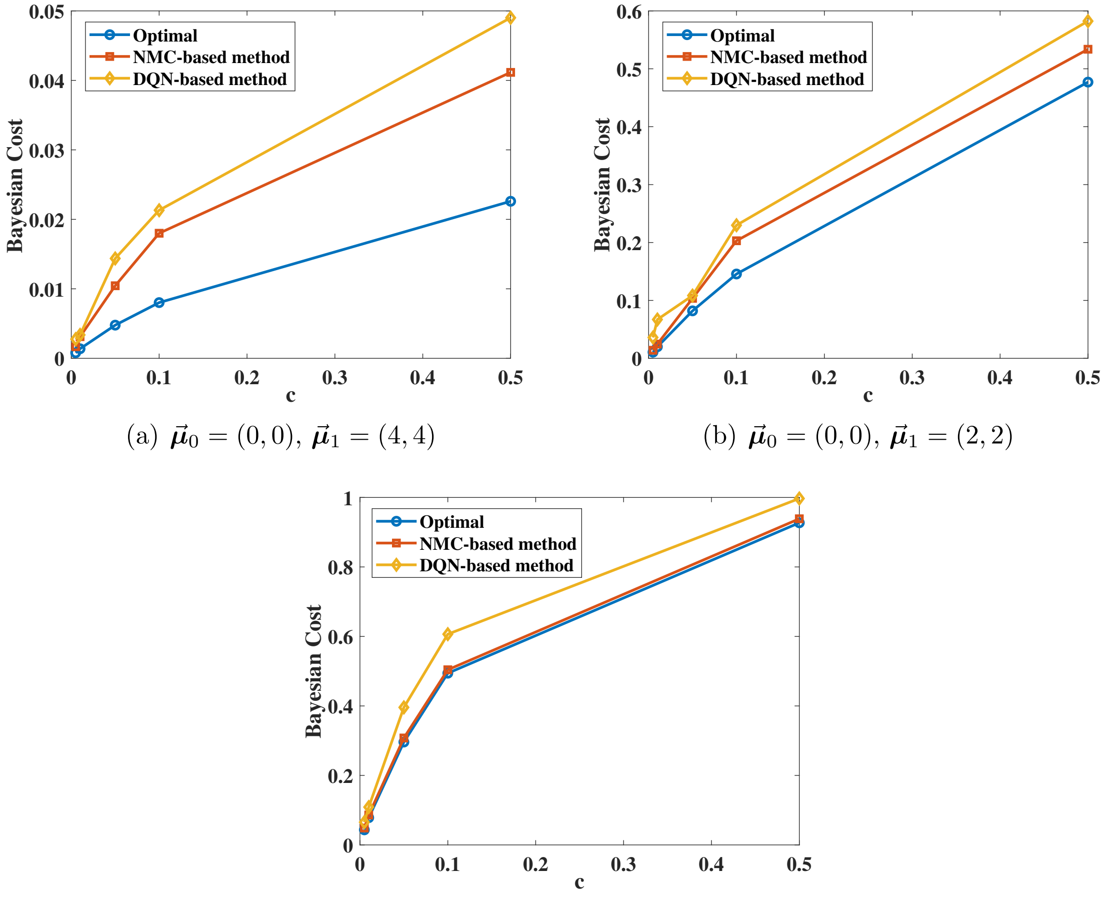
Table 4.1: Performances of the three QCD rules in the i.i.d. Bernoulli experiment

		$p_0 = 0.2$, $p_1 = 0.8$, $\rho = 0.01$			$p_0 = 0.3$, $p_1 = 0.7$, $\rho = 0.01$		
		Delay	False alarm probability	Bayesian cost	Delay	False alarm probability	Bayesian cost
Optimal	c=0.5	0.1912	0.8427	0.9383	0.0401	0.9505	0.9706
	c=0.1	3.6891	0.1733	0.5422	2.7852	0.511	0.7895
	c=0.05	5.2904	0.0695	0.334	6.8788	0.2224	0.5663
	c=0.01	7.6478	0.0123	0.0888	14.1973	0.03	0.172
	c=0.005	8.9288	0.0044	0.049	16.1054	0.0157	0.0962
NMC-based method	c=0.5	0.2178	0.8316	0.9405	0.0137	0.9688	0.9757
	c=0.1	3.5632	0.191	0.5473	2.3779	0.5549	0.79269
	c=0.05	5.4253	0.0717	0.343	6.6513	0.2508	0.5832
	c=0.01	7.9296	0.0177	0.097	13.8156	0.0639	0.2021
	c=0.005	8.4346	0.0113	0.05347	17.3196	0.0382	0.1248
DQN-based method	c=0.5	0.017	0.9707	0.9792	0.0792	0.9528	0.9924
	c=0.1	3.8384	0.2089	0.5928	3.73	0.5373	0.9042
	c=0.05	3.2874	0.2423	0.4066	5.4489	0.4519	0.7243
	c=0.01	7.4965	0.0268	0.1017	13.0369	0.1922	0.3225
	c=0.005	7.4978	0.0539	0.0914	17.4648	0.1199	0.2072

4.6 Numerical results

To evaluate the performance of the proposed Neural Monte Carlo QCD method, different numerical examples are provided. In the following examples, we evaluate the performance of the NMC-based QCD method in i.i.d. and non-i.i.d. cases. We also test the robustness of the NMC-based QCD method using data generated by distributions that are different from the training data. Moreover, we compare the performances of the optimal solution, the DQN-based solution, and the NMC-based solution in these numerical examples.

In all the following experiments, we assume the pre-change and post-change distributions, and the prior distribution of the change point is known for the optimal QCD method. On the other hand, for training the NMC-based and DQN-based QCD rules, we only use a limited historical dataset, including data sequences and the corresponding change points. Concretely, we build the NMC-based and DQN-based QCD rules with data set including 20000 observation sequences and corresponding change points. Each sequence includes 600 observations. For the NMC-based method and DQN method, 14000 sequences are used to train the neural network, 3000 sequences are used for the validation set (tuning hyperpa-



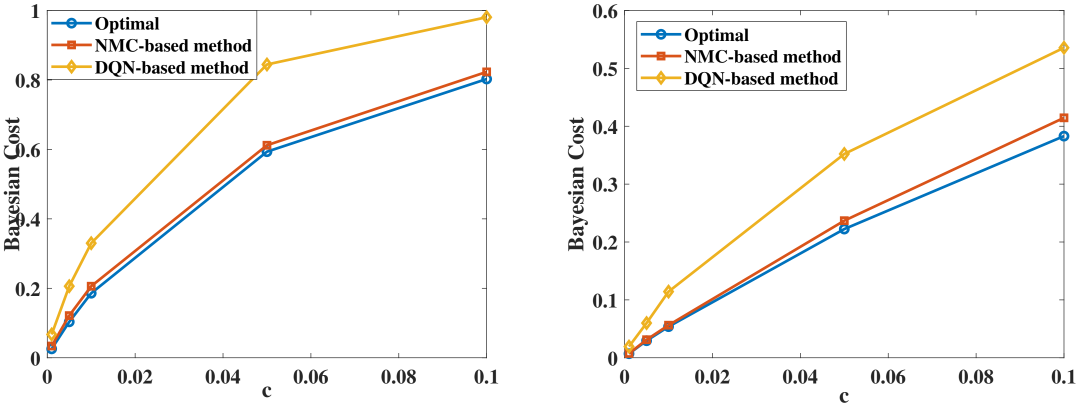
(a) $\vec{\boldsymbol{\mu}}_0 = (0,0)$, $\vec{\boldsymbol{\mu}}_1 = (4,4)$

(b) $\vec{\boldsymbol{\mu}}_0 = (0,0)$, $\vec{\boldsymbol{\mu}}_1 = (2,2)$

(c) $\vec{\boldsymbol{\mu}}_0 = (0,0)$, $\vec{\boldsymbol{\mu}}_1 = (1,1)$

Figure 4.3: The Bayesian costs of the Gaussian QCD experiment with change in mean vector

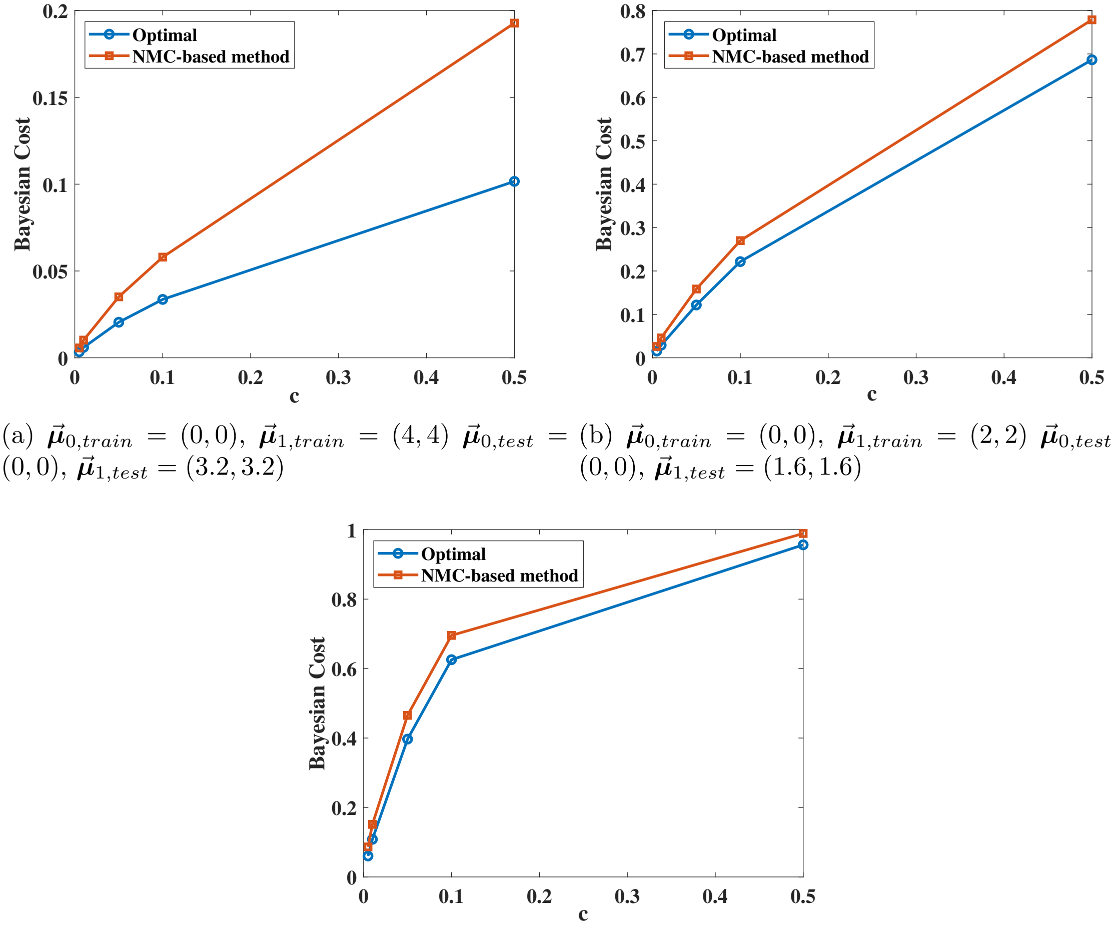
rameters such as the size of the neural network, the width of the sliding window, the data rebalance parameter $\tilde{T}$, and the best threshold $\hat{\pi}^*$, etc.) and the rest 3000 sequences are used to test the performance of these methods. The hidden layer of the neural network in the NMC-based method has 1000 nodes. The DQN model includes two hidden layers with 200 and 100 nodes respectively. ReLU is used as the activation function of all the hidden layers in these two methods. For the training of the NMC-based method, the rebalance parameter is set as $\tilde{T} = 100$. In addition, the width of sliding windows for the NMC-based method and the DQN-based method are both 10. In most of the following experiments, the DQN-based and NMC-based models are implemented following these instructions. Further information will be provided if we need to make changes to these parameters in specific experiments.



(a) $\Sigma_0 = \boldsymbol{I}_2$, $\Sigma_1 = 2\boldsymbol{I}_2$

(b) $\Sigma_0 = \boldsymbol{I}_2$, $\Sigma_1 = 4\boldsymbol{I}_2$

Figure 4.4: The Bayesian costs of the Gaussian QCD experiment with change in covariance



(a) $\vec{\boldsymbol{\mu}}_{0,train} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,train} = (4,4)$ $\vec{\boldsymbol{\mu}}_{0,test} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,test} = (3.2,3.2)$

(b) $\vec{\boldsymbol{\mu}}_{0,train} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,train} = (2,2)$ $\vec{\boldsymbol{\mu}}_{0,test} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,test} = (1.6,1.6)$

(c) $\vec{\boldsymbol{\mu}}_{0,train} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,train} = (1,1)$ $\vec{\boldsymbol{\mu}}_{0,test} = (0,0)$, $\vec{\boldsymbol{\mu}}_{1,test} = (0.8,0.8)$

Figure 4.5: Robustness test: testing on 2D Gaussian data which has mean vector different from the training data

Table 4.2: Performances of the three QCD rules in the i.i.d. 2D Gaussian experiment: change in mean vector

		$\vec{\mu}_0 = (0,0)$, $\vec{\mu}_1 = (1,1)$			$\vec{\mu}_0 = (0,0)$, $\vec{\mu}_1 = (2,2)$			$\vec{\mu}_0 = (0,0)$, $\vec{\mu}_1 = (4,4)$		
		Delay	False alarm probability	Bayesian cost	Delay	False alarm probability	Bayesian cost	Delay	False alarm probability	Bayesian cost
Optimal	c=0.5	0.2412	0.8059	0.9265	0.6128	0.1707	0.4771	0.0269	0.0091	0.02255
	c=0.1	3.4536	0.1447	0.49	1.2526	0.0236	0.1489	0.054	0.0024	0.0078
	c=0.05	4.7335	0.0574	0.294	1.4577	0.011	0.0839	0.0015	0.0659	0.0048
	c=0.01	6.6182	0.01	0.0783	1.7059	0.0044	0.0215	0.1071	0.0006	0.0017
	c=0.005	7.1565	0.0064	0.0422	2.0608	0.0012	0.0115	0.1449	0.0004	0.0011
NMC-based method	c=0.5	0.1731	0.8521	0.9387	0.227	0.6139	0.5338	0.0495	0.01644	0.04115
	c=0.1	3.374	0.1668	0.5041	1.9943	0.00361	0.203	0.109	0.00711	0.018
	c=0.05	4.5207	0.0818	0.3077	2.0039	0.0341	0.1036	0.1269	0.00411	0.01045
	c=0.01	7.0666	0.01532	0.08567	2.0391	0.0042	0.02459	0.2018	0.0011	0.0031
	c=0.005	7.5891	0.0106	0.04855	2.3923	0.002	0.01396	0.2329	0.0005	0.00166
DQN-based method	c=0.5	0.2645	0.8648	0.9971	0.5649	0.2999	0.5823	0.0589	0.0196	0.049
	c=0.1	3.7571	0.2308	0.6066	1.3214	0.0977	0.2298	0.0418	0.0171	0.0213
	c=0.05	3.7968	0.2059	0.3958	1.4892	0.0339	0.1084	0.0629	0.0112	0.0144
	c=0.01	9.2189	0.0166	0.1088	2.8778	0.0383	0.0671	0.0968	0.0024	0.0034
	c=0.005	10.2965	0.0154	0.0668	3.6808	0.0176	0.036	0.493	0.0003	0.0028

4.6.1 QCD experiments in i.i.d. case

In the first example, we study the performance of the optimal QCD solution, the NMC-based solution, and the DQN-based QCD solution when the observations are i.i.d. discrete random variables. Concretely, $f_0 = Bern(p_0)$ and $f_1 = Bern(p_1)$, where p_0 and p_1 are the parameters of the pre-change and post-change Bernoulli distributions. The prior distribution of the change-points, P_λ, is a geometric distribution with parameter $\rho = 0.01$. Based on this information, we can calculate the posterior probability and further obtain the optimal solution by dynamic programming. Using the training data set, we obtain the DQN-based and NMC-based QCD rules. After that, we compare the Bayesian costs, $C(\tau)$, of the optimal QCD solution, DQN-based method, and the NMC-based method under different Bernoulli settings on the test set. The results are shown in Fig. 4.2. From Fig. 4.2(a), compared with the DQN-based method, the Bayesian costs of the NMC-based QCD method are generally closer to that of the optimal QCD method. As the unit delay cost decreases, the performance gap between the two solutions also decreases. Besides, by comparing Fig.4.2(a) and

Table 4.3: Performances of the three QCD rules in the i.i.d. 2D Gaussian experiment: change in variance

		$\Sigma_0 = \boldsymbol{I}_2$, $\Sigma_1 = 2\boldsymbol{I}_2$			$\Sigma_0 = \boldsymbol{I}_2$, $\Sigma_1 = 4\boldsymbol{I}_2$		
		Delay	False alarm probability	Bayesian cost	Delay	False alarm probability	Bayesian cost
Optimal	c=0.1	2.6591	0.5368	0.8027	2.9756	0.0875	0.38296
	c=0.05	6.762	0.2553	0.5934	3.6813	0.0383	0.2224
	c=0.01	15.3527	0.03228	0.1858	4.7954	0.0058	0.05375
	c=0.005	18.2008	0.01242	0.1034	5.3881	0.0025	0.02944
	c=0.001	23.5036	0.0021	0.0256	5.9819	0.001	0.00698
NMC-based method	c=0.1	2.5955	0.5637	0.82325	3.1248	0.104	0.41458
	c=0.05	6.8413	0.2699	0.6119	3.9871	0.0396	0.2369
	c=0.01	15.5789	0.05033	0.206	5.1454	0.0048	0.05625
	c=0.005	18.6246	0.02835	0.1214	5.8751	0.0018	0.03117
	c=0.001	26.7755	0.007	0.03378	6.801	0.0006	0.0074
DQN-based method	c=0.1	2.4158	0.7391	0.9807	2.9831	0.2373	0.5357
	c=0.05	8.0369	0.4427	0.8445	5.6804	0.0681	0.3521
	c=0.01	16.9336	0.1606	0.3299	9.5317	0.019	0.1143
	c=0.005	23.093	0.0906	0.2061	11.5702	0.0021	0.06
	c=0.001	41.7379	0.0254	0.0671	18.4294	0.0009	0.0193

Fig.4.2(b), we can see that when the KL-divergence between pre-change and post-change distributions gets smaller, i.e., the QCD task becomes harder, the costs of the three solutions increase. Although the Bayesian cost of the NMC-based solution increases as the QCD problem becomes harder, the performance gap between the NMC-based solution and the optimal solution is still small. In addition, Table 4.1 presents the more detailed performance of the three methods, including the delay and false alarm probability. In Table 4.1, the DQN-based often achieves a lower delay or false alarm probability than the NMC-based method. But it still can not beat the NMC-based method on Bayesian cost. This result indicates that the NMC-based method achieves a better balance between the false alarm cost and delay cost than the DQN-based method. In addition, as the key of the NMC-based SCD rule, we also evaluate the estimation of the posterior probability $\Pi_t^{(0)}$. On the test set, the mean absolute errors of the posterior probability of the two experiment cases, i.e. the mean value of $|\hat{\Pi}_t^{(0)} - \Pi_t^{(0)}|$, are 0.0530 and 0.0935, respectively.

In the second numerical example, we study the performance of the three QCD methods when the observations are continuous random variables. The observations in this experiment

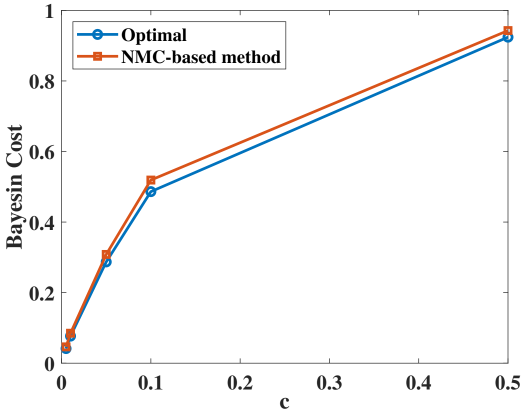


Figure 4.6: Robustness test: Change from Gaussian distribution to logistic distribution

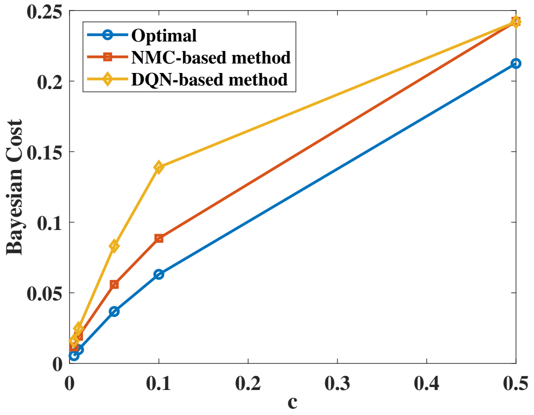


Figure 4.7: Bayesian costs of 10D Gaussian data

are 2-D Gaussian random variables with $f_0 = \mathcal{N}(\vec{\boldsymbol{\mu}}_0, \boldsymbol{\Sigma}_0)$ and $f_1 = \mathcal{N}(\vec{\boldsymbol{\mu}}_1, \boldsymbol{\Sigma}_1)$. To illustrate the performance of the NMC-based QCD method facing different kinds of changes, we study 2 different cases : (1). The change happens to the mean vector of the 2-D Gaussian distribution; (2) The change happens to the covariance matrix of the 2-D Gaussian distribution. For the first case, we set $\boldsymbol{\Sigma}_0 = \boldsymbol{\Sigma}_1 = \boldsymbol{I}_2$ and $\boldsymbol{\mu}_0 = (0, 0)$. Then we carry out three experiments with $\boldsymbol{\mu}_1 = (1, 1)$, $\boldsymbol{\mu}_1 = (2, 2)$, and $\boldsymbol{\mu}_1 = (4, 4)$, respectively. In addition, P_λ, is a geometric distribution with parameter $\rho = 0.01$. These three NMC-based models are denoted as model A, B and C. In Fig. 4.3 and Table 4.2, we compare the performance of the three QCD methods in this mean vector QCD problem. On the test set, the mean absolute errors of the posterior probability of the three models are 0.1321, 0.0319 and 0.0315, respectively. For the

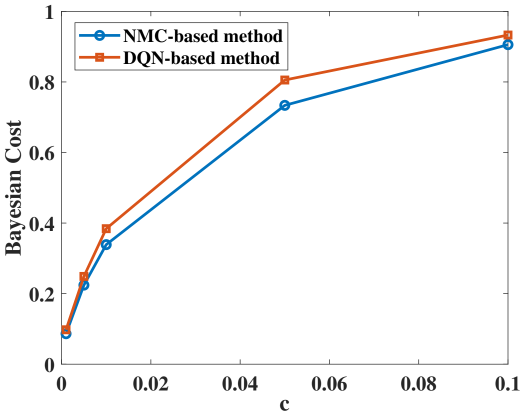
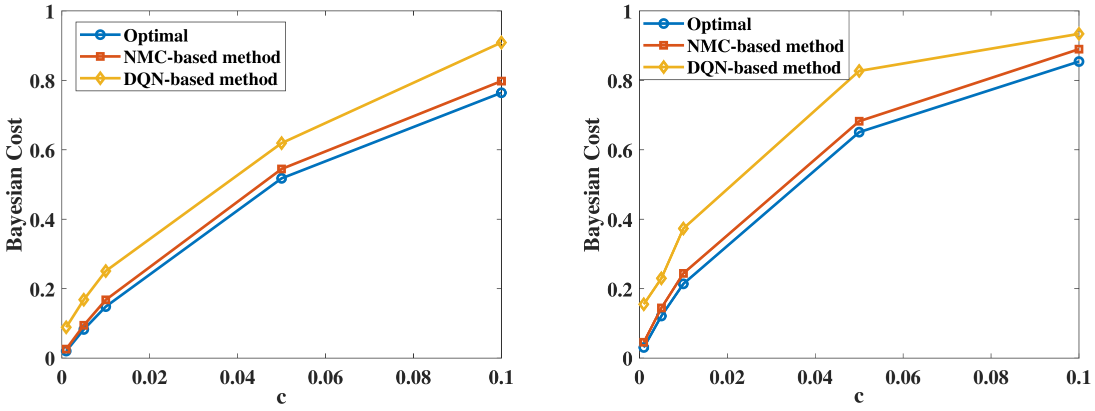


Figure 4.8: Bayesian costs of the NMC-based method and the DQN-based method in Non-i.i.d. case

second case, we set $\boldsymbol{\mu}_0 = \boldsymbol{\mu}_1 = (0,0)$ and $\boldsymbol{\Sigma}_0 = \boldsymbol{I}_2$. Then we implement two experiments with $\boldsymbol{\Sigma}_1 = 2\boldsymbol{I}_2$, $\boldsymbol{\Sigma}_1 = 4\boldsymbol{I}_2$. In Fig. 4.4 and Table 4.3, we compare the performance of the three QCD methods in this variance QCD problem. On the test set, the mean absolute errors of the posterior probability of the two cases are 0.1320 and 0.1382, respectively. In Fig. 4.3 and 4.4, the Bayesian costs achieved by the NMC-based method are close to the costs of the optimal QCD rule and are lower than the costs of the QCD-based methods. These results validate our conclusion that the NMC-based QCD method has a good performance for the 2D continuous QCD problem. From Table 4.2 and 4.3, we can also see that the NMC-based method performs better in balancing the false alarm probability and delay costs than the DQN-based method. Next, we conduct another two experiments to test the robustness of the NMC-based method in the 2D Gaussian experiment. Firstly, we test the performance of models A, B, and C on data whose mean vector is different from the training data. Concretely, the mean vectors of the testing data for Model A, B and C are $(3.2, 3.2)$, $(1.6, 1.6)$ and $(0.8, 0.8)$, receptively. Other parameters of the testing data distributions are the same as the training data. The Bayesian costs of the optimal QCD rules and the NMC-based model A, B, and C obtained on the testing data are shown in Fig. 4.5. Secondly, we investigate the performance of the NMC-based model C on data whose post-change distribution is non-Gaussian. For the testing data distribution, we define a 2-D distribution, $F_L(1,1)$ as



(a) case 1: $p_0 = 0.1, p_1 = 0.9$

(b) case 2: $p_0 = 0.2, p_1 = 0.8$

Figure 4.9: The Bayesian costs of the NMC-based and optimal QCD solutions in the HMM case

the post-change distribution of the testing data. With $F_L(1,1)$, the two elements in each data sample are independent and follow the Logistics distribution, $L(1, \sqrt{3/2\pi})$. The other parameters of the testing data are the same as the training data of NMC-based model C. The Bayesian costs of the optimal QCD rules and the NMC-based model C are shown in Fig. 4.6. In Fig. 4.5 and 4.6, the optimal solutions are obtained using the distribution of the testing data. From the results shown in Fig. 4.5 and 4.6, we can see that the performance of the NMC-based method is close to the optimal QCD rules for data generated with distributions different from the training data. These results indicate that the NMC-based QCD rule is robust to the instability of data.

In addition, we also implement an experiment for high dimensional data with a more complicated covariance matrix. Here, the observation data follows a 10D Gaussian distribution with mean vector $\boldsymbol{\mu}_1 = (1, ..., 1)$, $\boldsymbol{\mu}_1 = (0, ..., 0)$. The covariance matrix is a randomly generated positive-definite matrix. The width of the sliding window in this experiment is set as 1. The Bayesian costs of the optimal rule and the NMC-based method are shown in Fig. 4.7. From this figure, we can see that the NMC-based QCD method works well for high-dimensional data. On the test set, the mean absolute error of the posterior probability estimated by the NMC approximation model is 0.0887.

Table 4.4: Performances of the three QCD rules in the HMM experiment

		case 1: $p_0 = 0.1, p_1 = 0.9$			case 2: $p_0 = 0.2, p_1 = 0.8$		
		Delay	False alarm probability	Bayesian cost	Delay	False alarm probability	Bayesian cost
Optimal	c=0.1	3.1978	0.4444	0.7642	2.0889	0.6452	0.8541
	c=0.05	6.9195	0.1717	0.5177	6.6656	0.3174	0.6507
	c=0.01	12.0809	0.0276	0.1484	17.3519	0.0404	0.2139
	c=0.005	14.4003	0.0105	0.0825	19.9785	0.0215	0.1214
	c=0.001	18.0703	0.0023	0.0204	28.3053	0.0024	0.0307
NMC-based method	c=0.1	3.4926	0.4485	0.7978	2.3752	0.6522	0.8897
	c=0.05	6.7486	0.2073	0.54473	6.3758	0.3634	0.68219
	c=0.01	14.6158	0.0217	0.1679	17.5915	0.0681	0.2440
	c=0.005	15.4936	0.0171	0.09457	21.1274	0.0389	0.1445
	c=0.001	20.4434	0.0055	0.02594	34.2069	0.0114	0.0456
DQN-based method	c=0.1	3.2805	0.5808	0.9089	1.7592	0.7579	0.9338
	c=0.05	7.5862	0.2399	0.6192	6.1065	0.5217	0.827
	c=0.01	14.1083	0.1097	0.2508	14.8038	0.2252	0.3733
	c=0.005	21.4553	0.06093	0.1682	20.7133	0.1262	0.2298
	c=0.001	80.0278	0.00877	0.0888	15.701	0.1394	0.155

4.6.2 QCD experiments in non-i.i.d. case

To evaluate the performance of the NMC-based method with non-i.i.d. data, two numerical examples are conducted.

In the first numerical example, the data samples are generated by the Markov Gaussian sequence

$$x_t = \begin{cases} 0.5x_{t-1} + \epsilon_t, & \text{if} \quad \text{t} < \lambda \\ -0.5x_{t-1} + \epsilon_t, & \text{if} \quad \text{t} \geq \lambda \end{cases}$$

where $\epsilon_t \overset{\text{i.i.d}}{\sim} \mathcal{N}(0,1)$ for $t > 0$. Since the distribution of the current data sample only depends on the last data sample, we set $w = 2$ for both the NMC-based method and the DQN-based method. The Bayesian cost of the NMC-based rule and the QCD-based rule are shown in Fig. 4.8. From the results, we can see that the NMC-based QCD method outperforms the DQN-based methods in this non-i.i.d. QCD problem.

In the second example, we study the performance of the NMC-based method when the data follows an HMM. There are two hidden states in the HMM. For state 1: the data will generate data following Bernoulli distribution with parameter 0.1. For state 2: the data will generate data following a Bernoulli distribution with parameter 0.9. The change happens to the transition probability of the HMM. Before the change, the transition probability between states 1 and 2 is p_0. After change, the transition probability becomes to p_1. In addition, P_λ, is a geometric distribution with parameter $\rho = 0.01$. In the HMM experiment, we set the width of the sliding window $w = 15$ for both the NMC-based method and the DQN-based method. In Fig. 4.9 and Table 4.4, we compare the performances of the NMC-based QCD method and the optimal solution with different values of p_0 and p_1. Similar to the results of the i.i.d experiments, the performance of the NMC-based QCD method is still generally closer to the optimal QCD rule than the QCD method. In addition, the mean absolute errors of the posterior probability of the two experiment cases are 0.0957 and 0.1244, respectively.

Chapter 5

Conclusions and Extensions

In this chapter, we summarize the contributions of this dissertation and propose several possible extensions.

5.1 Summary and Conclusions

The goal of this dissertation is to formulate and solve more realistic change-point analysis problems. In Chapter 2, we have formulated the Bayesian two-stage sequential change diagnosis problem. We have converted the problem into two optimal single stopping time problems and obtained the optimality equations of them. After solving these equations using dynamic programming, we have obtained the optimal rule for the Bayesian two-stage SCD problem. However, the complexity of the proposed optimal solution is high due to the DP steps. To reduce the computational complexity, we have designed a threshold two-stage SCD rule and proved that this threshold rule is asymptotically optimal as the per-unit delay costs of the two stages go to zero.

In Chapter 3, we have formulated the Bayesian two-stage sequential change diagnosis over a linear sensor array problem. By analyzing the posterior probability, we have converted the multi-sensor version SCD problem to a normal SCD problem and characterized the optimal solution. However, the complexity of the proposed optimal solution is high due to the

DP steps. To reduce the computational complexity, we have designed a threshold multi-sensor two-stage SCD rule. For the general case in which the first sensor affected by the change is randomly chosen and unknown, we have proved that the threshold SCD rule is asymptotically optimal under Condition 1. For the special case that the first affected sensor is fixed and known, we have proved that the threshold rule is generally asymptotically optimal. Furthermore, we have extended the threshold SCD rule to a more general 2D sensor array case and proved its asymptotic optimality. Finally, we have analyzed how increasing the number of sensors can improve the performance of the threshold SCD rule.

In Chapter 4, we have studied the online data-driven Bayesian QCD problem with geometrically distributed change points and proposed two approaches for this problem. Firstly, After formulating the online Bayesian QCD process as a POMDP, we have proposed a DQN-based QCD rule for the data-driven Bayesian QCD problem. Secondly, inspired by the key role that the posterior false alarm probability plays in the i.i.d. QCD problem, we have proposed an NMC-based QCD rule for the data-driven Bayesian QCD problem. Trained by the Gradient Monte Carlo algorithm, a randomized neural network is applied to approximate the posterior false alarm probability. By comparing the posterior false alarm probability with a well-chosen threshold, we obtain an NMC-based QCD rule. These two methods work not only for the i.i.d. QCD problem, but also for the HMM QCD problem or the more general non-i.i.d. QCD problems. Moreover, the NMC-based method is guaranteed to converge. Numerical results have been carried out to evaluate the performance of the NMC-based method. The results have validated that the performance of the DQN-based QCD solution is generally better than the DQN-based method and close to the performance of the optimal solution.

5.2 Extensions

We expect several possible extensions of the research works related to change-point analysis as follows:

- **Online Bayesian Change Point Detection and Post-change Distribution Estimation:** In the SCD problems studied in this proposal, there is a finite candidate set of possible post-change distributions and we need to select the correct post-change distribution from the candidate set. However, in many applications in the real world, the post-change distribution is $f_\theta(x)$ and θ is the unknown parameter vector that belongs to a parameter space Ω and follows a prior distribution with pdf $h(\theta)$. In the traditional non-Bayesian change-point detection method, e.g., the GLR test, we first estimate the parameter of the post-change distribution and then use this estimation for the change-point detection. In these methods, we only care about the performance of change point detection and disregard the accuracy of the parameter estimation. However, in some specific applications, we not only want to detect the change point as soon as possible but also to estimate θ as accurately as possible. For example, in network traffic monitoring, knowing the network state after an abrupt change in the network traffic is crucial for the administrator to decide what action should be taken next to avoid congestion. Another application is online seismic event analysis. Information of the distribution of signal after an abrupt seismic event is important to know the magnitude and other properties of the seismic event.

- **Two stage SCD problem in more general change propagation model and sensor array structure:** In this dissertation, we have studied the two-stage SCD problem in linear sensor array and the 2D lattice sensor array. However, the high complexity of calculating the prevent us from applying the proposed method to more general change propagation model and sensor array structure. In terms of future work, it is of interest to investigate the proposed two-stage change diagnosis model in more general scenarios, for example, the more general change propagation models and more complex sensor arrays, the case with unknown parameters (such as v_i) and the case that the post-change distributions are unknown, etc.

- **Data driven SCD problem:** In this dissertation, we have proposed two approaches for the data-driven QCD problem. As a topic evolved from the QCD problem, the SCD problem has many similarities with the QCD problem. In the Bayesian setting, the SCD process can be abstracted to a POMDP, like the QCD process. On the other hand, the posterior probability also plays a key role in the optimal solution of the SCD problem. Therefore, it can be expected that the two approaches provided in this dissertation also can be extended to the data-driven SCD problem. It will be interesting to investigate whether these two methods can have good performance in the data-driven SCD problem.

Appendix A

Appendix of Chapter 2

A.1 Proof of Theorem 2.1

Now we consider the infinite-horizon DP and show that it is well-defined. Towards this end, we need to establish that $\lim_{T_2 \to \infty} V_k^{T_2+\tau_1}(\cdot)$ exists, which is done as the following derivation.

By an induction argument, we know that for any Π and $T_2+\tau_1$ fixed, $V_k^{T_2+\tau_1}(\Pi) \leq V_{k+1}^{T_2+\tau_1}(\Pi)$ for $k \in [\tau_1, T_2+\tau_1-1]$. Similarly, by an induction argument, we have that for any Π and $T_2+\tau_1$ fixed, $V_k^{T_2+\tau_1+1}(\Pi) \leq V_k^{T_2+\tau_1}(\Pi)$. Heuristically, this is true because the set of stopping times increases with the time upper bound $T_2+\tau_1$. With a larger set of stopping times, a lower expected cost can be achieved. Since $\max_{i,j \in \mathcal{I}_0} b_{ij} \geq V_k^{T_2+\tau_1}(\Pi) \geq 0$ for all k and $T_2+\tau_1$ for any fixed k, let $T_2+\tau_1 \to \infty$, then

$$\lim_{T_2+\tau_1 \to \infty} V_k^{T_2+\tau_1}(\Pi) = \inf_{T_2+\tau_1 : T_2+\tau_1 > k} V_k^{T_2+\tau_1}(\Pi) \triangleq V_k^{\infty}(\Pi).$$

Furthermore, the memorylessness property and the i.i.d. observation process results in the invariance of $V_k^{\infty}(\Pi)$ on k. This is shown by a simple time-shift argument. This common limit is denoted as $V(\Pi)$.

Since $V_k^{T_2+\tau_1}(\Pi)$ is decreasing with $T_2+\tau_1$ and has a well-defined limit as $T_2+\tau_1 \to \infty$, dominated convergence theorem can be applied to the bounded $G_k^{T_2+\tau_1}(\Pi)$. By doing this, we know that $\lim_{T_2+\tau_1\to\infty} G_k^{T_2+\tau_1}(\Pi)$ is well defined and independent of k. Denote the limit as $G_V(\Pi)$, it equals to

$$\mathbb{E}[V(\tilde{\Pi})|\mathcal{F}] = \int \left[V(\tilde{\Pi}(\Pi, x)) \sum_{i\in\mathcal{I}_0} f_i(x)\Pi^{(i)} \right] dx.$$

Here, $\tilde{\Pi}$ and $\tilde{X}$ denote the posterior probability and data sample at time next to the time of Π and $\mathcal{F}$. Hence, the infinite-horizon cost-to-go function for the distribution identification stage can be written as $V(\Pi) = \min(B(\Pi), c_2 + G_V(\Pi))$.

A.2 Proof of Proposition 2.4

By Proposition 2.3, for $i \in \mathcal{I}$, we know that

$$\frac{\Lambda_n(i,j)}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i - a.s.} l(i,j).$$

This implies that, for any $\varepsilon, \sigma > 0$ and any $j \in \mathcal{I}\backslash\{i\}$, there exists N_j such that

$$\mathbb{P}_i \left\{ \left| \frac{\Lambda_n(i,j)}{n} - l(i,j) \right| < \varepsilon \text{ and } \left| \frac{\Lambda_n(i,j(i))}{n} - l(i,j(i)) \right| < \varepsilon \text{ for all } n > N_j \right\} > 1 - \sigma.$$

Thus

$$\mathbb{P}_i \left\{ \frac{\Lambda_n(i,j)}{n} - \frac{\Lambda_n(i,j(i))}{n} > l(i,j) - l(i,j(i)) - 2\varepsilon \text{ for all } n > N_j \right\} > 1 - \sigma.$$

By letting $2\varepsilon < l(i,j) - l(i,j(i))$, there exists $N_j > 0$,

$$\mathbb{P}_i \left\{ \Lambda_n(i,j) > \Lambda_n(i,j(i)) \text{ for all } n > N_j \right\} > 1 - \sigma.$$

Let $N = \max_{j\in\mathcal{I}} N_j$. Summing up the cases for $i \in \mathcal{I}$, we have

$$\mathbb{P}_i\left\{-\log \sum_{j\in\mathcal{I}_0\backslash\{i\}} \exp(-\Lambda_n(i,j)) > -\log\left[M\exp(-\Lambda_n(i,j(i)))\right] \text{ for all } n > N\right\} > 1 - M\sigma.$$

Thus

$$\mathbb{P}_i\left\{\frac{\Phi_n^{(i)} - \Lambda_n(i,j(i))}{n} > -\frac{\log M}{n} \text{ for all } n > N\right\} > 1 - M\sigma.$$

By definition of $\Lambda(i,j)$ and $\Phi_n^{(i)}$, we know that

$$-\log\left[\sum_{j\in\mathcal{I}_0\backslash\{i\}} \exp(-\Lambda_n(i,j))\right] = \Phi_n^{(i)} = \log\frac{\Pi_n^{(i)}}{1-\Pi_n^{(i)}} \leq \log\frac{\Pi_n^{(i)}}{\Pi_n^{(j)}} = \Lambda_n(i,j)$$

for any $j \in \mathcal{I}\backslash\{i\}$. Therefore,

$$\mathbb{P}_i\left\{\left|\frac{\Phi_n^{(i)} - \Lambda_n(i,j(i))}{n}\right| < \frac{\log M}{n} \text{ for all } n > N\right\} > 1 - M\sigma.$$

Let $\tilde{\varepsilon} = \frac{-\log M}{K}, K > 0$. Then for $\tilde{\varepsilon} > 0$, $\tilde{\sigma} = M\sigma > 0$, there exists a $\tilde{N} = N \vee K$ such that

$$\mathbb{P}_i\left\{\left|\frac{\Phi_n^{(i)} - \Lambda_n(i,j(i))}{n}\right| < \tilde{\varepsilon} \text{ for all } n > \tilde{N}\right\} > 1 - \tilde{\sigma}.$$

Hence, by squeeze theorem and Proposition 2.3, we know that $\mathbb{P}_i$ a.s.

$$\lim_{n\to\infty}\frac{\Phi_n^{(i)}}{n} = \lim_{n\to\infty}\frac{\Lambda_n(i,j(i))}{n} = l(i,j(i)).$$

By Proposition 2.3. For $i \in \mathcal{I}$, we know that

$$\frac{\Lambda_n(i,j)}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i-a.s.} l(i,j).$$

This means that, for any $\varepsilon, \sigma > 0$ and any $j \in \mathcal{I}\backslash\{i\}$, there exists N_j such that

$$\mathbb{P}_i\left\{\left|\frac{\Lambda_n(0,j)-\Lambda_n(0,i)}{n} - l(i,j)\right| < \varepsilon \text{ for all } n > N_j\right\} > 1-\sigma.$$

Here we should note that $\Lambda_n(0,j)-\Lambda_n(0,i) = \Lambda_n(i,j)$.

If we pick $0 < \varepsilon < l(i,j)$, we will have

$$\mathbb{P}_i\left\{\Lambda_n(0,i) < \Lambda_n(0,j) \text{ for all } n > N_j\right\} > 1-\sigma.$$

Therefore

$$\mathbb{P}_i\left\{e^{-\Lambda_n(0,i)} > e^{-\Lambda_n(0,j)} \text{ for all } n > N_j\right\} > 1-\sigma.$$

Let $N = \max_{j\in\mathcal{I}} N_j$. Summing up the cases for $i \in \mathcal{I}$, we have

$$\mathbb{P}_i\left\{-\frac{1}{n}\log\left[\sum_{j\in\mathcal{I}}\exp\left(-\Lambda_n(0,j)\right)\right] > -\frac{1}{n}\log\left[M\exp\left(-\Lambda_n(0,i)\right)\right] \text{ for all } n > N\right\} > 1-M\sigma.$$

By definition of $\Lambda(0,i)$ and $\Phi_n^{(0)}$, we know that

$$-\log\left[\sum_{j\in\mathcal{I}}\exp(-\Lambda_n(0,j))\right] = \Phi_n^{(0)} = \log\frac{\Pi_n^{(0)}}{1-\Pi_n^{(0)}} \le \log\frac{\Pi_n^{(0)}}{\Pi_n^{(i)}} = \Lambda_n(0,i).$$

Hence

$$\mathbb{P}_i\left\{\left|-\frac{1}{n}\Phi_n^{(0)} - \frac{1}{n}\Lambda_n(0,i)\right| < \frac{1}{n}\log M \text{ for all } n > N\right\} > 1-M\sigma.$$

Let $\tilde{\varepsilon} = \frac{-\log M}{K}, K > 0$. Then for $\tilde{\varepsilon} > 0$, $\tilde{\sigma} = M\sigma > 0$, there exists a $\tilde{N} = N \vee K$ such that

$$\mathbb{P}_i\left\{\left|-\frac{1}{n}\Phi_n^{(0)} - \frac{1}{n}\Lambda_n(0,i)\right| < \tilde{\varepsilon} \text{ for all } n > \tilde{N}\right\} > 1-\tilde{\sigma}.$$

Therefore, by squeeze theorem and proposition 2.3, we know that $\mathbb{P}_i$ a.s.

$$\lim_{n\to\infty} \frac{\Phi_n^{(0)}}{n} = \lim_{n\to\infty} \frac{\Lambda_n(0,i)}{n} = -l(i,0).$$

A.3 Proof of Proposition 2.5

By Propositions 2.2 and 2.3, we know that for any $i \in \mathcal{I}$

$$\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A \xrightarrow[B_i\to 0]{\mathbb{P}_i - a.s.} \infty$$

and

$$\frac{\Phi_n^{(i)}}{n} \xrightarrow[n\to\infty]{\mathbb{P}_i - a.s.} l(i).$$

This means that, for any $N > 0$ and $\sigma_1 > 0$, there exists $\overline{B_i} > 0$ such that

$$\mathbb{P}_i \left\{ \tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A > N \text{ for all } B_i \leq \overline{B_i} \right\} > 1 - \sigma_1$$

and for any $\varepsilon > 0$ and $\sigma_2 > 0$, there exists $\tilde{N} > 0$ such that

$$\mathbb{P}_i \left\{ \left| \frac{\Phi_n^{(i)}}{n} - l(i) \right| < \varepsilon \text{ for all } n \geq \tilde{N} \right\} > 1 - \sigma_2.$$

Therefore, by picking $N > \tilde{N}$, we can see that, for any $\varepsilon > 0$, $\sigma_1, \sigma_2 > 0$, there exists $\overline{B_i} > 0$, such that

$$\mathbb{P}_i \left\{ \left| \frac{\Phi^{(i)}_{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A}}{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A} - l(i) \right| < \varepsilon \text{ for all } B_i \leq \overline{B_i} \right\} > 1 - \sigma_1 - \sigma_2,$$

i.e.,

$$\frac{\Phi^{(i)}_{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A}}{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A} \xrightarrow[B_i\to 0]{\mathbb{P}_i - a.s.} l(i).$$

Similarly,

$$\frac{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1} \xrightarrow[B_i\to 0]{\mathbb{P}_i-a.s.} l(i).$$

So for $\varepsilon > 0$, $\sigma > 0$, there exists $\overline{B_i} > 0$, such that

$$\mathbb{P}_i\left\{\frac{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A} < l(i)+\varepsilon \text{ and } \frac{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1} > l(i)-\varepsilon \text{ for all } B_i \le \overline{B_i}\right\} > 1-\sigma.$$

This happens if and only if

$$\mathbb{P}_i\left\{\frac{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}} > \frac{1}{l(i)+\varepsilon} \text{ and } \frac{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}} < \frac{1}{l(i)-\varepsilon} \text{ for all } B_i \le \overline{B_i}\right\} > 1-\sigma.$$

Following the proposed SCD rule, we know

$$\frac{-\log B_i}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1} \ge \frac{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}$$

and

$$\frac{-\log B_i}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A} < \frac{\Phi^{(i)}_{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}}{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}.$$

Therefore, we have

$$\mathbb{P}_i\left\{\frac{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}{-\log B_i} > \frac{1}{l(i)+\varepsilon} \text{ and } \frac{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A-1}{-\log B_i} < \frac{1}{l(i)-\varepsilon} \text{ for all } B_i \le \overline{B_i}\right\} > 1-\sigma.$$

This implies that

$$\mathbb{P}_i\left\{-\frac{\varepsilon}{l(i)\left(l(i)+\varepsilon\right)} < \frac{\tau^{(i)}_{\vec{\boldsymbol{B}}}+\tau_A}{-\log B_i} - \frac{1}{l(i)} < \frac{\varepsilon}{l(i)\left(l(i)-\varepsilon\right)} - \frac{1}{\log B_i},\ \text{ for all } B_i \le \overline{B_i}\right\} > 1-\sigma.$$

Let

$$\tilde{\varepsilon} = \frac{\varepsilon}{l(i)\left(l(i)-\varepsilon\right)} - \frac{1}{\log \overline{B_i}}.$$

Therefore, for $\tilde{\varepsilon} > 0$, $\sigma > 0$, there exists $\overline{B_i} > 0$ such that

$$\mathbb{P}_i \left\{ \left| \frac{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A}{-\log B_i} - \frac{1}{l(i)} \right| < \tilde{\varepsilon} \text{ for all } B_i \leq \overline{B_i} \right\} > 1 - \sigma.$$

So

$$\frac{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A}{-\log B_i} \xrightarrow[B_i \to 0]{\mathbb{P}_i - a.s.} \frac{1}{l(i)}.$$

In addition, we know that $\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A - \lambda \leq (\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A - \lambda)_+ \leq \tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A$ and λ is almost surely finite. Therefore, the first part of this lemma is true.

By Propositions 2.2 and 2.3, we know that for any $i \in \mathcal{I}$

$$\tau_A \xrightarrow[A \to \infty]{\mathbb{P}_i - a.s.} \infty$$

and

$$\frac{\Phi_n^{(0)}}{n} \xrightarrow[n \to \infty]{\mathbb{P}_i - a.s.} -l(i, 0).$$

This means that, for any $N > 0$ and $\sigma_1 > 0$, there exists $\overline{A} > 0$ such that

$$\mathbb{P}_i \left\{ \tau_A > N \text{ for all } A \geq \overline{A} \right\} > 1 - \sigma_1$$

and for any $\varepsilon > 0$ and $\sigma_2 > 0$, there exists $\tilde{N} > 0$ such that

$$\mathbb{P}_i \left\{ \left| \frac{\Phi_n^{(0)}}{n} - \Big(- l(i, 0) \Big) \right| < \varepsilon \text{ for all } n \geq \tilde{N} \right\} > 1 - \sigma_2.$$

Therefore, by picking $\tilde{N} > N$, we can see for any $\varepsilon > 0$, $\sigma_1, \sigma_2 > 0$, there exists $\overline{A} > 0$, such that

$$\mathbb{P}_i \left\{ \left| \frac{\Phi_{\tau_A}^{(0)}}{\tau_A} - \Big(- l(i, 0) \Big) \right| < \varepsilon \text{ for all } A \geq \overline{A} \right\} > 1 - \sigma_1 - \sigma_2,$$

i.e.,

$$\frac{\Phi_{\tau_A}^{(0)}}{\tau_A} \xrightarrow[A \to \infty]{\mathbb{P}_i - a.s.} -l(i, 0).$$

Similarly,

$$\frac{\Phi^{(0)}_{\tau_A-1}}{\tau_A-1} \xrightarrow[A\to\infty]{\mathbb{P}_i-a.s.} -l(i,0).$$

So for $\varepsilon > 0$, $\sigma > 0$, there exists $\overline{A} > 0$, such that

$$\mathbb{P}_i\left\{\frac{\Phi^{(0)}_{\tau_A}}{\tau_A} < -l(i,0)+\varepsilon \text{ and } \frac{\Phi^{(0)}_{\tau_A-1}}{\tau_A-1} > -l(i,0)-\varepsilon \text{ for all } A \geq \overline{A}\right\} > 1-\sigma.$$

This happens if and only if

$$\mathbb{P}_i\left\{\frac{\tau_A}{\Phi^{(0)}_{\tau_A}} > \frac{1}{-l(i,0)+\varepsilon} \text{ and } \frac{\tau_A-1}{\Phi^{(0)}_{\tau_A-1}} < \frac{1}{-l(i,0)-\varepsilon} \text{ for all } A \geq \overline{A}\right\} > 1-\sigma.$$

Following the proposed SCD rule, we know

$$\frac{-\log A}{\tau_A} < \frac{\Phi^{(0)}_{\tau_A}}{\tau_A}$$

and

$$\frac{-\log A}{\tau_A-1} \geq \frac{\Phi^{(0)}_{\tau_A-1}}{\tau_A-1}.$$

Therefore, we have

$$\mathbb{P}_i\left\{\frac{\tau_A}{-\log A} > \frac{1}{-l(i,0)+\varepsilon} \text{ and } \frac{\tau_A-1}{-\log A} < \frac{1}{-l(i,0)-\varepsilon} \text{ for all } A \geq \overline{A}\right\} > 1-\sigma.$$

This implies that

$$\mathbb{P}_i\left\{-\frac{\varepsilon}{-l(i,0)\left(-l(i,0)+\varepsilon\right)} < \frac{\tau_A}{-\log A} - \frac{1}{-l(i,0)} < \frac{\varepsilon}{-l(i,0)\left(-l(i,0)-\varepsilon\right)} - \frac{1}{\log A} \right.$$
$$\left. \text{for all } A \geq \overline{A}\right\} > 1-\sigma.$$

Let

$$\tilde{\varepsilon} = \frac{\varepsilon}{-l(i,0)\left(-l(i,0)-\varepsilon\right)}.$$

Therefore, for $\tilde{\varepsilon} > 0$, $\sigma > 0$, there exists $\overline{A} > 0$ such that

$$\mathbb{P}_i \left\{ \left| \frac{\tau_A}{-\log A} - \frac{1}{-l(i,0)} \right| < \tilde{\varepsilon} \text{ for all } A \geq \overline{A} \right\} > 1 - \sigma.$$

So

$$\frac{\tau_A}{-\log A} \xrightarrow[A\to\infty]{\mathbb{P}_i - a.s.} \frac{1}{-l(i,0)}.$$

A.4 Proof of Proposition 2.8

By equation (3.2),

$$\begin{aligned} \Pi_n^{(0)} &= \left[1 + \sum_{i\in\mathcal{I}} \tfrac{p_0 v_i}{1-p_0} \prod_{k=1}^{n} \tfrac{f_i(X_k)}{f_0(X_k)(1-p)} + \sum_{i\in\mathcal{I}} p v_i \sum_{k=1}^{n} \prod_{m=k}^{n} \tfrac{f_i(X_m)}{f_0(X_m)(1-p)} \right]^{-1} \\ &= \left\{ \sum_{i\in\mathcal{I}} \tfrac{p_0 v_i}{1-p_0} \exp \left(n \log \left(\tfrac{1}{1-p} \right) + \sum_{k=1}^{n} \log \tfrac{f_i(X_k)}{f_0(X_k)} \right) \right. \\ &\quad \left. + 1 + \sum_{i\in\mathcal{I}} p v_i \sum_{k=1}^{n} \exp \left[(n-k+1) \log \left(\tfrac{1}{1-p} \right) + \sum_{m=k}^{n} \log \left(\tfrac{f_i(X_m)}{f_0(X_m)} \right) \right] \right\}^{-1}. \end{aligned}$$

To analyze the value of $\Pi_n^{(0)}$, different cases should be considered here.

Case 1: If $\log \left(\frac{1}{1-p} \right) > q(0,i)$ for any $i \in \mathcal{I}$, then

$$\sum_{i\in\mathcal{I}} \tfrac{p_0 v_i}{1-p_0} \exp \left(n \log \left(\tfrac{1}{1-p} \right) + \sum_{k=1}^{n} \log \tfrac{f_i(X_k)}{f_0(X_k)} \right) \xrightarrow[n\to\infty]{\mathbb{P}_i - a.s.} \infty.$$

Thus the proposition is true in this case.

Case 2: If $\log \left(\frac{1}{1-p} \right) = q(0,i)$ for any $i \in \mathcal{I}$, then

$$\lim_{n\to\infty} \left[\tfrac{p_0 v_i}{1-p_0} \exp \left(n \log \left(\tfrac{1}{1-p} \right) + \sum_{k=1}^{n} \log \tfrac{f_i(X_k)}{f_0(X_k)} \right) \right] = \tfrac{p_0 v_i}{1-p_0},$$

which is a positive constant. So the proposition is true for this case.

Case 3: If $\log\left(\frac{1}{1-p}\right) < q(0,i)$ for any $i \in \mathcal{I}$, then

$$\begin{aligned}
&\sum_{i\in\mathcal{I}} pv_i \sum_{k=1}^{n} \exp\left[(n-k+1)\log\left(\tfrac{1}{1-p}\right) \; + \sum_{m=k}^{n} \log\left(\tfrac{f_i(X_m)}{f_0(X_m)}\right)\right] \\
=&\sum_{i\in\mathcal{I}} pv_i \exp\left(\log\left(\tfrac{1}{1-p}\right) + \log\left(\tfrac{f_i(X_n)}{f_0(X_n)}\right)\right) + \\
&\sum_{i\in\mathcal{I}} pv_i \sum_{k=1}^{n-1} \exp\left[(n-k+1)\log\left(\tfrac{1}{1-p}\right) + \sum_{m=k}^{n} \log\left(\tfrac{f_i(X_m)}{f_0(X_m)}\right)\right].
\end{aligned}$$

Under the condition $\lambda = 0$, there is a lower bound for $\log(f_i(x_n)/f_0(x_n))$. So the above quantity does not converge to 0. Therefore, there is a corresponding upper bound for $\Pi_n^{(0)}$, which is less than 1. In conclusion, the proposition is true in all cases.

A.5 Proof of Proposition 2.9

The proof of Proposition 2.9 is close to Theorem 5.1 in [58]. By Proposition 2.5,

$$\frac{(\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A - \lambda)_+}{-\log B_i} \xrightarrow[B_i \to 0]{\mathbb{P}_i - a.s.} \frac{1}{l(i)}.$$

Since λ is finite almost surely, for any $\varepsilon > 0$, $\sigma > 0$, there exists $\overline{\boldsymbol{B}}$ such that

$$\mathbb{P}_i \left\{ \left| \tfrac{\tau_{\vec{\boldsymbol{B}}}^{(i)} + \tau_A}{-\log B_i} - \tfrac{1}{l(i)} \right| > \varepsilon \text{ for all } \left|\vec{\boldsymbol{B}}\right|_\infty \leq \overline{\mathrm{B}} \right\} < \sigma,$$

where $\left|\vec{\boldsymbol{B}}\right|_\infty$ is the infinity norm of $\vec{\boldsymbol{B}}$. Thus

$$\begin{aligned}
&\mathbb{P}_i \left\{ \left| \tfrac{\tau_{\vec{\boldsymbol{B}}} + \tau_A}{-\log B_i} - \tfrac{1}{l(i)} \right| > \varepsilon \text{ for all } \left|\vec{\boldsymbol{B}}\right|_\infty \leq \overline{\mathrm{B}} \right\} \\
=&\sum_{j\in\mathcal{I}_0} \mathbb{P}_i \left\{ \left| \tfrac{\tau_{\vec{\boldsymbol{B}}}^{(j)} + \tau_A}{-\log B_i} - \tfrac{1}{l(i)} \right| > \varepsilon \text{ for all } \left|\vec{\boldsymbol{B}}\right|_\infty \leq \overline{\mathrm{B}}, \tau_{\vec{\boldsymbol{B}}} + \tau_A = \tau_{\vec{\boldsymbol{B}}}^{(j)} + \tau_A \right\} \\
\leq&\, \mathbb{P}_i \left\{ \left| \tfrac{\tau_A + \tau_{\vec{\boldsymbol{B}}}^{(j)}}{-\log B_i} - \tfrac{1}{l(i)} \right| > \varepsilon \text{ for all } \left|\vec{\boldsymbol{B}}\right|_\infty \leq \overline{\mathrm{B}} \right\} + \sum_{j\in\mathcal{I}_0\setminus\{i\}} \mathbb{P}_i \left\{ \tau_{\vec{\boldsymbol{B}}} + \tau_A = \tau_{\vec{\boldsymbol{B}}}^{(j)} + \tau_A \right\} < \sigma + v_i B_i.
\end{aligned}$$

Since $B_i \to 0$ and σ can take any positive value, for any $\overline{\sigma} = \sigma + v_i B_i > 0$ and $\varepsilon > 0$, there exists a $\overline{B} > 0$ such that

$$\mathbb{P}_i \left\{ \left| \tfrac{\tau_{\vec{\boldsymbol{B}}} + \tau_A}{-\log B_i} - \tfrac{1}{l(i)} \right| > \varepsilon \text{ for all } \left| \vec{\boldsymbol{B}} \right|_\infty \leq \overline{\mathrm{B}} \right\} < \overline{\sigma},$$

i.e.,

$$\frac{(\tau_{\vec{\boldsymbol{B}}} + \tau_A)}{-\log B_i} \xrightarrow[B_i \to 0]{\mathbb{P}_i - a.s.} \frac{1}{l(i)}.$$

Since λ is almost surely finite, the proposition is true.

A.6 Proof of Lemma 2.2

The main idea of the proof is similar to that in [67], which focuses on one-stage SCD. Here we extend and modify the techniques developed in [67] to the considered two-stage SCD case. Since the proofs of the two inequalities in this lemma follow similar steps, here we only give proof of the first one.

Before proving Lemma 2.2, some supplemental lemmas are introduced as follows.

Lemma A.1. Let $\delta = (\tau_1, \tau_2, d) \in \Delta$. For every $i \in \mathcal{I}$, $j \in \mathcal{I}_0 \backslash \{i\}$, $L > 0$, $f > 1$, then

$$\begin{aligned} &\mathbb{P}_i(\tau_1 + \tau_2 - \lambda \geq L) \geq 1 - \sum_{k \in \mathcal{I}_0 \backslash \{i\}} \tfrac{R_{ik}(\delta)}{v_i} - \sum_{k \in \mathcal{I}} R_{0k}(\delta) \\ &- \tfrac{e^{fLl(i,j)} R_{ji}(\delta)}{v_i} - \mathbb{P}_i \left\{ \sup_{n \leq \lambda + L} \Lambda_n(i,j) > fLl(i,j) \right\}. \end{aligned}$$

Proof. The misdiagnosis probabilities

$$\begin{aligned} &R_{ji}(\delta) = v_i \mathbb{E}_i[1_{\{d=i, \lambda \leq \tau_1 + \tau_2 < \infty\}} e^{-\Lambda_{\tau_1+\tau_2}(i,j)}] \\ &= \mathbb{E}[1_{\{d=i, \lambda \leq \tau_1+\tau_2 < \infty, \theta = i\}} e^{-\Lambda_{\tau_1+\tau_2}(i,j)}] \geq \mathbb{E}[1_{\{d=i, \lambda \leq \tau_1+\tau_2 < \lambda + L, \theta = i, \Lambda_{\tau_1+\tau_2}(i,j) \leq B\}} e^{-\Lambda_{\tau_1+\tau_2}(i,j)}] \\ &\geq e^{-B} \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i, \Lambda_{\tau_1+\tau_2}(i,j) \leq B\} \end{aligned}$$

for every fixed $B > 0$.

$$\begin{aligned}
&\mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i, \Lambda_{\tau_1+\tau_2}(i,j) \leq B\} \\
&\geq \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) \leq B\} \\
&= \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i\} - \\
&\quad \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} \\
&\geq \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \lambda + L, \theta = i\} - \mathbb{P}\{\theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} \\
&= \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \infty, \theta = i\} - \mathbb{P}\{d = i, \theta = i, \lambda + L \leq \tau_1 + \tau_2 < \infty\} \\
&\quad -\mathbb{P}\{\theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} \\
&\geq \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \infty, \theta = i\} - \mathbb{P}\{\theta = i, \lambda + L \leq \tau_1 + \tau_2 < \infty\} \\
&\quad \mathbb{P}\{\theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\}.
\end{aligned}$$

With this lower bound, we have

$$\begin{aligned}
R_{ji}(\delta) \geq e^{-B}\{&\mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \infty, \theta = i\} - \mathbb{P}\{\lambda + L \leq \tau_1 + \tau_2 < \infty, \theta = i\} \\
&- \mathbb{P}\{\theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\}\}
\end{aligned}$$

and hence

$$\begin{aligned}
&\mathbb{P}\{L \leq \tau_1 + \tau_2 - \lambda, \theta = i\} = \mathbb{P}\{\lambda + L \leq \tau_1 + \tau_2 < \infty, \theta = i\} \\
&\geq \mathbb{P}\{d = i, \lambda \leq \tau_1 + \tau_2 < \infty, \theta = i\} - \mathbb{P}\{\theta = i, \sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} - e^{B} R_{ji}(\delta).
\end{aligned}$$

Divide v_i on both sides,

$$\begin{aligned}&\mathbb{P}_i\{L \leq \tau_1 + \tau_2 - \lambda\} \geq \mathbb{P}_i\{d = i, \lambda \leq \tau_1 + \tau_2 < \infty\} - \tfrac{e^B R_{ji}(\delta)}{v_i} - \mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} \\ &= 1 - \sum_{k \in \mathcal{I}_0 \backslash \{i\}} \mathbb{P}_i\{d = k, \lambda \leq \tau_1 + \tau_2 < \infty\} + \mathbb{P}_i\{\tau_1 + \tau_2 < \lambda\} - \tfrac{e^B R_{ji}(\delta)}{v_i} \\ &\quad -\mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\} \\ &= 1 - \sum_{k \in \mathcal{I}_0 \backslash \{i\}} \tfrac{R_{ik}(\delta)}{v_i} - \mathbb{P}_i\{\tau_1 + \tau_2 < \lambda\} - \tfrac{e^B R_{ji}(\delta)}{v_i} - \mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\}.\end{aligned}$$

Since the stopping time is independent to the state after change, so $\mathbb{P}_i\{\tau_1 + \tau_2 < \lambda\} = \mathbb{P}\{\tau_1 + \tau_2 < \lambda\} = \sum_{k \in \mathcal{I}} R_{0k}(\delta)$. Therefore,

$$\mathbb{P}_i\{\tau_1 + \tau_2 - \lambda \geq L\} \geq 1 - \sum_{k \in \mathcal{I}_0 \backslash \{i\}} \tfrac{R_{ik}(\delta)}{v_i} - \sum_{k \in \mathcal{I}} R_{0k}(\delta) - \tfrac{e^B R_{ji}(\delta)}{v_i} - \mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > B\}.$$

Finally, the lemma is proved by setting $B = fLl(i,j)$. □

By Lemma A.1, we can easily have the following lemma.

Lemma A.2. Let $\delta = (\tau_1, \tau_2, d)$ be an SCD rule in Δ. For every $i \in \mathcal{I}$, $j \in \mathcal{I}_0 \backslash \{i\}$, $L > 0$, $f > 1$, then

$$\begin{aligned}\inf_{\delta \in \Delta(\overline{R})} \mathbb{P}_i\{\tau_1 + \tau_2 - \lambda \geq L\} \geq 1 - \sum_{k \in \mathcal{I}_0 \backslash \{i\}} \frac{\overline{R_{ik}}}{v_i} + \\ \sum_{k \in \mathcal{I}} \overline{R_{0k}} - \frac{e^{fLl(i,j)} \overline{R_{ji}}}{v_i} - \mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > fLl(i,j)\}.\end{aligned}$$

To control the probability part on the right hand side of Lemma A.2, we derive the following lemma.

Lemma A.3. For every $i \in \mathcal{I}$, $j \in \mathcal{I}_0 \backslash \{i\}$, $f > 1$, then

$$\mathbb{P}_i\{\sup_{n \leq \lambda + L} \Lambda_n(i,j) > fLl(i,j)\} \xrightarrow[L \to \infty]{} 0.$$

Proof. By Proposition 4.1 in [67], we know that $\Lambda_n(i,j)/n$ converges $\mathbb{P}_i$ a.s. to $l(i,j)$. Therefore, there must exist a $\mathbb{P}_i$ a.s. finite random variable K_f such that

$$\sup_{n>K_f} \frac{\Lambda_n(i,j)_+}{n} = \sup_{n>K_f} \frac{\Lambda_n(i,j)}{n} < (\frac{1+f}{2})l(i,j), \mathbb{P}_i \; a.s..$$

Moreover,

$$\begin{aligned}
&\lim_{L\to\infty} \mathbb{P}_i\Big\{ \sup_{n\le\lambda+L} \Lambda_n(i,j) > fLl(i,j)\Big\} \le \lim_{L\to\infty} \mathbb{P}_i\Big\{ \sup_{n\le\lambda+L} \Lambda_n(i,j)_+ > fLl(i,j)\Big\} \\
&\le \lim_{L\to\infty} \mathbb{P}_i\Big\{ \sup_{n\le K_f} \Lambda_n(i,j)_+ + \sup_{K_f<n\le\lambda+L} \Lambda_n(i,j)_+ > fLl(i,j)\Big\} \\
&\le \lim_{L\to\infty} \mathbb{P}_i\left\{ \sup_{n\le K_f} \Lambda_n(i,j)_+ + (\lambda+L) \sup_{K_f<n\le\lambda+L} \tfrac{\Lambda_n(i,j)_+}{n} > fLl(i,j)\right\} \\
&= \lim_{L\to\infty} \mathbb{P}_i\left\{ \frac{\sup_{n\le K_f} \Lambda_n(i,j)_+}{L} + \tfrac{\lambda+L}{L} \sup_{K_f<n\le\lambda+L} \tfrac{\Lambda_n(i,j)_+}{n} > fl(i,j)\right\}.
\end{aligned}$$

Since λ and K_f are $\mathbb{P}_i$ a.s. finite, then

$$\lim_{L\to\infty} \left[\frac{\sup_{n\le K_f} \Lambda_n(i,j)_+}{L} + \tfrac{\lambda+L}{L} \sup_{K_f<n\le\lambda+L} \tfrac{\Lambda_n(i,j)_+}{n} \right] = \sup_{K_f<n} \tfrac{\Lambda_n(i,j)_+}{n} \le \tfrac{f+1}{2} l(i,j) \le fl(i,j).$$

Therefore,

$$\lim_{L\to\infty} \mathbb{P}_i\left\{ \frac{\sup_{n\le K_f} \Lambda_n(i,j)_+}{L} + \tfrac{\lambda+L}{L} \sup_{K_f<n\le\lambda+L} \tfrac{\Lambda_n(i,j)_+}{n} > fl(i,j)\right\} = 0.$$

Hence Lemma A.3 is proved. □

By lemma A.2 and A.3, we have the following result.

Lemma A.4. Let $\delta = (\tau_1, \tau_2, d)$ be an SCD rule in Δ. For $0 < \gamma < 1$, $i \in \mathcal{I}$, and $j = j(i)$, then

$$\liminf_{\overline{R}\to 0} \inf_{\delta\in\Delta(\overline{R})} \mathbb{P}_i \left\{ \tau_1 + \tau_2 - \lambda \ge \frac{\gamma \left|\log(\overline{R_{j(i)i}}/v_i)\right|}{l(i)} \right\} \ge 1.$$

Proof. If we set $j = j(i)$ and $L = \frac{\gamma\left|\log(\overline{R_{j(i)i}}/v_i)\right|}{l(i)}$, and choose $f > 1$ such that $0 < f\gamma < 1$, then

$L \to \infty$ as $\overline{R} \to 0$. Then plug them in Lemma A.2 and apply Lemma A.3, we have

$$\begin{aligned}
&\liminf_{\overline{R}\to 0} \inf_{(\delta)\in\Delta(\overline{R})} \mathbb{P}_i\left\{ \tau_1+\tau_2-\lambda \ge \frac{\gamma\left|\log(\overline{R_{j(i)i}}/v_i)\right|}{l(i)} \right\} \\
&\ge \liminf_{\overline{R}\to 0} \left[1 - \sum_{k\in\mathcal{I}_0\setminus\{i\}} \frac{\overline{R_{ik}}}{v_i} - \sum_{k\in\mathcal{I}} \overline{R_{0k}} - (\frac{\overline{R_{j(i)i}}}{v_i})^{1-f\gamma} - \mathbb{P}_i\left\{ \sup_{n\le\lambda+L} \Lambda_n(i,j(i)) > fLl(i) \right\} \right] \\
&= 1 - o(1).
\end{aligned}$$

□

Now we prove the first inequality in Lemma 2.2. Fix a set of positive constants $\overline{R}$, $0 < \gamma < 1$ and $\delta = (\tau_1, \tau_2, d) \in \Delta$. By Markov inequality

$$\begin{aligned}
\mathbb{E}_i\left[\tfrac{(\tau_1+\tau_2-\lambda)_+}{\left|\log(\overline{R_{j(i)i}}/v_i)\right|/l(i)}\right] &\ge \gamma\mathbb{P}_i\left[\tfrac{(\tau_1+\tau_2-\lambda)_+}{\left|\log(\overline{R_{j(i)i}}/v_i)\right|/l(i)} \ge \gamma\right] \\
&\ge \gamma \inf_{\tilde{\delta}\in\Delta(\overline{R})} \mathbb{P}_i\left[(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+ \ge \tfrac{\gamma}{l(i)}\left|\log(\overline{R_{j(i)i}}/v_i)\right|\right].
\end{aligned}$$

Here $\tilde{\delta} = (\tilde{\tau}_1, \tilde{\tau}_2, \tilde{d})$ is any SCD rule in $\Delta(\overline{R})$. Hence

$$\inf_{\tilde{\delta}\in\Delta(\overline{R})} \mathbb{E}_i\left[\tfrac{(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+}{\left|\log(\overline{R_{j(i)i}}/v_i)\right|/l(i)}\right] \ge \gamma \inf_{\tilde{\delta}\in\Delta(\overline{R})} \mathbb{P}_i\left[(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+ \ge \tfrac{\gamma}{l(i)}\left|\log(\overline{R_{j(i)i}}/v_i)\right|\right].$$

Therefore,

$$\begin{aligned}
&\liminf_{\overline{R}\to 0} \inf_{\tilde{\delta}\in\Delta(\overline{R})} \mathbb{E}_i\left[\tfrac{(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+}{\left|\log(\overline{R_{j(i)i}}/v_i)\right|/l(i)}\right] \ge \\
&\gamma \liminf_{\overline{R}\to 0} \inf_{\tilde{\delta}\in\Delta(\overline{R})} \mathbb{P}_i\left[(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+ \ge \tfrac{\gamma}{l(i)}\left|\log(\overline{R_{j(i)i}}/v_i)\right|\right] \underset{(a)}{\ge} \gamma.
\end{aligned}$$

The inequality (a) is due to Lemma A.4 and the fact $(\tilde{\tau}_1+\tilde{\tau}_2-\lambda)_+ \ge (\tilde{\tau}_1+\tilde{\tau}_2-\lambda)$. Finally, the first inequality in (2.39) is proved since γ is arbitrary constant between 0 and 1. The proof of second inequality in (2.39) is similar and thus omitted.

A.7 Proof of Proposition 2.10

The proof follows the idea of the proof of Proposition 6.2 in [67]. Here we extend the technique in [67] to the two-stage SCD case considered in this section. Assume that

$$\lim_{c_2 \to 0} \frac{\inf_{\delta \in \Delta} C^{(c_2)}(\delta)}{C^{(c_2)}(\delta_T)} < 1$$

for contradiction. This means that there exists a monotonically decreasing sequence $\{c_{2,n}\}_{n \geq 1} \to 0$ and their corresponding SCD rules $\delta^*_{c_{2,n}} = (\tau^*_{1,c_{2,n}}, \tau^*_{2,c_{2,n}}, d^*_{2,c_{2,n}})$ such that

$$\lim_{n \to \infty} \frac{C^{(c_{2,n})}(\delta^*_{c2,n})}{C^{(c_{2,n})}(\delta_T)} < 1.$$

Since we know that $C^{(c_2)}(\delta_T) \to 0$ as $c_2 \to 0$. Therefore, $C^{(c_{2,n})}(\delta_T) \to 0$ as $n \to \infty$. This further implies that

$$\begin{cases} R_0(\delta^*_{c_{2,n}}) \xrightarrow[n \to \infty]{} 0, \\ R_{ij}(\delta^*_{c_{2,n}}) \xrightarrow[n \to \infty]{} 0, i \in \mathcal{I}_0, j \in \mathcal{I}_0 \backslash \{i\}. \end{cases}$$

By Lemma 2.2, as these false alarm and misdiagnosis probabilities go to 0,

$$\begin{cases} \mathbb{E}_i[(\tau^*_{1,c_{2,n}} - \lambda)_+] \geq \inf_{\tau_1 \in \Delta_1(R_0(\delta^*_{c_{2,n}}))} \mathbb{E}[(\tau_1 - \lambda)_+] \\ \geq \left| \log \frac{R_0(\tau^*_{1,c_{2,n}})}{v_i} \right| / l(i,0) \\ \mathbb{E}_i[(\tau^*_{1,c_{2,n}} + \tau^*_{2,c_{2,n}} - \lambda)_+] \\ \geq \inf_{\delta \in \Delta(R(\delta^*_{c_{2,n}}))} \mathbb{E}_i[(\tau_1 + \tau_2 - \lambda)_+] \\ \geq \left| \log \frac{R_{j(i)i}(\delta^*_{c_{2,n}})}{v_i} \right| / l(i). \end{cases} \tag{A.1}$$

Now we can apply these results to analyze the total Bayesian cost. We know that $\tau^*_{1,c_{2,n}} > \lambda$ a.s. when the false alarm of the first stage goes to 0. Then, as $n \to \infty$ (i.e. $c_{2,n} \to 0$),

$$\begin{aligned}
&C^{(c_{2,n})}(\delta^*_{c_{2,n}}) = c_{2,n} \sum_{i\in\mathcal{I}} v_i \mathbb{E}_i[(\tau^*_{1,c_{2,n}} + \tau^*_{2,c_{2,n}} - \lambda)] + c_{2,n}\left(\tfrac{1}{r} - 1\right) \sum_{i\in\mathcal{I}} v_i \mathbb{E}_i[(\tau^*_{1,c_{2,n}} - \lambda)] \\
&\quad + \sum_{i\in\mathcal{I}_0} \sum_{j\in\mathcal{I}_0\setminus\{i\}} b_{ji} R_{ji}(\delta^*_{c_{2,n}}) + aR_0(\delta^*_{c_{2,n}}) \\
&\geq \sum_{i\in\mathcal{I}} \left[c_{2,n} v_i \mathbb{E}_i[(\tau^*_{1,c_{2,n}} + \tau^*_{2,c_{2,n}} - \lambda)] + b_{j(i)i} R_{j(i)i}(\delta^*_{c_{2,n}}) \right] \\
&\quad + c_{2,n}\left(\tfrac{1}{r} - 1\right) \sum_{i\in\mathcal{I}} v_i \mathbb{E}_i[(\tau^*_{1,c_{2,n}} - \lambda)] + aR_0(\delta^*_{c_{2,n}}) \\
&\underset{(a)}{\geq} \underbrace{\sum_{i\in\mathcal{I}} v_i \left[\frac{c_{2,n}}{-l(i)} \log \frac{R_{j(i)i}(\delta^*_{c_{2,n}})}{v_i} + \frac{b_{j(i)i}}{v_i} R_{j(i)i}(\delta^*_{c_{2,n}}) \right]}_{\text{Item1}} \\
&\quad \underbrace{+ aR_0(\delta^*_{c_{2,n}}) - c_{2,n}\left(\frac{1}{r} - 1\right) \sum_{i\in\mathcal{I}} \frac{v_i \log R_0(\delta^*_{c_{2,n}})}{l(i,0)}}_{\text{Item2}} \underbrace{+ c_{2,n}\left(\frac{1}{r} - 1\right) \sum_{i\in\mathcal{I}} \frac{v_i \log v_i}{l(i,0)}}_{\text{Item3}}.
\end{aligned}$$

Here, inequality (a) is due to (A.1).

Since $(1/r - 1) \sum_{i\in\mathcal{I}} (v_i \log v_i / l(i,0))$ is a finite constant and

$$\begin{cases}
\left(\frac{1}{r} - 1\right) \sum_{i\in\mathcal{I}} \frac{v_i \log R_0(\delta^*_{c_{2,n}})}{l(i,0)} \to \infty, \\
\frac{1}{-l(i)} \log\left(\frac{R_{j(i)i}(\delta^*_{c_{2,n}})}{v_i}\right) \to \infty,
\end{cases}$$

as $n \to \infty$, Item 3 is negligible compared with Item 1 and Item 2. So we can conclude that $C^{(c_{2,n})}(\delta^*_{c_{2,n}}) \geq \text{Item 1} + \text{Item 2}$. Let

$$\begin{cases}
A' = \frac{1}{R_0(\delta^*_{c_{2,n}})}, \\
\vec{\boldsymbol{B}}' = (B'_1, \ldots, B'_M), \\
B'_i = \frac{R_{j(i)i}(\delta^*_{c_{2,n}})}{v_i}, i \in \mathcal{I}.
\end{cases}$$

Then, we have

$$Item1 + Item2 = \sum_{i \in \mathcal{I}} v_i \left[\frac{c_{2,n}}{-l(i)} \log \left(B_i' \right) + b_{j(i)i} B_i' \right] + c_{2,n} \left(\frac{1}{r} - 1 \right) \sum_{i \in \mathcal{I}} \frac{v_i}{l(i,0)} \log A' + \frac{a}{A'}.$$

Now we can find that Item 1+ Item 2 have a very similar form of the Bayesian cost function of the threshold rule, $C^{(c_2)}(\delta_T)$. But there are two differences between them. One difference is that the false alarm probability in item 2 is $1/A'$, while it is $k_a/(1+A)$ in $C^{(c_2)}(\delta_T)$. But they are almost equivalent when A and A' goes to infinity. The other difference is the coefficient of false alarm in Item 1 is $b_{j(i)i}$, while it's k_i in $C^{(c_2)}(\delta_T)$. However, as we discussed in Section 2.4, taking different value of k_i will not change the asymptotic behavior of the Bayesian cost. So this difference becomes negligible as $c_2 \to 0$. So we can conclude that

$$C^{(c_{2,n})}(\delta^*_{c_{2,n}}) \geq Item1 + Item2 \approx C^{(c_{2,n})}(A', \vec{\boldsymbol{B}}') \geq C^{(c_{2,n})}(\delta_T),$$

where $C^{(c_{2,n})}(A', \vec{\boldsymbol{B}}')$ is the cauculated as (2.29) with thresholds A' and $\vec{\boldsymbol{B}}'$. As the result, the proposition is true.

Appendix B

Appendix of Chapter 3

B.1 Proof of Proposition 3.1

Before we prove Proposition 3.1, we introduce some helpful results.

Lemma B.1. Let $\{\xi_k\}_{k\geq 1}$ be a positive stochastic process and T be an a.s. finite random time defined on the same probability space $(\Omega, \varepsilon, \mathbb{P})$. Given T, the random variables $\{\xi_k\}_{k\geq 1}$ are conditionally independent, and $\{\xi_k\}_{1\leq k\leq T-1}$ and $\{\xi_k\}_{k\geq T}$ have common conditional probability distributions $\mathbb{P}^{(\infty)}$ and $\mathbb{P}^{(0)}$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, the expectations with respect to which are denoted by $\mathbb{E}^{(\infty)}$ and $\mathbb{E}^{(0)}$, respectively. Suppose that $\mathbb{E}^{(\infty)}[log\xi_1]$ and $\mathbb{E}^{(0)}[log\xi_1]$ exist, and define $\eta := \mathbb{E}^{(0)}[log\xi_1]$. Then for any fixed constant $c > 0$

$$\tfrac{1}{k}\log\left(c + \sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} \eta_+. \tag{B.1}$$

This lemma is the first part of Lemma 5.5 in the paper [67]. Here we further extend this lemma so that it can be applied to our sensor array problem.

Lemma B.2. Let $\{\xi_k\}_{k\geq 1}$ be a positive stochastic process and $T_{L-1} \leq T_L$ are two a.s. finite random times defined on the same probability space $(\Omega, \varepsilon, \mathbb{P})$. Given T_{L-1} and T_L, the random variables $\{\xi_k\}_{k\geq 1}$ are conditionally independent, and $\{\xi_k\}_{T_{L-1}\leq k\leq T_L-1}$ and $\{\xi_k\}_{k\geq T_L}$ have common conditional probability distributions $\mathbb{P}^{(\infty)}$ and $\mathbb{P}^{(0)}$ on $(\mathbb{R}, \mathcal{B}(\mathbb{R}))$, the expectations with respect to which are denoted by $\mathbb{E}^{(\infty)}$ and $\mathbb{E}^{(0)}$, respectively. Suppose that $\mathbb{E}^{(\infty)}[log\xi_1]$ and $\mathbb{E}^{(0)}[log\xi_1]$ exist, $0 < \xi_k < \infty$ for all $k \geq 1$ and define $\eta := \mathbb{E}^{(0)}[log\xi_1]$. Then for any fixed constant $c > 0$

$$\tfrac{1}{k}\log\left(c + \sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} \eta_+. \tag{B.2}$$

Proof.

$$\begin{aligned}
&\tfrac{1}{k}\log(c + \sum_{l=1}^{k}\prod_{n=1}^{l}\xi_n) \\
&= \tfrac{1}{k}\log\left(c + \sum_{l=1}^{T_{L-1}-1}\prod_{n=1}^{l}\xi_n + \sum_{l=T_{L-1}}^{k}\prod_{n=1}^{l}\xi_n\right) \\
&= \tfrac{1}{k}\log\left(\left(c + \sum_{l=1}^{T_L-1}\prod_{n=1}^{l}\xi_n\right)\prod_{n=1}^{T_L-1}\xi_n^{-1} + \sum_{l=T_{L-1}}^{k}\prod_{n=T_{L-1}}^{l}\xi_n\right) + \tfrac{1}{k}\log\left(\prod_{n=1}^{T_{L-1}-1}\xi_n\right).
\end{aligned}$$

The last equality holds by setting

$$c' = \left(c + \sum_{l=1}^{T_L-1}\prod_{n=1}^{l}\xi_n\right)\prod_{n=1}^{T_L-1}\xi_n^{-1} > 0.$$

By Lemma B.1, we can see that

$$\tfrac{1}{k}\log\left(c' + \sum_{l=T_{L-1}}^{k}\prod_{n=T_{L-1}}^{l}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} \eta_+.$$

On the other hand, since T_{L-1} is a.s. finite, we have

$$\tfrac{1}{k}\log\left(\prod_{n=1}^{T_{L-1}-1}\xi_n\right) \xrightarrow[k\to\infty]{\mathbb{P}-a.s.} 0.$$

Then the lemma is proved. □

Now, we first prove that, for any $i, j \in \mathcal{I}$, we have

$$\liminf_{k\to\infty}\frac{1}{k}\log H_k^{(i)} \geq (\log(1-\rho) + Lq(j,i) - Lq(j,0))_+ \tag{B.3}$$

$\mathbb{P}_j$ almost surely.

For any $i \in \mathcal{I}$, define

$$\begin{cases} \xi_k^{(l)}(i) = \frac{f_0(x_{k,l})}{f_i(x_{k,l})}, 2 \leq l \leq L-1, \\ \xi_k^{(1)}(i) = \frac{f_0(x_{k,l})}{f_i(x_{k,l})}(1-\rho_1), \\ \xi_k^{(L)}(i) = \frac{f_0(x_{k,l})}{f_i(x_{k,l})}(1-\rho_2). \end{cases} \tag{B.4}$$

With this definition, we can have

$$H_k^{(i)} = \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \left(\prod_{n=1}^{n_s-1} \left(\frac{(1-\rho)\xi_n^{(s)}(i)}{(1-\rho_1)(1-\rho_2)} \right) \right) \cdot \psi_{s-1}^{(i)}(k,n_s)\phi_{s+1}^{(i)}(k,n_s) \tag{B.5}$$

where

$$\begin{cases} \psi_l^{(i)}\left(k, n_{l+1}\right) = \prod\limits_{n=1}^{k}\left(1-\rho_1\right)\prod\limits_{t=1}^{l}\xi_n^{(t)}(i) + \rho_1 \sum\limits_{n_l=n_{l+1}}^{k} \prod\limits_{n=1}^{n_l-1} \xi_n^{(l)}(i)\psi_{l-1}^{(i)}\left(k,n_l\right), L-1 \geq l \geq 1 \\ \psi_0^{(i)}\left(k,n_1\right) = 1 \end{cases}, \tag{B.6}$$

and

$$\begin{cases} \phi_l^{(i)}\left(k, n_{l-1}\right) = \prod\limits_{n=1}^{k}\left(1-\rho_2\right)\prod\limits_{t=l}^{L}\xi_n^{(t)}(i) \\ +\rho_2 \sum\limits_{n_l=n_{l-1}}^{k} \prod\limits_{n=1}^{n_l-1} \xi_n^{(l)}(i)\phi_{l+1}^{(i)}\left(k,n_l\right), 2 \leq l \leq L \ . \\ \phi_{L+1}^{(i)}\left(k,n_L\right) = 1. \end{cases} \tag{B.7}$$

Then, we can see that

$$\begin{cases} \psi_l^{(i)}(k,n_{l+1}) \geq \rho_1 \left(\prod\limits_{n=1}^{n_{l+1}-1} \xi_n^{(l)}(i)\psi_{l-1}^{(i)}(k,n_{l+1}) \right), L-1 \geq l \geq 1 \\ \psi_l^{(i)}(k,n_{l+1}) \geq \rho_1 \left(\prod\limits_{n=1}^{k-1} \xi_n^{(l)}(i)\psi_{l-1}^{(i)}(k,k) \right), \\ \phi_l^{(i)}(k,n_{l-1}) \geq \rho_2 \left(\prod\limits_{n=1}^{n_{l-1}-1} \xi_n^{(l)}(i)\phi_{l+1}^{(i)}(k,n_{l-1}) \right), 2 \leq l \leq L \\ \phi_l^{(i)}(k,n_{l-1}) \geq \rho_2 \left(\prod\limits_{n=1}^{k-1} \xi_n^{(l)}(i)\phi_{l+1}^{(i)}(k,k) \right). \end{cases} \tag{B.8}$$

Applying equation (B.8) repeatedly, we have

$$\begin{aligned}&H_k^{(i)} \geq \sum_{s=1}^{L} \kappa_s \sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} \tfrac{(1-\rho) f_0(x_{s,n})}{(1-\rho_1)(1-\rho_2) f_i(x_{s,n})} \cdot \left[\rho_1^{s-1} \left(\prod_{t=1}^{s-1} \prod_{n=1}^{n_s-1} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right) (1-\rho_1)^{n_s-1}\right] \cdot \\ &\left[\rho_2^{L-s-1} \left(\prod_{t=s+1}^{L} \prod_{n=1}^{n_s-1} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right) (1-\rho_2)^{n_s-1}\right] = \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1}\right) \left(\sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} (1-\rho) \prod_{t=1}^{L} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right).\end{aligned}$$

Then we have

$$\tfrac{1}{k} \log H_k^{(i)} \geq \tfrac{1}{k} \log \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1}\right) + \tfrac{1}{k} \log \left(\sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} (1-\rho) \prod_{t=1}^{L} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right). \tag{B.9}$$

Since the parameters ρ_{s-1} are all positive for all $1 \leq s \leq L$, we have

$$\frac{1}{k} \log \left(\sum_{s=1}^{L} \kappa_s \rho_1^{s-1} \rho_2^{L-s-1}\right) \xrightarrow[k\to\infty]{} 0. \tag{B.10}$$

Since the change will happen at all sensors at an almost surely finite time T, then by applying Lemma B.2, we have

$$\begin{aligned}&\tfrac{1}{k} \log \left(\sum_{n_l=0}^{k} \prod_{n=1}^{n_l-1} (1-\rho) \prod_{t=1}^{L} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right) = \tfrac{1}{k} \log \left(2 + \sum_{n_l=2}^{k} \prod_{n=1}^{n_l-1} (1-\rho) \prod_{t=1}^{L} \tfrac{f_0(x_{t,n})}{f_i(x_{t,n})}\right) \\ &\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} \left(\log(1-\rho) + Lq(j,i) - Lq(j,0)\right)_+.\end{aligned} \tag{B.11}$$

Combining (B.9), (B.10) and (B.11), we can see that (B.3) is proved. Next we need to prove the other direction, i.e., for any $i, j \in \mathcal{I}$,

$$\limsup_{k\to\infty} \frac{1}{k} \log H_k^{(i)} \leq \left(\log(1-\rho) + Lq(j,i) - Lq(j,0)\right)_+, \tag{B.12}$$

$\mathbb{P}_j$ almost surely.

For any integer $n_x \geq 0$, we can see that for $L-1 \geq l \geq 1$

$$\begin{aligned}\prod_{n=1}^{k}(1-\rho_1)\prod_{t=1}^{l}\xi_n^{(t)}(i) &= \left(\prod_{n=1}^{k}\xi_n^{(l)}(i)\right)\left(\prod_{n=1}^{k}(1-\rho_1)\prod_{t=1}^{l-1}\xi_n^{(t)}(i)\right) &\text{(B.13)}\\ &\leq \left(\prod_{n=1}^{k}\xi_n^{(l)}(i)\right)\psi_{l-1}^{(i)}(k,n_x). &\text{(B.14)}\end{aligned}$$

Similarly, for $2 \leq l \leq L$ we have

$$\prod_{n=1}^{k}(1-\rho_2)\prod_{t=l}^{L}\xi_n^{(t)}(i) \leq \left(\prod_{n=1}^{k}\xi_n^{(l)}(i)\right)\phi_{l+1}^{(i)}(k,n_x). \tag{B.15}$$

From (B.7), using (B.14) and (B.15) with $n_x = k+1$ and the fact that $\rho_1 < 1$ and $\rho_2 < 1$, we can see that

$$\begin{cases}\psi_l^{(i)}(k,n_{l+1}) \leq \sum_{n_l=n_{l+1}}^{k+1}\left(\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\right)\psi_{l-1}^{(i)}(k,n_l), L-1 \geq l \geq 1\\ \phi_l^{(i)}(k,n_{l-1}) \leq \sum_{n_l=n_{l-1}}^{k+1}\left(\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\right)\phi_{l+1}^{(i)}(k,n_l), 2 \leq l \leq L.\end{cases} \tag{B.16}$$

Applying these two inequalities in (B.16) recursively, we have

$$H_k^{(i)} \leq \sum_{s=1}^{L}\kappa_s\sum_{n_s=0}^{k}\left(\prod_{n=1}^{n_s-1}\left(\frac{(1-\rho)\xi_k^{(s)}(i)}{(1-\rho_1)(1-\rho_2)}\right)\right)\cdot\tilde{\psi}_{s-1}^{(i)}(k,n_s)\tilde{\phi}_{s+1}^{(i)}(k,n_s) \tag{B.17}$$

where

$$\begin{cases}\tilde{\psi}_l^{(i)}(k,n_{l+1}) = \sum_{n_l=n_{l+1}}^{k+1}\left(\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\right)\tilde{\psi}_{l-1}^{(i)}(k,n_l), L-1 \geq l \geq 1\\ \tilde{\psi}_0^{(i)}(k,n_1) = 1,\\ \tilde{\phi}_l^{(i)}(k,n_{l-1}) = \sum_{n_l=n_{l-1}}^{k+1}\left(\prod_{n=1}^{n_l-1}\xi_n^{(l)}(i)\right)\tilde{\phi}_{l+1}^{(i)}(k,n_l), 2 \leq l \leq L\\ \tilde{\phi}_{L+1}^{(i)}(k,n_L) = 1.\end{cases} \tag{B.18}$$

Since n_l in (B.17) is no larger than n_L and n_1 in (B.18), so the right hand side of (B.17) will become larger if we cancel all $(1-\rho_1)$ and $(1-\rho_2)$ in (B.17). Furthermore, we know that

$$\begin{cases} \tilde{\psi}_l^{(i)}(k, n_{l+1}) \leq \sum\limits_{n_l=0}^{k+1} \left(\prod\limits_{n=1}^{n_l-1} \xi_n^{(l)}(i) \right) \tilde{\psi}_{l-1}^{(i)}(k, n_l), L-1 \geq l \geq 1 \\ \tilde{\phi}_l^{(i)}(k, n_{l-1}) \leq \sum\limits_{n_l=0}^{k+1} \left(\prod\limits_{n=1}^{n_l-1} \xi_n^{(l)}(i) \right) \tilde{\phi}_{l+1}^{(i)}(k, n_l), 2 \leq l \leq L. \end{cases} \tag{B.19}$$

By canceling all $(1-\rho_1)$ and $(1-\rho_2)$ in (B.17) and applying (B.19) reversely, we have that $H_k^{(i)} \leq \sum\limits_{s=1}^{L} \kappa_s \gamma_s$ where

$$\gamma_s = \left(\sum_{n_s=0}^{k} \left(\prod_{n=1}^{n_s-1} \left((1-\rho) \xi_k^{(s)}(i) \right) \right) \right) \cdot \left(\prod_{l_1=1}^{s-1} \left(\sum_{t=0}^{k+1} \left(\prod_{n=1}^{t-1} \xi_n^{(l_1)}(i) \right) \right) \right) \cdot \left(\prod_{l_2=s+1}^{L} \left(\sum_{t=0}^{k+1} \left(\prod_{n=1}^{t-1} \xi_n^{(l_2)}(i) \right) \right) \right). \tag{B.20}$$

By Lemma B.2, for any $1 \leq s \leq L$ we have

$$\frac{1}{k} \log\left(\gamma_s\right) \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} (L-1) \left(q(j,i) - q(j,0) \right)_+ + \left(\log(1-\rho) + q(j,i) - q(j,0) \right)_+ .$$

Since $\kappa_s \geq 0$ and $\sum\limits_{s=1}^{L} \kappa_s = 1$, we have

$$\min\left(\log\left(\tfrac{\gamma_1}{k}\right), \log\left(\tfrac{\gamma_2}{k}\right), \ldots, \log\left(\tfrac{\gamma_L}{k}\right) \right) \leq \frac{\log\left(\sum\limits_{s=1}^{L} \kappa_s \gamma_s \right)}{k} \leq \max\left(\log\left(\tfrac{\gamma_1}{k}\right), \log\left(\tfrac{\gamma_2}{k}\right), \ldots, \log\left(\tfrac{\gamma_L}{k}\right) \right).$$

We can have

$$\frac{1}{k} \log\left(\sum_{s=1}^{L} \kappa_s \gamma_s \right) \xrightarrow[k \to \infty]{\mathbb{P}_j - a.s.} (L-1) \left(q(j,i) - q(j,0) \right)_+ + \left(\log(1-\rho) + q(j,i) - q(j,0) \right)_+ .$$

When Condition 1 is satisfied, we have

$$\left(\log(1-\rho) + Lq(j,i) - Lq(j,0) \right)_+ = \left(\log(1-\rho) + q(j,i) - q(j,0) \right)_+ + (L-1) \left(q(j,i) - q(j,0) \right)_+ . \tag{B.21}$$

Hence (B.12) is proved. Therefore, Proposition 3.1 is true.

B.2 Proof of Proposition 3.3

Now, we first prove that, for any $i, j \in \mathcal{I}$,

$$\liminf_{k\to\infty} \frac{1}{k} \log H_k^{(i)} \geq h(i,j) \tag{B.22}$$

$\mathbb{P}_j$ almost surely. Please note that the $h(i,j)$ in this section is defined as (3.27) since we are studying the special case.

In (B.8), we have four inequalities about $\psi_l^{(i)}(k, n_{l+1})$ and $\phi_l^{(i)}(k, n_{l-1})$. For (3.24), we apply the first inequality of (B.8) to $\{\psi_l^{(i)}\}_{s-1\geq l\geq a_2^{M(i,j)}(i,j)+1}$, the second inequality of (B.8) to $\{\psi_l^{(i)}\}_{1\leq l\leq a_2^{M(i,j)}(i,j)}$, the third inequality of (B.8) to $\{\phi_l^{(i)}\}_{s+1\leq l\leq b_2^{N(i,j)}(i,j)-1}$ and the fourth inequality of (B.8) to $\{\phi_l^{(i)}\}_{L\geq l\geq b_2^{N(i,j)}(i,j)+1}$. Then we have

$$H_k^{(i)} \geq \sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} \left(\tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)} \prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \xi_n^{(l)}(i) \right) \cdot \left(\prod_{l=1}^{a_2^{M(i,j)}(i,j)} \prod_{n=1}^{k-1} \xi_n^{(l)}(i) \right) \left(\prod_{l=b_2^{N(i,j)}(i,j)}^{L} \prod_{n=1}^{k-1} \xi_n^{(l)}(i) \right) \cdot$$
$$\rho_1{}^{s-1} \rho_2{}^{L-s} (1-\rho_1)^{k-1} (1-\rho_2)^{k-1}.$$

Therefore,

$$\tfrac{\log H_k^{(i)}}{k} \geq \tfrac{1}{k} \sum_{l=1}^{a_2^{M(i,j)}(i,j)} \log \left(\prod_{n=1}^{k-1} \xi_n^{(l)}(i) \right) + \tfrac{1}{k} \sum_{l=b_2^{N(i,j)}(i,j)}^{L} \log \left(\prod_{n=1}^{k-1} \xi_n^{(l)}(i) \right) + \tfrac{1}{k} \log \left(\rho_1{}^{s-1} \rho_2{}^{L-s} \right) +$$
$$\tfrac{1}{k} \log \left(\sum_{n_s=0}^{k} \prod_{n=1}^{n_s-1} \tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)} \prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \xi_n^{(l)}(i) \right) + \tfrac{1}{k} \log \left((1-\rho_1)^{k-1} \right) + \tfrac{1}{k} \log \left((1-\rho_2)^{k-1} \right).$$

Since L is a finite integer, we have

$$\tfrac{1}{k} \log \left(\rho_1{}^{s-1} \rho_2{}^{L-s} \right) \xrightarrow[k\to\infty]{} 0. \tag{B.23}$$

By Lemma B.2 and the definition of η_l in (3.25), we can see that

$$\tfrac{1}{k}\log\left(\sum_{n_s=0}^{k}\prod_{n=1}^{n_s-1}\tfrac{(1-\rho)}{(1-\rho_1)(1-\rho_2)}\prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\xi_n^{(l)}(i)\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1}\eta_l(i,j)\right)_+ . \tag{B.24}$$

In addition, by the definition of η_l and Algorithm 1, we can see that

$$\begin{aligned}&\tfrac{1}{k}\sum_{l=1}^{a_2^{M(i,j)}(i,j)}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)+\tfrac{1}{k}\log\left((1-\rho_1)^{k-1}\right)+\tfrac{1}{k}\sum_{l=b_2^{N(i,j)}(i,j)}^{L}\log\left(\prod_{n=1}^{k-1}\xi_n^{(l)}(i)\right)\\&+\tfrac{1}{k}\log\left((1-\rho_2)^{k-1}\right)\xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}\sum_{l=1}^{a_2^{M(i,j)}(i,j)}\eta_l(i,j)+\sum_{l=b_2^{N(i,j)}(i,j)}^{L}\eta_l(i,j).\end{aligned} \tag{B.25}$$

Combining (B.23), (B.24) and (B.25), (B.22) is proved. Next, we need to prove the other direction, i.e., for any $i,j\in\mathcal{I}$,

$$\liminf_{k\to\infty}\frac{1}{k}\log H_k^{(i)}\leq h(i,j) \tag{B.26}$$

$\mathbb{P}_j$ almost surely.

Applying (B.16) recursively, we have

$$H_k^{(i)}\leq\sum_{n_l=0}^{k}\left(\prod_{n=1}^{n_l-1}\left(\frac{(1-\rho)\xi_k^{(l)}(i)}{(1-\rho_1)(1-\rho_2)}\right)\right)\cdot\tilde{\psi}_{l-1}^{(i)}(k,n_l)\tilde{\phi}_{l+1}^{(i)}(k,n_l). \tag{B.27}$$

Here $\tilde{\phi}_{l+1}^{(i)}(k,n_l)$ and $\tilde{\psi}_{l-1}^{(i)}(k,n_l)$ are given (B.18). We apply the first inequality in (B.19) to $\tilde{\psi}_{l+1}^{(i)}(k,n_l)$ for $l=a_2^m(i,j)$ and $1\leq m\leq M(i,j)$, following the order from $m=1$ to $m=M(i,j)$. Then we also apply the second inequality in (B.19) to $\tilde{\phi}_{l+1}^{(i)}(k,n_l)$ for $l=b_2^n(i,j)$

and $1 \leq n \leq N(i,j)$, following the order from $n = 1$ to $n = N(i,j)$. We define

$$\begin{cases} \Omega_m = \sum\limits_{n_{a_2^m(i,j)}=0}^{k+1} \left[\left(\prod\limits_{n=1}^{n_{a_2^m(i,j)}-1} \xi_n^{(a_2^m(i,j))}(i) \right) \zeta_{m,a_2^m(i,j)-1}^{(i)}(n_{a_2^m(i,j)}) \right], 1 \leq m \leq M(i,j), \\ \Theta_m = \sum\limits_{n_{b_2^n(i,j)}=0}^{k+1} \left[\left(\prod\limits_{m=1}^{n_{b_2^n(i,j)}-1} \xi_m^{(b_2^n(i,j))}(i) \right) \varepsilon_{n,b_2^n(i,j)+1}^{(i)}(n_{b_2^n(i,j)}) \right], N(i,j) \geq n \geq 1, \\ \Gamma = \sum\limits_{n_s=0}^{k+1} \left[\left(\prod\limits_{n=1}^{n_s-1} \xi_n^{(s)}(i) \right) \zeta_{M(i,j)+1,s-1}^{(i)}(n_s) \varepsilon_{N(i,j)+1,s+1}^{(i)}(n_s) \right] \end{cases} \tag{B.28}$$

where

$$\begin{cases} \zeta_{m,t}^{(i)}(n_{t+1}) = \sum\limits_{n_t=n_{t+1}}^{k+1} \left[\left(\prod\limits_{n=1}^{n_t-1} \xi_n^{(t)}(i) \right) \zeta_{m,t-1}^{(i)}(n_t) \right], a_2^m(i,j) - 1 \geq t \geq a_1^m(i,j), \\ \qquad\qquad\qquad\qquad\qquad\qquad 1 \leq m \leq M(i,j) + 1, \\ \zeta_{m,t}^{(i)}(n_{t+1}) = 1, t = a_1^m(i,j) - 1, \\ \varepsilon_{n,t}^{(i)}(n_{t-1}) = \sum\limits_{n_t=n_{t-1}}^{k+1} \left[\left(\prod\limits_{m=1}^{n_t-1} \xi_m^{(t)}(i) \right) \varepsilon_{n,t+1}^{(i)}(n_t) \right], b_1^n(i,j) \geq t \geq b_2^n(i,j) + 1, \\ \qquad\qquad\qquad\qquad\qquad\qquad N(i,j) + 1 \geq n \geq 1. \end{cases} \tag{B.29}$$

With the definitions in (B.28), we have

$$H_k^{(i)} \leq \left(\prod_{m=1}^{M(i,j)} \Omega_m \right) \left(\prod_{n=1}^{N(i,j)} \Theta_n \right) \Gamma. \tag{B.30}$$

In (B.28), we denote

$$\begin{cases} a_1^{M(i,j)+1}(i,j) = a_2^{M(i,j)}(i,j) + 1, \\ a_2^{M(i,j)+1}(i,j) = s - 1, \\ b_1^{N(i,j)+1}(i,j) = b_2^{N(i,j)}(i,j) - 1, \\ b_2^{N(i,j)+1}(i,j) = s + 1 \end{cases} . \tag{B.31}$$

Now, it suffices to show that

$$\begin{cases} \limsup\limits_{k\to\infty} \frac{1}{k}\log\Omega_m = \sum\limits_{l=a_1^m(i,j)}^{a_2^m(i,j)} \eta_l(i,j), 1 \le m \le M(i,j) \\ \limsup\limits_{k\to\infty} \frac{1}{k}\log\Theta_m = \sum\limits_{l=b_2^n(i,j)}^{b_1^n(i,j)} \eta_l(i,j), 1 \le n \le N(i,j) \\ \limsup\limits_{k\to\infty} \frac{1}{k}\log\Gamma = \sum\limits_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \eta_l(i,j) \end{cases} \tag{B.32}$$

$\mathbb{P}_j$ almost surely. The proof of the three inequalities are similar, and the third one is more complicated. So here we only provide the proof of the third one. For any $1 \le l \le L$,

$$\begin{aligned} & \sum_{n_l=n_{l-1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \left(\sum_{n_{l+1}=n_l}^{k+1} \prod_{m_{l+1}=1}^{n_{l+1}-1} \xi_{m_{l+1}}^{(l+1)} \right) \\ & = \sum_{n_l=n_{l-1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \xi_{m_l}^{(l+1)} \left(\sum_{n_{l+1}=n_l}^{k+1} \prod_{m_{l+1}=n_l}^{n_{l+1}-1} \xi_{m_{l+1}}^{(l+1)} \right) \\ & \le \sum_{n_l=n_{l-1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \xi_{m_l}^{(l+1)} \left(\sum_{n_{l+1}=n_l}^{k+n_l+1} \prod_{m_{l+1}=n_l}^{n_{l+1}-1} \xi_{m_{l+1}}^{(l+1)} \right) \\ & \le \sum_{n_l=n_{l-1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \xi_{m_l}^{(l+1)} \left(\max_{n_l \le k+1} \sum_{n_{l+1}=n_l}^{k+n_l+1} \prod_{m_{l+1}=n_l}^{n_{l+1}-1} \xi_{m_{l+1}}^{(l+1)} \right). \end{aligned}$$

Similarly, we have

$$\sum_{n_l=n_{l+1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \left(\sum_{n_{l-1}=n_l}^{k+1} \prod_{m_{l-1}=1}^{n_{l-1}-1} \xi_{m_{l-1}}^{(l-1)} \right) \le \sum_{n_l=n_{l+1}}^{k+1} \prod_{m_l=1}^{n_l-1} \xi_{m_l}^{(l)} \xi_{m_l}^{(l-1)} \left(\max_{n_l \le k+1} \sum_{n_{l-1}=n_l}^{k+n_l+1} \prod_{m_{l-1}=n_l}^{n_{l-1}-1} \xi_{m_{l-1}}^{(l-1)} \right). \tag{B.33}$$

For Γ, apply (B.33) from $l = a_2^{M(i,j)}(i,j) + 2$ to $l = s$, then apply (B.2) from to $l = b_2^{N(i,j)}(i,j) - 2$ to $l = s$, we have

$$\Gamma \le \sum_{n_s=0}^{k+1} \left(\prod_{q=1}^{n_s-1} \prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \xi_q^{(l)}(i) \right) \prod_{l=a_2^{M(i,j)}(i,j)+1}^{s} C_l \prod_{l=s}^{b_2^{N(i,j)}(i,j)-1} D_l, \tag{B.34}$$

where

$$\begin{cases} C_l = \max\limits_{n_l \le k+1} \left(1 + \sum\limits_{n_{l-1}=n_l}^{k+n_l} \prod\limits_{m_{l-1}=n_l}^{n_{l-1}} \prod\limits_{t=a_2^{M(i,j)}(i,j)}^{l} \xi^{(t)}_{m_{l-1}}\right), \\ D_l = \max\limits_{n_l \le k+1} \left(1 + \sum\limits_{n_{l+1}=n_l}^{k+n_l} \prod\limits_{m_{l+1}=n_l}^{n_{l+1}} \prod\limits_{t=l}^{b_2^{N(i,j)}(i,j)} \xi^{(t)}_{m_{l+1}}\right). \end{cases}$$

By Lemma B.2, for $s \le l \le b_2^{N(i,j)}(i,j) - 1$, we have

$$\frac{1}{k}\log\left(1 + \sum_{n_{l+1}=n_l}^{k+n_l} \prod_{m_{l+1}=n_l}^{n_{l+1}} \prod_{t=l}^{b_2^{N(i,j)}(i,j)} \xi^{(l)}_{m_{l+1}}\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} \left(\sum_{t=l}^{b_2^{N(i,j)}(i,j)} \eta_t(i,j)\right)_+ = 0.$$

And for $a_2^{M(i,j)}(i,j) + 1 \le l \le s$, we have

$$\frac{1}{k}\log\left(1 + \sum_{n_{l-1}=n_l}^{k+n_l} \prod_{m_{l-1}=n_l}^{n_{l-1}} \prod_{t=a_2^{M(i,j)}(i,j)}^{l} \xi^{(t)}_{m_{l-1}}\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} \left(\sum_{t=a_2^{M(i,j)}(i,j)}^{l} \eta_t(i,j)\right)_+ = 0.$$

Therefore, we have

$$\begin{cases} \frac{1}{k}\log C_l \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} 0, \\ \frac{1}{k}\log D_l \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} 0. \end{cases}$$

Similarly, by lemma B.2, we can see that,

$$\sum_{n_s=0}^{k+1}\left(\prod_{q=1}^{n_s-1} \prod_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \xi_q^{(l)}(i)\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} \left(\sum_{l=a_2^{M(i,j)}(i,j)+1}^{b_2^{N(i,j)}(i,j)-1} \eta_l(i,j)\right)_+ . \tag{B.35}$$

By (B.34), (B.2) and (B.35), we know that the third inequality in (B.32) is true. Using similar steps, we can prove the other two inequalities in (B.32). Hence (B.22) is proved. Finally, by (B.22) and (B.26), the proof of Proposition 3.3 is complete.

B.3 Proof of Proposition 3.4

we first prove that, for any $i, j \in \mathcal{I}$,

$$\liminf_{k\to\infty} \frac{1}{k} \log H_k^{(i)} \geq h(i,j) \tag{B.36}$$

$\mathbb{P}_j$ almost surely. Please note that the $h(i,j)$ in this section is defined in Proposition 3.4 since we are studying the 2D case.

Then, we can see that

$$\psi_{r+1}^{(i)}\left(k, n_r, S_1, S_2\right) \geq \rho_1 \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \psi_{r+2}^{(i)}\left(k, n_r, S_1, S_2\right), R(S_1,S_2) > r \geq 0\ . \tag{B.37}$$

Applying equation (B.37) repeatedly, we have

$$\begin{aligned}
&H_k^{(i)} \geq \sum_{S_1=1}^{H} \sum_{S_1=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} \tfrac{(1-\rho)f_0(x_{n,S_1,S_2})}{(1-\rho_1)f_i(x_{n,S_1,S_2})} \cdot \\
&\left[\rho_1^{R(S_1,S_2)-1} \left(\prod_{(a,b)\in\mathcal{O}(S_1,S_2,0)} \prod_{n=1}^{n_0-1} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) (1-\rho_1)^{n_0-1} \right] \\
&= \left(\sum_{S_1=1}^{H} \sum_{S_1=1}^{W} \kappa_{S_1,S_2} \rho_1^{R(S_1,S_2)-1} \right) \left(\sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} (1-\rho) \prod_{a=1}^{H} \prod_{b=1}^{W} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right).
\end{aligned} \tag{B.38}$$

Since $R(S_1, S_2)$ is finite for $1 \leq S_1 \leq H, 1 \leq S_2 \leq W$, we have

$$\frac{1}{k} \log \left(\sum_{S_1=1}^{H} \sum_{S_1=1}^{W} \kappa_{S_1,S_2} \rho_1^{R(S_1,S_2)-1} \right) \xrightarrow[k\to\infty]{} 0. \tag{B.39}$$

Since the change will happen at all sensors at an almost surely finite time T, then by applying Lemma B.2, we have

$$\begin{aligned}
&\tfrac{1}{k} \log \left(\sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} (1-\rho) \prod_{a=1}^{H} \prod_{b=1}^{W} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) = \tfrac{1}{k} \log \left(2 + \sum_{n_0=2}^{k} \prod_{n=1}^{n_0-1} (1-\rho) \prod_{a=1}^{H} \prod_{b=1}^{W} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) \\
&\xrightarrow[k\to\infty]{\mathbb{P}_j - a.s.} \left(\log(1-\rho) + HWq(j,i) - HWq(j,0)\right)_+.
\end{aligned} \tag{B.40}$$

Combining (B.38), (B.39) and (B.40), we can see that (B.36) is proved. Next we need to prove the other direction, i.e., for any $i, j \in \mathcal{I}$,

$$\limsup_{k\to\infty} \frac{1}{k} \log H_k^{(i)} \leq h(i,j). \tag{B.41}$$

For any integer $n_x \geq 0$, we can see that

$$\begin{aligned}
&\prod_{n=1}^{k} (1-\rho_1) \prod_{(a,b)\in\mathcal{O}(S_1,S_2,r)} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \\
&= \left(\prod_{n=1}^{k} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) \left(\prod_{n=1}^{k} (1-\rho_1) \prod_{(a,b)\in\mathcal{O}(S_1,S_2,r+1)} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) \\
&\leq \left(\prod_{n=1}^{k} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \tfrac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \right) \psi_{r+2}^{(i)} \left(k, n_x, S_1, S_2\right).
\end{aligned} \tag{B.42}$$

From (3.33), using (B.42) with $n_x = k+1$ and the fact that $\rho_1 < 1$, we can see that

$$\begin{aligned}
\psi_{r+1}^{(i)} \left(k, n_r, S_1, S_2\right) \leq \sum_{n_{r+1}=n_r}^{k+1} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_r-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \psi_{r+2}^{(i)} \left(k, n_{r+1}, S_1, S_2\right), \\
R(S_1,S_2) > r \geq 0.
\end{aligned} \tag{B.43}$$

Applying these two inequalities in (B.43) recursively, we have

$$H_k^{(i)} \leq \sum_{S_1=1}^{H} \sum_{S_1=1}^{W} \kappa_{S_1,S_2} \sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} \frac{(1-\rho)f_0(x_{n,S_1,S_2})}{(1-\rho_1)f_i(x_{n,S_1,S_2})} \cdot \tilde{\psi}_1^{(i)}(k, n_0, S_1, S_2) \tag{B.44}$$

where

$$\begin{cases}
\tilde{\psi}_{r+1}^{(i)} \left(k, n_r, S_1, S_2\right) = \sum_{n_{r+1}=n_r}^{k+1} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_{r+1}-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \tilde{\psi}_{r+2}^{(i)} \left(k, n_{r+1}, S_1, S_2\right), \\
\qquad R(S_1,S_2) > r \geq 0 \\
\tilde{\psi}_{R(S_1,S_2)+1}^{(i)} \left(k, n_r, S_1, S_2\right) = (1-\rho_1)^{n_{R(S_1,S_2)}-1}.
\end{cases} \tag{B.45}$$

Since n_0 in (B.44) is no larger than $n_{R(S_1,S_2)}$ in (B.45), so the right hand side of (B.44) will become larger if we cancel all $(1-\rho_1)$ in (B.44).

Furthermore, we know that

$$\tilde{\psi}^{(i)}_{r+1}\left(k, n_r, S_1, S_2\right) \le \sum_{n_{r+1}=0}^{k+1} \prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{n_{r+1}-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})} \tilde{\psi}^{(i)}_{r+2}\left(k, n_{r+1}, S_1, S_2\right), \tag{B.46}$$

$$R(S_1, S_2) > r \ge 0.$$

By canceling all $(1-\rho_1)$ in (B.44) and applying (B.46) reversely, we have that $H^{(i)}_k \le \sum\limits_{S_1=1}^{H} \sum\limits_{S_2=1}^{W} \kappa_{a,b}\gamma_{S_1,S_2}$ where

$$\gamma_{S_1,S_2} = \left(\sum_{n_0=0}^{k} \prod_{n=1}^{n_0-1} \frac{(1-\rho) f_0(x_{n,S_1,S_2})}{f_i(x_{n,S_1,S_2})}\right) \cdot \left(\prod_{r=1}^{R(S_1,S_2)} \left(\sum_{t=0}^{k+1} \left(\prod_{(a,b)\in\mathcal{C}(S_1,S_2,r+1)} \prod_{n=1}^{t-1} \frac{f_0(x_{n,a,b})}{f_i(x_{n,a,b})}\right)\right)\right).$$

By Lemma B.2, for any $1 \le S_1 \le H$ and $1 \le S_2 \le W$, we have

$$\frac{1}{k}\log\left(\gamma_{S_1,S_2}\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.} (HW-1)\left(q(j,i)-q(j,0)\right)_+ + \left(\log(1-\rho)+q(j,i)-q(j,0)\right)_+.$$

Since $\kappa_{a,b} \ge 0$ and $\sum\limits_{1\le a\le H, 1\le b\le W} \kappa_{a,b} = 1$, we have

$$\min\left(\log\left(\tfrac{\gamma_{1,1}}{k}\right), \log\left(\tfrac{\gamma_{1,2}}{k}\right), \ldots, \log\left(\tfrac{\gamma_{H,W}}{k}\right)\right) \le \frac{\log\left(\sum\limits_{S_1=1}^{H}\sum\limits_{S_2=1}^{W} \kappa_{S_1,S_2}\gamma_{S_1,S_2}\right)}{k}$$
$$\le \max\left(\log\left(\tfrac{\gamma_{1,1}}{k}\right), \log\left(\tfrac{\gamma_{1,2}}{k}\right), \ldots, \log\left(\tfrac{\gamma_{H,W}}{k}\right)\right).$$

We can have

$$\frac{1}{k}\log\left(\sum_{S_1=1}^{H}\sum_{S_2=1}^{W} \kappa_{S_1,S_2}\gamma_{S_1,S_2}\right) \xrightarrow[k\to\infty]{\mathbb{P}_j-a.s.}$$
$$(HW-1)\left(q(j,i)-q(j,0)\right)_+ + \left(\log(1-\rho)+q(j,i)-q(j,0)\right)_+.$$

When Condition 1 is satisfied, we have

$$
\begin{aligned}
&(\log(1-\rho) + HWq(j,i) - HWq(j,0))_+ = (\log(1-\rho) + \\
&q(j,i) - q(j,0))_+ + (HW - 1)\left(q(j,i) - q(j,0)\right)_+ .
\end{aligned}
\tag{B.47}
$$

Hence (B.41) is proved. Therefore, Proposition 3.4 is true.

Bibliography

[1] G. B. Wetherill and D. W. Brown. *Statistical process control: theory and practice*, volume 199. Chapman and Hall London, 1991.

[2] E. Järpe. Surveillance of the interaction parameter of the Ising model. *Communications in Statistics-Theory and Methods*, 28(12):3009–3027, Jun. 1999.

[3] D. Jarušková. Some problems with application of change-point detection methods to environmental data. *Environmetrics*, 8(5):469–483, Dec. 1998.

[4] M. Pettersson. Monitoring a freshwater fish population: Statistical surveillance of biodiversity. *Environmetrics: The official journal of the International Environmetrics Society*, 9(2):139–150, Dec. 1998.

[5] C. Sonesson and D. Bock. A review and discussion of prospective statistical surveillance in public health. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 166(1):5–21, Jan. 2003.

[6] S. Trivedi and R. Chandramouli. Secret key estimation in sequential steganography. *IEEE Transactions on Signal Processing*, 53(2):746–757, Feb. 2005.

[7] M. Basseville. Edge detection using sequential methods for change in level–Part II: Sequential detection of change in mean. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 29(1):32–50, Feb. 1981.

[8] E. Andreou and E. Ghysels. The impact of sampling frequency and volatility estimators on change-point tests. *Journal of Financial Econometrics*, 2(2):290–318, Mar. 2004.

[9] M. Beibel and H. R. Lerche. A new look at optimal stopping problems related to mathematical finance. *Statistica Sinica*, 7(1):93–108, Jan. 1997.

[10] A. N. Shiryaev. Quickest detection problems in the technical analysis of the financial data. In *Mathematical finance—Bachelier congress 2000*, pages 487–521. Springer, 2002.

[11] G. Rovatsos, X. Jiang, A. D. Domínguez-García, and V. V. Veeravalli. Statistical power system line outage detection under transient dynamics. *IEEE Transactions on Signal Processing*, 65(11):2787–2797, Feb. 2017.

[12] Y. C. Chen, T. Banerjee, A. D. Domínguez-García, and V. V. Veeravalli. Quickest line outage detection and identification. *IEEE Transactions on Power Systems*, 31(1):749–758, Feb. 2015.

[13] M. Petzold, C. Sonesson, E. Bergman, and H. Kieler. Surveillance in longitudinal models: detection of intrauterine growth restriction. *Biometrics*, 60(4):1025–1033, Dec. 2004.

[14] M. C. Yang, Y. Y. Namgung, R. G. Marks, I. Magnusson, and W. B. Clark. Change detection on longitudinal data in periodontal research. *Journal of Periodontal Research*, 28(2):152–160, Mar. 1993.

[15] T. Friede, F. Miller, W. Bischoff, and M. Kieser. A note on change point estimation in dose–response trials. *Computational Statistics & Data Analysis*, 37(2):219–232, Aug. 2001.

[16] M. Frisén. Evaluations of methods for statistical surveillance. *Statistics in Medicine*, 11(11):1489–1502, 1992.

[17] P. K. Enge. The global positioning system: signals, measurements, and performance. *International Journal of Wireless Information Networks*, 1(2):83–105, 1994.

[18] I. Nikiforov, V. Varavva, and V. Kireichikov. Application of statistical fault detection algorithms to navigation systems monitoring. *Automatica*, 29(5):1275–1290, Sept. 1993.

[19] A. A. Cardenas, J. S. Baras, and V. Ramezani. Distributed change detection for worms, DDoS and other network attacks. In *Proc. American Control Conference*, volume 2, pages 1008–1013. Boston, MA, Jun. 2004.

[20] R. K. Chang. Defending against flooding-based distributed denial-of-service attacks: A tutorial. *IEEE Communications Magazine*, 40(10):42–51, Dec. 2002.

[21] H. Kim, B. L. Rozovskii, and A. G. Tartakovsky. A nonparametric multichart CUSUM test for rapid detection of DOS attacks in computer networks. *International Journal of Computing & Information Sciences*, 2(3):149–158, Dec. 2004.

[22] P. Papantoni-Kazakos. Algorithms for monitoring changes in quality of communication links. *IEEE Transactions on Communications*, 27(4):682–693, Apr. 1979.

[23] A. G. Tartakovsky, B. L. Rozovskii, R. B. Blazek, and H. Kim. A novel approach to detection of intrusions in computer networks via adaptive sequential and batch-sequential change-point detection methods. *IEEE Transactions on Signal Processing*, 54(9):3372–3382, Aug. 2006.

[24] M. Thottan and C. Ji. Anomaly detection in IP networks. *IEEE Transactions on Signal Processing*, 51(8):2191–2204, Jul. 2003.

[25] V. F. Pisarenko, A. F. Kushnir, and I. V. Savin. Statistical adaptive algorithms for estimation of onset moments of seismic phases. *Physics of the Earth and Planetary Interiors*, 47:4–10, Aug.1987.

[26] M. Marcus and P. Swerling. Sequential detection in radar with multiple resolution elements. *IRE Transactions on Information Theory*, 8(3):237–245, Apr. 1962.

[27] C. Han, P. K. Willett, and D. A. Abraham. Some methods to evaluate the performance of Page's test as used to detect transient signals. *IEEE Transactions on Signal Processing*, 47(8):2112–2127, Aug. 1999.

[28] D. Lelescu and D. Schonfeld. Statistical sequential analysis for real-time video scene change detection on compressed multimedia bitstream. *IEEE Transactions on Multimedia*, 5(1):106–117, Apr. 2003.

[29] A. N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Probability & Its Applications*, 8(1):22–46, 1963.

[30] G. Lorden et al. Procedures for reacting to a change in distribution. *The Annals of Mathematical Statistics*, 42(6):1897–1908, Dec. 1971.

[31] J. Geng and L. Lai. Quickest change-point detection over multiple data streams via sequential observations. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4404–4408. Calgary, Canada, Apr. 2018.

[32] H. Chen. Sequential change-point detection based on nearest neighbors. *The Annals of Statistics*, 47(3):1381–1407, Jun. 2019.

[33] L. Chu and H. Chen. Sequential change-point detection for high-dimensional and non-euclidean data. *IEEE Transactions on Signal Processing*, 70:4498–4511, Sept. 2022.

[34] J. Geng and L. Lai. Bayesian quickest detection with unknown post-change parameter. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4169–4173. Shanghai, China, 2016.

[35] J. Geng, B. Zhang, L. M. Huie, and L. Lai. Online change-point detection of linear regression models. *IEEE Transactions on Signal Processing*, 67(12):3316–3329, May. 2019.

[36] G. Rovatsos, S. Zou, and V. V. Veeravalli. Quickest change detection under transient dynamics. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4785–4789. New Orleans, LA, Mar. 2017.

[37] S. Zou, G. Fellouris, and V. V. Veeravalli. Quickest change detection under transient dynamics: theory and asymptotic analysis. *IEEE Transactions on Information Theory*, 65(3):1397–1412, Oct. 2018.

[38] T. Banerjee and V. V. Veeravalli. Data-efficient quickest change detection with on–off observation control. *Sequential Analysis*, 31(1):40–77, Feb. 2012.

[39] T. Banerjee and V. V. Veeravalli. Data-efficient quickest change detection in minimax settings. *IEEE Transactions on Information Theory*, 59(10):6917–6931, Jul. 2013.

[40] T. Banerjee and V. V. Veeravalli. Data-efficient quickest change detection in sensor networks. *IEEE Transactions on Signal Processing*, 63(14):3727–3735, May. 2015.

[41] R. Zhang, Y. Xie, R. Yao, and F. Qiu. Online detection of cascading change-points using diffusion networks. In *Proc. Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1–6. Monticello, IL, Sept. 2022.

[42] G. Rovatsos, G. V. Moustakides, and V. V. Veeravalli. Quickest detection of moving anomalies in sensor networks. *IEEE Journal on Selected Areas in Information Theory*, 2(2):762–773, Apr. 2021.

[43] D. Li, L. Lai, and S. Cui. Quickest change detection and identification across a sensor array. In *Proc. IEEE Global Conference on Signal and Information Processing*, pages 145–148. Austin, TX, Dec. 2013.

[44] V. V. Veeravalli. Decentralized quickest change detection. *IEEE Transactions on Information Theory*, 47(4):1657–1665, May. 2001.

[45] A. G. Tartakovsky and V. V. Veeravalli. Asymptotically optimal quickest change detection in distributed sensor systems. *Sequential Analysis*, 27(4):441–475, Oct. 2008.

[46] V. Raghavan and V. V. Veeravalli. Quickest change detection of a Markov process across a sensor array. *IEEE Transactions on Information Theory*, 56(4):1961–1981, Mar. 2010.

[47] S. Aminikhanghahi and D. J. Cook. A survey of methods for time series change point detection. *Knowledge and Information Systems*, 51(2):339–367, May. 2017.

[48] T. Banerjee, H. Firouzi, and A. O. Hero. Quickest detection for changes in maximal KNN coherence of random matrices. *IEEE Transactions on Signal Processing*, 66(17):4490–4503, Jul. 2018.

[49] Y. Xie and D. Siegmund. Sequential multi-sensor change-point detection. In *Proc. on IEEE Information Theory and Applications Workshop*, pages 1–20. San Diego, CA, May 2013.

[50] H. V. Poor and O. Hadjiliadis. *Quickest detection*. Cambridge University Press, 2008.

[51] Y. Wang and Y. Mei. Large-scale multi-stream quickest change detection via shrinkage post-change estimation. *IEEE Transactions on Information Theory*, 61(12):6926–6938, Oct. 2015.

[52] O. Hadjiliadis, H. Zhang, and H. V. Poor. One shot schemes for decentralized quickest change detection. *IEEE Transactions on Information Theory*, 55(7):3346–3359, Jun. 2009.

[53] R. Jana and S. Dey. Change detection in teletraffic models. *IEEE Transactions on Signal Processing*, 48(3):846–853, Mar. 2000.

[54] C. D. Fuh and Y. Mei. Quickest change detection and Kullback-Leibler divergence for two-state hidden Markov models. *IEEE Transactions on Signal Processing*, 63(18):4866–4878, Jun. 2015.

[55] A. Jain, P. Sarvepalli, S. Bhashyam, and A. P. Kannu. Algorithms for change detection with sparse signals. *IEEE Transactions on Signal Processing*, 68:1331–1345, Feb. 2020.

[56] E. Nitzan, T. Halme, and V. Koivunen. Bayesian methods for multiple change-point detection with reduced communication. *IEEE Transactions on Signal Processing*, 68:4871–4886, Aug. 2020.

[57] Y. Xie, J. Huang, and R. Willett. Change-point detection for high-dimensional time series with missing data. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):12–27, Dec. 2012.

[58] C. W. Baum and V. V. Veeravalli. A sequential procedure for multihypothesis testing. *IEEE Transactions on Information Theory*, 40(6), Nov. 1994.

[59] A. Novikov. Optimal sequential multiple hypothesis tests. *Kybernetika*, 45(2):309–330, 2009.

[60] S. Nitinawarat, G. K. Atia, and V. V. Veeravalli. Controlled sensing for multihypothesis testing. *IEEE Transactions on Automatic Control*, 58(10):2451–2464, May. 2013.

[61] J. Heydari and A. Tajer. Controlled sensing for multi-hypothesis testing with co-dependent actions. In *Proc. IEEE International Symposium on Information Theory*, pages 2321–2325. Vail, CO, Jun. 2018.

[62] C. Z. Bai, V. Katewa, V. Gupta, and Y. F. Huang. A stochastic sensor selection scheme for sequential hypothesis testing with multiple sensors. *IEEE Transactions on Signal Processing*, 63(14):3687–3699, Apr. 2015.

[63] B. Liu, J. Lan, and X. R. Li. A 2-SPRT based approach to multiple-model hypothesis testing for multi-distribution detection. *IEEE Transactions on Signal Processing*, 64(12):3221–3236, Mar. 2016.

[64] I. V. Nikiforov. A generalized change detection problem. *IEEE Transactions on Information Theory*, 41(1):171–187, Jan. 1995.

[65] T. L. Lai. Sequential multiple hypothesis testing and efficient fault detection-isolation in stochastic systems. *IEEE Transactions on Information Theory*, 46(2):595–608, Mar. 2000.

[66] S. Dayanik, C. Goulding, and H. V. Poor. Bayesian sequential change diagnosis. *Mathematics of Operations Research*, 33(2):475–496, May. 2008.

[67] S. Dayanik, W. B. Powell, and K. Yamazaki. Asymptotically optimal Bayesian sequential change detection and identification rules. *Annals of Operations Research*, 208(1):337–370, Sept. 2013.

[68] D. A. Tibaduiza Burgos, R. C. Gomez Vargas, C. Pedraza, D. Agis, and F. Pozo. Damage identification in structural health monitoring: A brief review from its implementation to the use of data-driven applications. *Sensors*, 20(3):733, Jan. 2020.

[69] L. Fillatre and I. Nikiforov. Asymptotically uniformly minimax detection and isolation in network monitoring. *IEEE Transactions on Signal Processing*, 60(7):3357–3371, Apr. 2012.

[70] A. Tartakovsky, I. Nikiforov, and M. Basseville. *Sequential analysis: hypothesis testing and changepoint detection*. CRC Press, 2014.

[71] M. Kobylanski, M. C. Quenez, and E. Rouy-Mironescu. Optimal multiple stopping time problem. *The Annals of Applied Probability*, 21(4):1365–1399, Nov. 2011.

[72] X. Ma, L. Lai, and S. Cui. Optimal two-stage Bayesian sequential change diagnosis. In *Proc. IEEE International Symposium on Information Theory*, pages 1130–1135. Los Angles, CA, Jun. 2020.

[73] X. Ma, L. Lai, and S. Cui. Two-stage Bayesian sequential change diagnosis. *IEEE Transactions on Signal Processing*, 69:6131–6147, Sept. 2021.

[74] K. Mechitov, W. Kim, G. Agha, and T. Nagayama. High-frequency distributed sensing for structure monitoring. In *Proc. First Intl. Workshop on Networked Sensing Systems (INSS 04)*, pages 101–105, 2004.

[75] H. Li, C. Li, and H. Dai. Quickest spectrum sensing in cognitive radio. In *Proc. Annual Conference on Information Sciences and Systems*, pages 203–208. Princeton, NJ, Mar. 2008.

[76] L. Xie, Y. Xie, and G. V. Moustakides. Asynchronous multi-sensor change-point detection for seismic tremors. In *Proc. IEEE International Symposium on Information Theory (ISIT)*, pages 787–791. Paris, France, Sept, 2019.

[77] L. Lai. Quickest change point identification across a sensor array. In *Proc. on IEEE Military Communications Conference (MILCOM)*, pages 1–5. Orlando, FL, Jan. 2012.

[78] S. Zou, V. V. Veeravalli, J. Li, and D. Towsley. Quickest detection of dynamic events in networks. *IEEE Transactions on Information Theory*, 66(4):2280–2295, Apr. 2020.

[79] M. Ludkovski. Bayesian quickest detection in sensor arrays. *Sequential Analysis*, 31(4):481–504, Oct. 2012.

[80] X. Ma, L. Lai, and S. Cui. Bayesian two-stage sequential change diagnosis via multi-sensor array. In *Proc. International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. Golden Coast, Australia, Oct. 2021.

[81] X. Ma, L. Lai, and S. Cui. Bayesian two-stage sequence change diagnosis across a sensor array. *IEEE Transactions on Information Theory*, submitted, 2022.

[82] T. L. Lai and H. Xing. Sequential change-point detection when the pre-and post-change parameters are unknown. *Sequential Analysis*, 29(2):162–175, Apr. 2010.

[83] H. Li. Data driven quickest change detection: An algorithmic complexity approach. In *Proc. IEEE International Symposium on Information Theory*, pages 21–25. Barcelona, Spain, Jul. 2016.

[84] M. N. Kurt, Y. Yılmaz, and X. Wang. Real-time nonparametric anomaly detection in high-dimensional settings. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2463–2479, Jan. 2020.

[85] T. Flynn, O. Hadjiliadis, I. Stamos, and F. J. Vázquez-Abad. Data driven stochastic approximation for change detection. In *Proc. on IEEE Winter Simulation Conference (WSC)*, pages 2279–2290. Las Vegas, NV, Dec. 2017.

[86] Y. Cao, L. Xie, Y. Xie, and H. Xu. Sequential change-point detection via online convex optimization. *Entropy*, 20(2):108, Feb. 2018.

[87] M. N. Kurt, O. Ogundijo, C. Li, and X. Wang. Online cyber-attack detection in smart grid: A reinforcement learning approach. *IEEE Transactions on Smart Grid*, 10(5):5174–5185, Sept. 2018.

[88] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction.* 2018.

[89] X. Ma, L. Lai, and S. Cui. A deep Q-network based approach for online Bayesian change point detection. In *Proc. on IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 1–6. Golden Coast, Australia, Oct. 2021.

[90] X. Ma, L. Lai, and S. Cui. A neural Monte Carlo based approach for online Bayesian change point detection. *IEEE Transactions on Signal Processing*, submitted, 2022.

[91] S. Dayanik and C. Goulding. Sequential detection and identification of a change in the distribution of a markov-modulated random sequence. *IEEE Transactions on Information Theory*, 55(7):3323–3345, Jun. 2009.

[92] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.

[93] S. Thrun and A. Schwartz. Issues in using function approximation for reinforcement learning. In *Proceedings of the Fourth Connectionist Models Summer School*, pages 255–263. Hillsdale, NJ, Dec. 1993.

[94] H. Hasselt. Double Q-learning. *Advances in neural information processing systems*, 23:2613–2621, 2010.

[95] G. B. Huang, L. Chen, C. K. Siew, et al. Universal approximation using incremental constructive feedforward networks with random hidden nodes. *IEEE Transaction on Neural Networks*, 17(4):879–892, Jul. 2006.

[96] G. B. Huang and L. Chen. Convex incremental extreme learning machine. *Neurocomputing*, 70(16-18):3056–3062, Oct. 2007.