Statistical Cyber-Security: Change Detection in an Enterprise Computer Network

Submitted in partial fulfilment of the
requirements for the MSc in Statistics
of Imperial College London
by

Karl Hallgren

Supervisors: Dr Nick Heard and Prof Niall Adams

Department of Mathematics
Imperial College London

August 2018

Plagiarism Statement

The work contained in this report is my own work unless otherwise stated.

Karl Hallgren
September 2018

Acknowledgements

First I would like to thank my supervisors Niall Adams and Nick Heard for the precious help they have provided me throughout this project, as well as in the writing of this report. I would like to thank my parents Laurence and Michael, my brother Harry and my grandfather Yvon for their unconditional love and support. Finally, I thank Bea for her patience.

Karl Hallgren

Abstract

Cyber security is a pressing concern as organisations and individuals increasingly rely on computers. A typical enterprise computer network generates a number of high volume data sources, which may be leveraged to detect and prevent cyber attacks. Traditional systems of cyber defense focus on detecting predetermined signatures for standard attacks in the data. Statistical anomaly detection techniques aim to identify subtle malicious activities and unseen types of threats by detecting anomalous patterns in the data, which would be missed by signature-based techniques. In particular, anomaly detection may be viewed as a change detection problem since a cyber attack typically changes the behaviour of the computers it is targeting (Morgan et al., 2016).

In this report, we present a statistical method to detect changes in the behaviour of computers within a network from multiple sources, such that the context each each change may be inferred. By considering data from the enterprise network of Los Alamos National Laboratory (LANL), we argue that various important aspects of a computer network may be modelled by discrete time multivariate counting processes, whose joint distribution undergoes changes when an attack occurs. We consider a Bayesian model based binary marked change point analysis framework introduced in Bolton and Heard (2018) to detect the change points and infer how they affect the multiple processes. The method is tested on synthetic data and in particular we propose an experiment to determine its detection threshold. Data from LANL, which represent both the internal activity and the communication activity of a computer in the network, are modified to simulate evidence for a WannaCry ransomware attack. The method is shown to successfully detect the attack.

Contents

1 Introduction **1**

2 Data from an Enterprise Computer Network **4**

2.1 Network flow data 4

2.2 Windows host log data 5

2.3 Motivation for the model building process 5

3 Background to Bayesian Model-based Change Point Analysis **9**

3.1 Change point model 9

3.2 Bayesian framework 10

3.3 Inference with simulation 11

3.3.1 Metropolis Hastings algorithm 11

3.3.2 Reversible Jump MCMC 11

4 Change Point Analysis for Discrete Time Multivariate Counting Processes **13**

4.1 Bayesian change point model 13

4.1.1 The likelihood function 13

4.1.2 Prior distribution 15

4.1.3 Marginal posterior distribution of the change points 15

4.2 RJMCMC sampling 16

4.2.1 The procedure 16

4.2.2 The acceptance probabilities 17

4.3 Simulation study 17

5 Extension to Binary Marked Change Point Analysis **20**

5.1 Binary marked change point model 20

5.1.1 Binary marked change points 20

5.1.2 Prior distribution 21

5.1.3 Marginal posterior distribution of the binary marked change points . . . 23
5.2 RJMCMC sampling . . . 23
5.2.1 The procedure . . . 23
5.2.2 The acceptance probabilities . . . 24
5.3 Simulation studies . . . 26
5.3.1 Estimation of binary marked change points . . . 26
5.3.2 Detection threshold . . . 30

6 Application to the LANL data **33**
6.1 Monitoring the behaviour of a computer in LANL network . . . 33
6.2 Prior sensitivity analysis . . . 36
6.3 Detection of a WannaCry ransomware attack . . . 36

7 Conclusion **39**

A Appendix **41**
A.1 Trace plots for the RJMCMC of sub-Section 5.3.1 . . . 41
A.2 Trace plots for the RJMCMC of Section 6.1 . . . 42
A.3 Trace plots for the RJMCMC of Section 6.3 . . . 43
A.4 Kullback-Leibler divergence . . . 44

Chapter 1

Introduction

Cyber security is a pressing concern. As organisations and individuals increasingly rely on computers, the threat of cyber attacks is becoming more important. The cost to businesses of cyber crime over the next five years is estimated to reach $8 trillion (World Economic Forum, 2018). In May 2017, the WannaCry ransomware attack affected more than 200,000 computers across 150 countries and prominent organisations such as the NHS were destabilised (CERT-EU, 2017).

When performing a cyber attack on an enterprise computer network, attackers leave evidence for their malicious actions in a number of data sources generated by the computer network. Examples of such data sources are given in Morgan et al. (2016) and include records of connections established between computers within the network. Traditional systems of cyber defense focus on detecting predetermined signatures for standard attacks in the data, such as known suspicious IP addresses. As pointed out in Morgan et al. (2016), signature-based techniques fail to detect zero-day attacks and variations of known attacks. Statistical anomaly detection methods aim to identify subtle malicious activities and unseen types of threats by detecting anomalous patterns in the data, which would be missed by signature-based techniques.

Although any evidence in the data for an attack may be coined a posteriori as an anomaly, the definition of an anomaly a priori is made problematic by the difficulty of characterising the normal behaviour of computer networks, which are constantly evolving and changing in nature. Patcha and Park (2007) provides a survey of anomaly detection systems in the context of cyber security and identifies additional challenges: any detection method should be scalable for realistic deployment across an entire network and anomalies should be detected as quickly as possible whilst maintaining the number of false alerts low. Moreover, Morgan et al. (2016) emphasises the importance of providing an explanation for the anomaly: cyber analysts need situational awareness to investigate an alert.

In this context, various statistical anomaly detection techniques have been proposed. For example, scan statistics have been considered to detect unusual network traversals using legitimate credentials (Neil et al., 2013), outlying connectivity behaviours within a network are detected by measuring deviations from a Bayesian nonparametric model in Heard and Rubin-Delanchy (2016) and Denial of Service attacks are detected by monitoring changes in the number of connections made to an IP address in Wang et al. (2004). In particular, as it is the case in this report, anomaly detection may be viewed as a change detection problem since a cyber

attack typically changes the behaviour of the computers it is targeting (Morgan et al., 2016). For example, CERT-EU (2017) indicates that the WannaCry ransomware attack exploited a vulnerability in port 445 of machines running the Microsoft Windows operating system, such that the analysis of network traffic within an enterprise network could have revealed a spike of activity on port 445.

In this report, we present a statistical framework to monitor changes in the behaviour of computers within a network. In Chapter 2, we illustrate, via the data derived from the network environment at Los Alamos National Laboratory (LANL), how various important aspects of the behaviour of computers within a network may be monitored through multivariate count statistics, which may be reasonably assumed to result from discrete time multivariate counting processes. Our general approach consists of detecting significant temporal changes in the joint distribution of these summary statistics, which represent different facets of the behaviour of computers within a network.

Change detection is a broad field. It is concerned with identifying changes in a process, which occurs at unknown points in time. Existing methods may be categorised in different ways. The first divide opposes classical to Bayesian approaches. An example of a classical approach is the CUSUM chart procedure by Page (1954), whilst Shiryaev (1963) is an example of an approach where the change points are treated as random with a prior distribution. Some techniques assume the number of change points is known, others do not. The field may also be divided between online and offline detection techniques. For online detection techniques, the inference is made sequentially as new observations are made available. In contrast, for offline detection techniques, the inference is made by restrospective study.

In this report, we consider an offline Bayesian model-based change point analysis framework, which corresponds to the framework described in Green (1995) and Bolton (2016). It has a number of advantages with regards to our purpose. Firstly, the number of change points is not assumed to be known a priori and it can be extended to a multivariate process. Moreover, Bolton and Heard (2018) provide an extension, which allows each change point to differently affect the multiple processes: A binary marked vector is associated to each change point to indicate which processes are affected. In the report, we use this extension to build a statistical framework to detect changes in the behaviour of a computer network from multiple sources, such that the context of each change may be inferred. Rapid understanding of the context of anomalous behaviours by cyber security analysts, who are typically confronted with a large number of alerts, is key to distinguish attacks from false positives.

The report is structured as follows. Chapter 2 presents data collected from the network environment at LANL to demonstrate that in the context of cyber security it is relevant to consider a Bayesian model based change point detection method for discrete time multvariate counting processes. Chapter 3 gives some background on the standard Bayesian model based change point analysis framework, which is the stepping stone of the change detection techniques discussed in the report. Chapter 4 discusses how to perform standard Bayesian model based change point analysis for discrete time multvariate counting processes. Results from a simulation study show that the method successfully infers the number and the positions of the change points. Chapter 5 deals with the extension of the framework presented in Chapter 4 to a binary marked change point analysis. Via a simulation study, we show that the method may be used

to detect an a priori unknown number of change points, their positions and how the multiple processes are affected. We also propose an experiment to determine the detection threshold of the method. In Chapter 6, the statistical framework discussed in Chapter 5 is applied to data from LANL introduced in Chapter 2, illustrating the relevance of the approach. Since the data are not labelled, we modify the data to simulate evidence for a WannaCry ransomware attack and we show that the method successfully detects the attack. Finally, Chapter 7 concludes and discusses future work.

Chapter 2

Data from an Enterprise Computer Network

The data collected over 90 days from the network environment at Los Alamos National Laboratory (LANL), introduced in Turcotte et al. (2017), consist of some of the typical data sources, which are available in an enterprise computer network: network flows and host (computer) event logs. In this chapter, we introduce LANL data to motivate the change detection frameworks considered in Chapter 4 and 5. As described in Morgan et al. (2016), if a cyber attack occurs on an enterprise network, evidence could be found in these data sources since the attackers would need to communicate with some of the devices on the network, as well as to influence their internal activity. It is noteworthy that many of the same computers may be found both in the NetFlow data and in the host event log data, and therefore the LANL data make it possible to test in Chapter 6 the change detection method discussed in Chapter 5, which simultaneously monitors the activity between and within computers to borrow strength from multiple sources.

2.1 Network flow data

The network flow data are comprised of over 10^{10} records describing communication events between devices within LANL network. The data result from the transformation of raw data consisting of Cisco NetFlow V9 (Claise, 2004) flow records originating from the routers within LANL network. The details of the transformation may be found in Turcotte, (2017). Each record in the resulting data corresponds to a summary of a directed bi-directional network communication between two devices in the network, a source device which likely initiated the communication and a destination device. Table 2.1 gives a short description for some relevant fields found in the records. The field *DstPort* is of particular interest when it comes to describing a communication event because it gives information about the type of service being used. For example, the port with associated number 80 usually corresponds to a web browser communication (HTTP) and the port 443 to a secure web browser communication (HTTPS). In LANL data, well known ports are referred to by their associated numbers (between 0 and 1024), but some other ports have been de-identified for security reasons and are referred to by a random number strictly greater than 1024 prepended with '*Port*'.

Field Name	Description
Time	Start time of the communication event in epoch time format.
Duration	Duration of the event in seconds.
SrcDevice	Device which likely initiated the communication event.
DstDevice	Receiving device.
Protocol	Protocol number.
SrcPort	Port used by *SrcDevice*.
DstPort	Port used by *DstDevice*.

Table 2.1: Description of some relevant fields of the network flow records from the LANL data. Reproduced from Turcotte et al. (2017).

2.2 Windows host log data

The host (computer) event log data consist of over 10^{10} records of authentication and process activity for each computer running the Microsoft Windows operating system on LANL network. Each record is a summary of an event happening on a device, and necessarily contains the following fields: *Time*, *EventID* and *LogHost*, which refer to the start time of the event in epoch time format, the four digit integer event identifier and the hostname of the network device where the event is recorded, respectively. *EventID* may take 20 different possible values corresponding to different types of event. Some important examples include 4688, which corresponds to the start of a process, 4608 to Windows starting up and 4625 to a logon failure. The full list of the *EventIDs* may be found in Turcotte et al. (2017).

2.3 Motivation for the model building process

Both for the network flow and the event log data, the fields are categorical or may be grouped into categories. In order to monitor the activity within and between computers, both for the network flow and the host log data we consider time ordered sequences of counts of identical records with respect to some fields over equal length intervals of time. Examples of such sequences are shown in Figure 2.1. In the top plot of Figure 2.1 we show the number of communication events recorded per destination port[1] (*DstPort*) for each hour over the course of one week. The bottom plot of Figure 2.2 displays the number of event records per event identifier (*EventID*) for each hour over the course of one week. Formally, each such sequence of T observations may be expressed by

$$y_{1:T} = (y_1, \ldots, y_T), \tag{2.1}$$

where for each time interval t, $y_t = (y_{t,1}, \ldots, y_{t,m})$, with $y_{t,r}$ denoting the number of times the r-th category of m possible categories is observed amongst the $n_t = \sum_{r=1}^{m} y_{t,r}$ observations made during the time interval t. For example, in the case of the bottom plot in Figure 2.2, we consider records of the Windows host log data, the categories correspond to the $m = 20$ possible *EventIDs* and each time interval t corresponds to one hour of the week. We note that these sequences may also be derived on filtered versions of the data, such that one can

[1]Only the 10 most recurrent destination ports in the records over the week are displayed.

monitor the behaviour of a particular computer in the network for example. Figure 2.2 shows the sequences of statistics shown in Figure 2.1 but for a single computer rather than for all the computers in the network.

We argue that it is pertinent to monitor the behaviour of computers in the network by considering jointly sequences of count statistics of the data defined as in (2.1). First, important aspects of the behaviour of the network may be observed in Figures 2.1. For example, we observe at the 8th hour of the day 16 an increase in the number of communications and an increase in the number of events recorded on the computers since it is the beginning of a working day. Moreover, some attacks would manifest in the sequences. An increase in the number of communication events might indicate the network is being scanned for vulnerabilities, whilst the exploitation of a vulnerability in a port by attackers would increase the relative popularity of the port in the network. We note that the infamous WannaCry ransomware attack of May 2017 (CERT-EU, 2017) exploited a vulnerability in port 445 of Windows hosts to propagate through enterprise computer networks. An increase in the occurences of logon failures (*EventID* 4625) could indicate an attacker is trying to access computers, whilst an unusual increase in the activity of a computer may indicate it has been infected. These examples illustrate that for each sequence, as defined in (2.1), a temporal change in the number of events, n_t, or in the relative importance of the categories, y_t given n_t, can be an indicator of an attack.

In this context, the distribution of each sequence $y_{1:T}$ is fully specified by assuming that independently for each time unit t, the overall number of events, n_t, has a Poisson distribution and that the relative importance of the categories, y_t given n_t, has a multinomial distribution. A Bayesian model-based change point approach, as described in Chapter 3, then allows temporal changes in the distribution of the sequence. In Chapter 4, we present a framework, which detects changes in the joint distribution of a given number of sequences of counts, as defined in (2.1), which provides a holistic representation of the behaviour of computers in an enterprise network. Once a change has been detected it is of interest to provide an explanation for the change since analysts need to distinguish attacks from false alerts and in the case of an attack they must understand its nature to limit its impact on the network. For instance, in the case of the WannaCry attack (CERT-EU, 2017), it is key to quickly detect that the increase in the number of communications is largely driven by connections to port 445. In Chapter 5, we extend the model discussed in Chapter 4 such that each change may be contextualised. Finally, we note that the change detection method of Chapter 5 will be tested on the data displayed in Figure 2.2, which provide a representation of a computer activity from multiple sources.

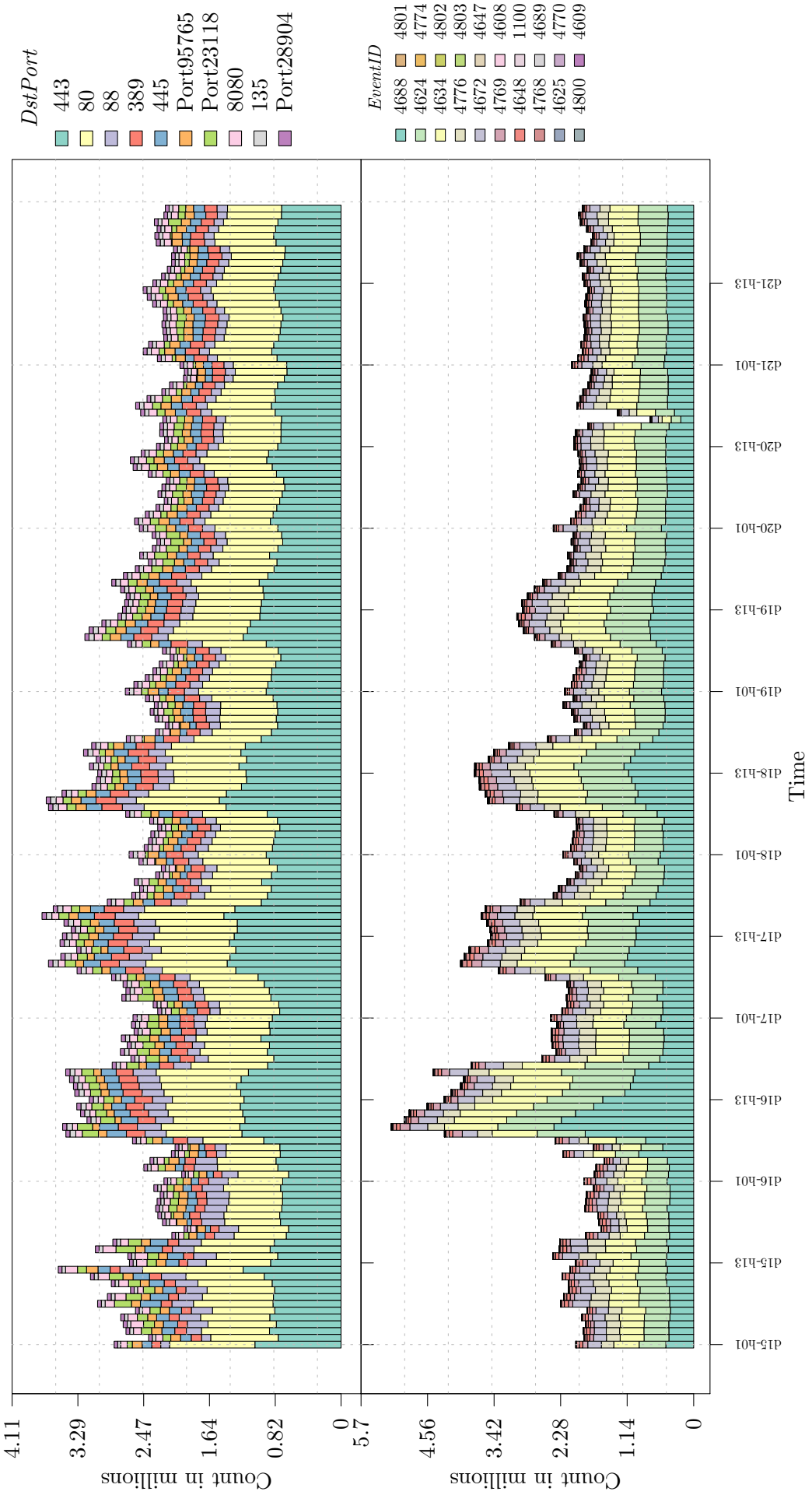

Figure 2.1: Number of communication events per destination port (top) and number of event records per event identifier (bottom) for each hour over the course of one week for LANL network.

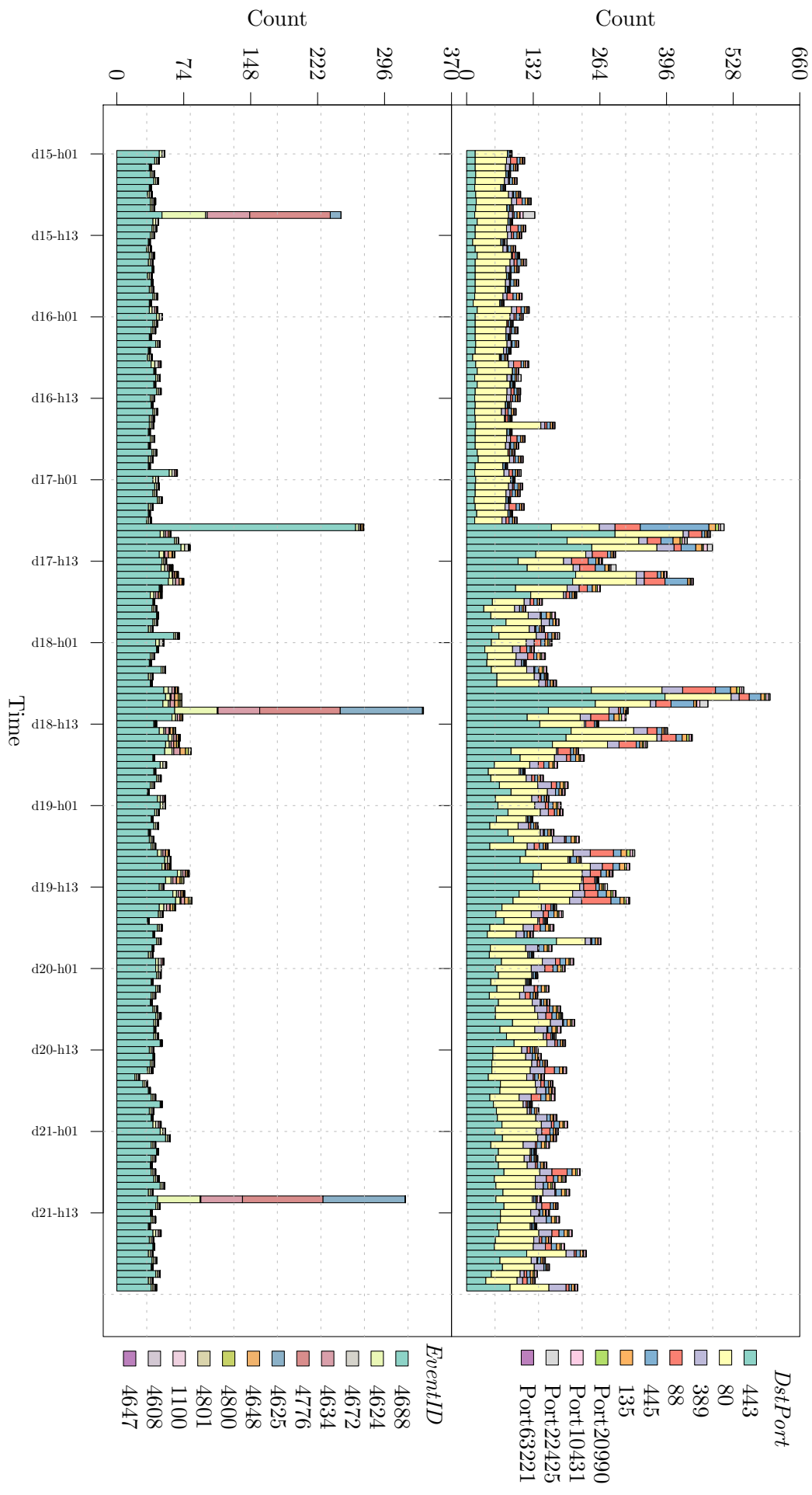

Figure 2.2: Number of communication events per destination port (top) and number of event records per event identifier (bottom) for each hour over the course of one week for the computer *Comp8417396*.

Chapter 3

Background to Bayesian Model-based Change Point Analysis

The statistical framework to monitor changes in the behaviour of computers within a network, which is motivated in Section 2.3, relies on a change point model for multivariate discrete time count processes. In this chapter, we give some background on the change point methodology considered. Change point models assume an unknown number of change points separate the data into disjoint segments, such that the data result from the same model within each segment but from different models across segments. The methodology of interest infers the number and the locations of the change points in a Bayesian framework. This chapter is concerned with the introduction of the change point model in a parametric setting as considered in Bolton and Heard (2018), Bolton (2016) and Turcotte (2014): the change points define segments where the distribution of the data has the same functional form dependent on some unknown parameters which change across segments, and with the description of the Reversible Jump MCMC methodology introduced in Green (1995), which is used to make inference on the change points.

3.1 Change point model

Let the data be $y_{1:T} = (y_1, \ldots, y_T)$, where T denotes the number of discrete time observations made. The change point model assumes that there exist k change points, denoted by $\tau_{1:k}$, from the set

$$\mathbb{T}_k = \{(\tau_1, \ldots, \tau_k) \in \mathbb{N}^k, \quad 1 \equiv \tau_0 < \tau_1 < \ldots < \tau_k < \tau_{k+1} \equiv T + 1\}, \tag{3.1}$$

which partition the data into $k+1$ independent segments with the j-th segment corresponding to $S(j) \equiv \{\tau_{j-1}, \ldots, \tau_j - 1\}$ and some segment parameters $\psi = (\psi_1, \ldots, \psi_{k+1}) \in \Psi^{k+1}$ such that for all $j = 1, \ldots, k+1$, we have independently for all $t \in S(j)$

$$y_t \sim f(\cdot \,|\, \psi_j), \tag{3.2}$$

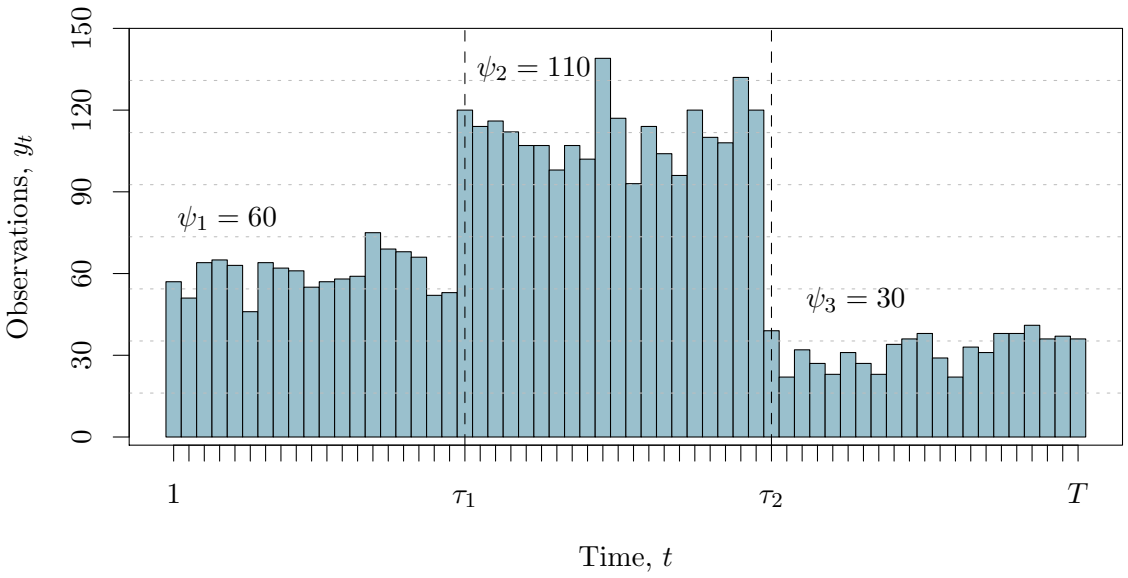

Figure 3.1: Example of observations from a change point model.

where $f(\cdot \mid \psi)$ denotes a probability distribution dependent on the parameter ψ. Conditional on the change points and the segment parameters, it follows that the likelihood of the data is

$$f(y_{1:T} \mid \tau_{1:k}, k, \psi) = \prod_{j=1}^{k+1} \prod_{t=\tau_{j-1}}^{\tau_j - 1} f(y_t \mid \psi_j). \tag{3.3}$$

Figure 3.1 shows an example of observations from a change point model with $k = 2$ change points, $\tau_{1:2} = (\tau_1, \tau_2)$, where $f(\cdot \mid \psi)$ is the Poisson distribution with rate ψ. The values of the segment parameters $\psi = (\psi_1, \psi_2, \psi_3)$ are displayed on the plot.

3.2 Bayesian framework

In the Bayesian framework, given a prior distribution on the model parameters $(k, \tau_{1:k}, \psi)$, denoted by $\pi(k, \tau_{1:k}, \psi)$, the change point detection consists of estimating the posterior distribution of $(k, \tau_{1:k}, \psi)$, or $(k, \tau_{1:k})$ in the case where the segment parameters may be marginalised.

Via Bayes' theorem, we obtain an expression of the posterior distribution of the model parameters up to a multiplicative constant:

$$\pi(k, \tau_{1:k}, \psi \mid y_{1:T}) \propto f(y_{1:T} \mid \tau_{1:k}, k, \psi)\pi(k, \tau_{1:k}, \psi). \tag{3.4}$$

Assuming independence of the segment parameters, the prior distribution is constructed as

$$\pi(k, \tau_{1:k}, \psi) = \pi(k, \tau_{1:k})\pi(\psi \mid k, \tau_{1:k}) = \pi(k, \tau_{1:k}) \prod_{j=1}^{k+1} \pi(\psi_j), \tag{3.5}$$

and in order to keep the computational cost of the inference manageable, the prior distribution on the segment parameters is chosen to be conjugate prior for the probability distribution of

the data, such that we obtain an expression of the posterior distribution of the change points up to a multiplicative constant

$$\pi(k, \tau_{1:k} \,|\, y_{1:T}) \propto \pi(k, \tau_{1:k}) f(y_{1:T} \,|\, \tau_{1:k}, k) = \pi(k, \tau_{1:k}) \int f(y_{1:T} \,|\, \tau_{1:k}, k, \psi) \pi(\psi \,|\, \tau_{1:k}, k) \mathrm{d}\psi \quad (3.6)$$

3.3 Inference with simulation

Since the posterior distribution of the parameters of interest is known only up to a multiplicative constant, inference using simulation is required. In (3.4) and (3.6) the dimension of the parameter of interest is not fixed because the number of change points is considered unknown. The Reversible Jump Markov Chain Monte Carlo (RJMCMC), which has been introduced in Green (1995), is a Metropolis-Hastings algorithm method used to sample from target distributions of varying dimension. We give below some background on the Metropolis Hastings algorithm and on the RJMCMC algorithm, as well as some general indications on how it will be used in the context of sampling from the posterior distribution of the change points.

3.3.1 Metropolis Hastings algorithm

The Metropolis Hastings algorithm generates a sample from a distribution f by constructing a Markov chain whose stationary distribution is f when only γ, an unnormalised version of f, is available. Details of the methodology may be found in Robert and Casella (2005). The corresponding algorithm is given in Algorithm 1. We note that it relies on a conditional distribution $q(\cdot \,|\, x)$ with respect to the measure of the model, which suggests a candidate for the next element of the chain given the latest element x.

Algorithm 1 Metropolis Hastings algorithm

1: **Input:** number of samples M, proposal density $q(\cdot \,|\, x)$, unnormalised target density γ
2: **Result:** Markov chain $x_1, \ldots, x_M$ with stationary distribution f
3: **Initialisation** at iteration $t = 1$:
4: set starting value x_1
5: **For iteration** $t = 2, \ldots, M$:
6: make a proposal $y \sim q(\cdot \,|\, x_{t-1})$
7: Obtain the acceptance probability of the proposal

$$\alpha = 1 \wedge \frac{\gamma(y) q(\, x_{t-1} \,|\, y)}{\gamma(x_{t-1}) q(\, y \,|\, x_{t-1})}$$

8: Sample $u \sim \mathrm{Unif}(0, 1)$
9: **if** $u < \alpha$ **then** $x_t = y$ **else** $x_t = x_{t-1}$

3.3.2 Reversible Jump MCMC

Consider a Bayesian variable dimension model defined as a collection of models

$$M_k = \{f(\cdot \,|\, \phi_k) \,:\, \phi_k \in \Phi_k\}, \quad k = 0, \ldots, K, \quad (3.7)$$

where Φ_k is the parameter space for the model M_k. The RJMCMC algorithm generates a sample from $\pi(k, \phi_k)$, the joint distribution of the model index and the corresponding parameters, by constructing a Markov chain with stationary distribution $\pi(k, \phi_k)$ when only $\gamma(k, \phi_k)$, an unnormalised version of $\pi(k, \phi_k)$, is available. This is made possible by the dimension matching moves between pairs of models proposed in Green (1995).

Suppose that a move from model M_{k_1} to model M_{k_2} is proposed with probability $r(k_1, k_2)$, and that the move may be reversed by proposing a move from model M_{k_2} from model M_{k_1} with probability $r(k_2, k_1)$. Dimension matching of the move is achieved by augmenting the parameter spaces Φ_{k_1} and Φ_{k_2} by sets U_1 and U_2, such that $\dim(\Phi_{k_1}) + \dim(U_1) = \dim(\Phi_{k_2}) + \dim(U_2)$. Specifically, (k_2, ϕ_{k_2}) is proposed from (k_1, ϕ_{k_1}) by sampling $u_1 \in U_1$ and $u_2 \in U_2$ from the proposal densities $g_{k_1,k_2}(\cdot)$ and $g_{k_2,k_1}(\cdot)$, respectively, and by setting $(\phi_{k_2}, u_2) = \mathrm{T}_{k_1,k_2}(\phi_{k_1}, u_1)$, where T_{k_1,k_2} is a diffeomorphism.

The general form of the RJMCMC sampler as well as the expression of the acceptance probability of a move are given in Algorithm 2.

Algorithm 2 Reversible Jump MCMC algorithm

1: **Input:** number of samples N, model jump probabilities, proposal densities, diffeomorphisms, unnormalised target density γ
2: **Result:** Markov chain $(k_1, \phi_{k_1}), \ldots, (k_N, \phi_{k_N})$ with stationary distribution $\pi(k, \phi_k)$,
3: **Initialisation** at iteration $t = 1$:
4: set starting value (k_1, ϕ_{k_1})
5: **For iteration** $t = 2, \ldots, N$:
6: propose a move from model $M_{k_{t-1}}$ to model M_{k^*}
7: sample $u_{k^*} \sim g_{k_{t-1},k^*}(\cdot)$ and $u_{k_{t-1}} \sim g_{k^*,k_{t-1}}(\cdot)$
8: set $(\phi_{k^*}, u_{k^*}) = \mathrm{T}_{k_{t-1},k^*}(\phi_{k_{t-1}}, u_{k_{t-1}})$
9: Obtain the acceptance probability of the proposal

$$\alpha = 1 \wedge \frac{\gamma(k^*, \phi_{k^*})}{\gamma(k_{t-1}, \phi_{k_{t-1}})} \frac{r(k^*, k_{t-1})}{r(k_{t-1}, k^*)} \frac{g_{k^*,k_{t-1}}(u_{k_{t-1}})}{g_{k_{t-1},k^*}(u_{k^*})} \left| \frac{\partial \mathrm{T}_{k_{t-1},k^*}(\phi_{k_{t-1}}, u_{k_{t-1}})}{\partial(\phi_{k_{t-1}}, u_{k_{t-1}})} \right|$$

10: Sample $u \sim \mathrm{Unif}(0, 1)$
11: **if** $u < \alpha$ **then** $(k_t, \phi_{k_t}) = (k^*, \phi_{k^*})$ **else** $(k_t, \phi_{k_t}) = (k_{t-1}, \phi_{k_{t-1}})$

As described in Green (1995) the RJMCMC algorithm may be used to obtain a sample from the posterior distribution of the parameters of interest in the Bayesian model based change point analysis. The collection of models in (3.7) is defined so that the model M_k corresponds to a change point model with k change points. In the case corresponding to (3.6) where the segment parameters are marginalised, we have that for each model M_k, the parameter space is $\Phi_k = \mathbb{T}_k$, and three types of moves are considered: deletion of one change point; addition of one change point and change in the position of one change point. In the case where the segment parameters are not marginalised such as in (3.4), we have that for each model M_k, the parameter space is $\Phi_k = \mathbb{T}_k \times \Psi^{k+1}$, and one additional type of moves corresponding to the resampling of one segment parameter is needed. The details of the moves, corresponding to the lines 7 to 9 of the Algorithm 2, are given in the context of the specific models considered in Chapter 4 and 5.

Chapter 4

Change Point Analysis for Discrete Time Multivariate Counting Processes

To detect cyber attacks it is relevant to monitor changes in the behaviour of computers within an enterprise computer network. As explained in Section 2.3, a Bayesian model based change point analysis for discrete time multivariate counting processes is appropriate in this context. It borrows strength from multiple sources to detect changes in both the activity between and within computers in the network. In this chapter, we describe how the Bayesian model based change point analysis discussed in Chapter 3 may be applied to the stochastic process described in Section 2.3, which provides a holistic representation of the behaviour of computers in an enterprise network. We also present results from a simulation study assessing the accuracy of the methodology in inferring the number and the positions of the change points.

4.1 Bayesian change point model

In this section, we discuss the Bayesian change point model for discrete time multivariate counting processes.

4.1.1 The likelihood function

Consider T equal length time intervals, and let the data $\boldsymbol{y}$ consist of the realisations $y_1, \ldots, y_P$ of P *independent* discrete time processes defined as in (2.1). It follows that, for all processes $\ell = 1, \ldots, P$, there exists a corresponding fixed number of categories m_ℓ, and $y_\ell \equiv (y_{\ell,1}, \ldots, y_{\ell,T})$, where for each t, $y_{\ell,t} = (y_{\ell,t,1}, y_{\ell,t,2}, \ldots, y_{\ell,t,m_\ell})$, with $y_{\ell,t,r}$ denoting the number of times the r-th category is observed amongst the $n_{\ell,t} \equiv \sum_{r=1}^{m_\ell} y_{\ell,t,r}$ observations made during the time interval t for the ℓ-th process. The sequence of the overall numbers of observations at each time point is denoted by $n_\ell = (n_{\ell,1}, \ldots, n_{\ell,T})$ for all processes $\ell = 1, \ldots, P$.

Let $\tau_{1:k}$ denote the k change points for the joint distribution of the data as defined in (3.1). Motivated by the discussion in Section 2.3, we assume that independently for all $\ell = 1, \ldots, P$,

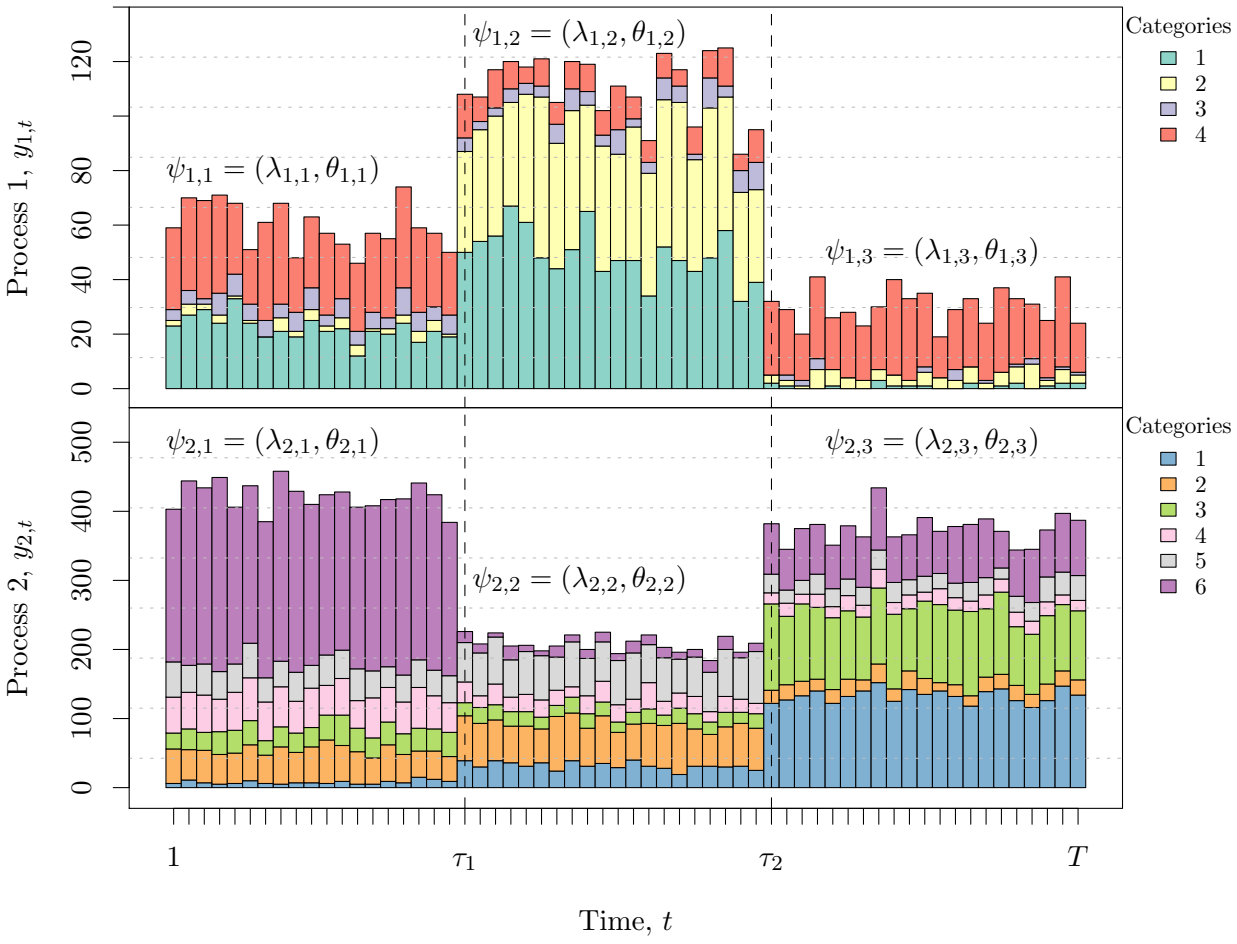

Figure 4.1: Data from two multivariate counting processes subject to change points.

$j = 1, \ldots, k+1$ and $t = \tau_{j-1}, \ldots, \tau_j - 1$

$$n_{\ell,t} \,|\, \lambda_{\ell,j} \sim \text{Poisson}(\lambda_{\ell,j}) \quad \text{and} \quad y_{\ell,t} \,|\, n_{\ell,t}, \theta_{\ell,j} \sim \text{Multinomial}(n_{\ell,t}, \theta_{\ell,j}), \tag{4.1}$$

where $\lambda_{\ell,j} > 0$ and $\theta_{\ell,j} = (\theta_{\ell,j,1}, \ldots, \theta_{\ell,j,m_\ell})$, such that $\sum_{r=1}^{m_\ell} \theta_{\ell,j,r} = 1$ and $\theta_{\ell,j,r} \geqslant 0$ for any $r = 1, \ldots, m_\ell$. Let the model parameter be $\boldsymbol{\psi} = (\psi_1, \ldots, \psi_P)$, where $\psi_\ell = (\psi_{\ell,1}, \ldots, \psi_{\ell,k+1}) \in \Psi^{k+1}$ is the vector of the segment parameters of the ℓ-th process, such that for all $j = 1, \ldots, k+1$, the parameter corresponding to the j-th segment resulting from the change points is $\psi_{\ell,j} = (\lambda_{\ell,j}, \theta_{\ell,j})$. As a result, conditional on k, $\tau_{1:k}$ and $\boldsymbol{\psi}$, the likelihood of the data is

$$f(\boldsymbol{y}|k, \tau_{1:k}, \boldsymbol{\psi}) = \prod_{\ell=1}^{P} \prod_{j=1}^{k+1} \prod_{t=\tau_{j-1}}^{\tau_j - 1} f(y_{\ell,t}|\psi_{\ell,j}) = \prod_{\ell=1}^{P} \prod_{j=1}^{k+1} \prod_{t=\tau_{j-1}}^{\tau_j - 1} f(y_{\ell,t}|n_{\ell,t}, \theta_{\ell,j}) f(n_{\ell,t}|\lambda_{\ell,j}). \tag{4.2}$$

Figure 4.1 shows data observed from the model described above in a special case where there are $k = 2$ change points $\tau_{1:k} = (\tau_1, \tau_2)$ and $P = 2$ processes are considered, with $m_1 = 4$ categories for the first one and $m_2 = 6$ ones for the second one. At each change point, for

each process, one observes a change in the distribution of the overall count as well and in the distribution of the relative importance of the categories per time unit.

4.1.2 Prior distribution

As in (3.5), independence of the change points with the segment parameters is assumed, such that the prior distribution of the parameters may be expressed as

$$\pi(k, \tau_{1:k}, \boldsymbol{\psi}) = \pi(k, \tau_{1:k}) \prod_{\ell=1}^{P} \prod_{j=1}^{k+1} \pi(\theta_{\ell,j}) \pi(\lambda_{\ell,j}). \tag{4.3}$$

As in Bolton and Heard (2018), a Bernoulli process is assumed a priori on the change points, so that we have the joint probability mass function: $\pi(k, \tau_{1:k}) = p^k (1-p)^{T-k}$, for some $0 < p < 1$, which is chosen to encode our prior belief on the number of change points. Moreover, for each $\ell = 1, \ldots, P$ and $j = 1, \ldots, k+1$, the conjugate prior distributions Dirichlet(α_ℓ) and Gamma(δ_ℓ, β_ℓ) are chosen for $\theta_{\ell,j}$ and $\lambda_{\ell,j}$, respectively, for some $\alpha_\ell = (\alpha_{\ell,1}, \ldots, \alpha_{\ell,m_\ell})$, such that $\alpha_{\ell,r} > 0$ for all $r = 1, \ldots, m_\ell$, and some $\delta_\ell > 0$ and $\beta_\ell > 0$.

4.1.3 Marginal posterior distribution of the change points

Following the approach described in Section 3.2, we obtain the following expression for an unnormalised marginal posterior probability mass function of the change points

$$\pi(k, \tau_{1:k} | \boldsymbol{y}) \propto \pi(k, \tau_{1:k}) f(\boldsymbol{y} | k, \tau_{1:k}) = \pi(k, \tau_{1:k}) \prod_{\ell=1}^{P} f(y_\ell | k, \tau_{1:k}), \tag{4.4}$$

where, for all $\ell = 1, \ldots, P$, we have, via (4.2) and (4.3),

$$\begin{aligned} f(y_\ell | k, \tau_{1:k}) &= \int f(y_\ell | k, \tau_{1:k}, \psi_\ell) \pi(\psi_\ell) \mathrm{d}\psi_\ell \\ &= \left(\int f(y_\ell | n_\ell, k, \tau_{1:k}, \theta_\ell) \pi(\theta_\ell) \mathrm{d}\theta_\ell \right) \left(\int f(n_\ell | k, \tau_{1:k}, \lambda_\ell) \pi(\lambda_\ell) \mathrm{d}\lambda_\ell \right) \\ &= f(y_\ell | n_\ell, k, \tau_{1:k}) f(n_\ell | k, \tau_{1:k}). \end{aligned} \tag{4.5}$$

Conjugacy of the gamma distribution with the Poisson distribution, and of the Dirichlet distribution with the multinomial distribution yield the exact expression of (4.5). For each $\ell = 1, \ldots, P$, we have

$$\begin{aligned} f(n_\ell | k, \tau_{1:k}) &= \prod_{j=1}^{k+1} \int \left\{ \prod_{t=\tau_{j-1}}^{\tau_j - 1} f(n_{\ell,t} | \lambda_{\ell,j}) \right\} \pi(\lambda_{\ell,j}) \mathrm{d}\lambda_{\ell,j} \\ &= \prod_{j=1}^{k+1} \int \left\{ \prod_{t=\tau_{j-1}}^{\tau_j - 1} \frac{\lambda_{\ell,j}^{n_{\ell,t}}}{n_{\ell,t}!} \exp(-\lambda_{\ell,j}) \right\} \frac{\beta_\ell^{\delta_\ell}}{\Gamma(\delta_\ell)} \lambda_{\ell,j}^{\delta_\ell - 1} \exp(-\beta_\ell \lambda_{\ell,j}) \mathrm{d}\lambda_{\ell,j} \\ &= \left(\prod_{t=1}^{T} \Gamma(n_{\ell,t} + 1) \right)^{-1} \left(\frac{\beta_\ell^{\delta_\ell}}{\Gamma(\delta_\ell)} \right)^{k+1} \left(\prod_{j=1}^{k+1} \frac{\Gamma(n_\ell^{(j)} + \delta_\ell)}{(\beta_\ell + s_j)^{n_\ell^{(j)} + \delta_\ell}} \right), \end{aligned} \tag{4.6}$$

where $s_j = |s(j)| = \tau_j - 1 - \tau_{j-1} + 1 = \tau_j - \tau_{j-1}$ is the number of time intervals contained in the j-th segment $S(j)$, and $n_\ell^{(j)} = \sum_{t=\tau_{j-1}}^{\tau_j - 1} n_{\ell,t}$ is the number of observations made on the j-th segment; and

$$\begin{aligned} f(y_\ell | n_\ell, k, \tau_{1:k}) &= \prod_{j=1}^{k+1} \int \left\{ \prod_{t=\tau_{j-1}}^{\tau_j - 1} f(y_{\ell,t} | n_{\ell,t}, \theta_{\ell,j}) \right\} \pi(\theta_{\ell,j}) \mathrm{d}\theta_{\ell,j} \\ &= \prod_{j=1}^{k+1} \int \left\{ \prod_{t=\tau_{j-1}}^{\tau_j - 1} \frac{\Gamma(1 + n_{\ell,t})}{\prod_{r=1}^{m_\ell} \Gamma(1 + y_{\ell,t,r})} \prod_{r=1}^{m_\ell} \theta_{\ell,j,r}^{y_{\ell,t,r}} \right\} \frac{\Gamma(\sum_{r=1}^{m_\ell} \alpha_{\ell,r})}{\prod_{r=1}^{m_\ell} \Gamma(\alpha_{\ell,r})} \prod_{r=1}^{m_\ell} \theta_{\ell,j,r}^{\alpha_{\ell,r} - 1} \mathrm{d}\theta_{\ell,j} \\ &= \left(\prod_{t=1}^{T} \frac{\Gamma(1 + n_{\ell,t})}{\prod_{r=1}^{m_\ell} \Gamma(1 + y_{\ell,t,r})} \right) \left(\prod_{j=1}^{k+1} \frac{\Gamma(\alpha_\bullet^\ell) \prod_{r=1}^{m_\ell} \Gamma(y_{\ell,\cdot,r}^{(j)} + \alpha_{\ell,r})}{\Gamma(n_\ell^{(j)} + \alpha_\bullet^\ell) \prod_{r=1}^{m_\ell} \Gamma(\alpha_{\ell,r})} \right), \end{aligned} \tag{4.7}$$

where $\alpha_\bullet^\ell = \sum_{r=1}^{m_\ell} \alpha_{\ell,r}$ and $y_{\ell,\cdot,r}^{(j)} = \sum_{t=\tau_{j-1}}^{\tau_j - 1} y_{\ell,t,r}$ denotes the number of times the r-th category is observed amongst the $n_\ell^{(j)} = \sum_{t=\tau_{j-1}}^{\tau_j - 1} n_{\ell,t}$ observations made on the j-th segment.

We obtained an expression of the marginal posterior probability mass function of the change points up to a normalising constant, γ, as in (3.6), which we can evaluate pointwise:

$$\gamma(k, \tau_{1:k}) \equiv \gamma(k, \tau_{1:k} | \boldsymbol{y}) = \pi(k, \tau_{1:k}) \prod_{\ell=1}^{P} f(y_\ell | n_\ell, k, \tau_{1:k}) f(n_\ell | k, \tau_{1:k}). \tag{4.8}$$

4.2 RJMCMC sampling

In this section, we discuss how to sample from the posterior distribution of the change points.

4.2.1 The procedure

Since the dimensionality of the parameter of interest, $(k, \tau_{1:k})$, is not fixed, in order to sample from the marginal posterior distribution of the change points, we consider the Reversible Jump MCMC procedure which we discussed in Section 3.3.2. The segment parameters are marginalised in (4.8), and therefore, for each model M_k, where k indicates the number of change points, the parameter space is $\Phi_k = \mathbb{T}_k$. To explore the support of the posterior distribution of the change points, $\bigcup_k \{k\} \times \mathbb{T}_k$, we consider the following three types of moves

- Shift: A change in the position of one of the change points,
- Birth: The addition of a change point,
- Death: The deletion of one of the change points.

Given that the latest element of the chain corresponds to k change points, let r_k^s, r_k^b and r_k^d, denote the probabilities of the shift, birth and death moves, respectively, such that $r_k^s + r_k^b + r_k^d = 1$, for all k. In particular, we set $r_{T-1}^b = r_0^d = r_0^s = 0$ because one cannot change the position or delete a change point, if there are no change points, and if all time points are change points, then one cannot add a change point.

The RJMCMC algorithm given in Algorithm 2 in Section 3.3.2, considered with these types of moves and by noting that $\Phi_k = \mathbb{T}_k$, yields a Markov chain with stationary distribution

$\pi(k, \tau_{1:k}|\boldsymbol{y})$. For each type of moves, a description and the corresponding acceptance probability corresponding to lines 7 to 9 of the Algorithm 2 are given below.

4.2.2 The acceptance probabilities

Suppose we are at the $(t+1)$-th iteration of the algorithm, with $(k, \tau_{1:k})$ being the element of the chain at time t.

Birth move: Addition of a change point

If a birth is proposed, then $k^* = k+1$, and τ'_h is randomly picked from $\{2, 3, \ldots, T\}\backslash\tau_{1:k}$ with probability $(T-1-k)^{-1}$, and added to $\tau_{1:k}$ to obtain $\tau^*_{1:k^*}$. The move may be reversed, with a death move, by deleting τ'_h from $\tau^*_{1:k^*}$ with probability $1/k^*$, so that the probability of accepting the move is

$$\alpha = 1 \wedge \frac{\gamma(k^*, \tau^*_{1:k^*})}{\gamma(k, \tau_{1:k})} \frac{r^d_{k^*}}{r^b_k} \frac{(T-1-k)}{k^*} = 1 \wedge \frac{p}{1-p} \frac{f(\boldsymbol{y}|k^*, \tau^*_{1:k^*})}{f(\boldsymbol{y}|k, \tau_{1:k})} \frac{r^d_{k^*}}{r^b_k} \frac{(T-1-k)}{k^*}. \tag{4.9}$$

Death move: Deletion of a change point

If a death move is proposed, then $k^* = k-1$, and the index of one element of $\tau_{1:k}$ is randomly chosen with probability $1/k$ and the corresponding change point is deleted to obtain $\tau^*_{1:k^*}$. The move may be reversed by a birth move, selecting the deleted location with probability $(T-1-k^*)^{-1}$, yielding the acceptance ratio

$$\alpha = 1 \wedge \frac{\gamma(k^*, \tau^*_{1:k^*})}{\gamma(k, \tau_{1:k})} \frac{r^b_{k^*}}{r^d_k} \frac{k}{(T-1-k^*)} = 1 \wedge \frac{1-p}{p} \frac{f(\boldsymbol{y}|k^*, \tau^*_{1:k^*})}{f(\boldsymbol{y}|k, \tau_{1:k})} \frac{r^b_{k^*}}{r^d_k} \frac{k}{(T-1-k^*)}. \tag{4.10}$$

Shift move: Position change of a change point

If a shift move is proposed, then $k^* = k$, and τ_h, one element of $\tau_{1:k}$, is replaced by τ'_h, an element randomly picked from $\{\tau_{h-1}+1, \ldots, \tau_{h+1}-1\}$. The move may be reversed by a shift move. Note that a shift move may be seen as a death move followed by a birth move. The probability of accepting the move is

$$\alpha = 1 \wedge \frac{\gamma(k^*, \tau^*_{1:k^*})}{\gamma(k, \tau_{1:k})} = 1 \wedge \frac{f(\boldsymbol{y}|k^*, \tau^*_{1:k^*})}{f(\boldsymbol{y}|k, \tau_{1:k})}. \tag{4.11}$$

4.3 Simulation study

We performed a simulation study to show that the RJMCMC sampler described in Section 4.2 may be used to successfully infer the number and the positions of the change points. We simulated 50 independent discrete time multivariate counting processes of length $T = 200$ subject to change points as defined in (4.1). For the sake of brevity, we limited ourselves to the case where each simulation consists of one 7-variate counting process, that is $P = 1$ and $m_1 = 7$ using the notations of Section 4.1.1. For each simulation, the parameters $(k, \tau_{1:k}, \boldsymbol{\psi})$ were generated from the prior distribution given in (4.3) with hyperparameters: $p = 8/200$, $\delta_1 = 20$, $\beta_1 = 0.05$ and $\alpha_1 = (1, \ldots, 1)$. The hyperparameters were chosen to obtain data

similar to the data from LANL displayed in Figure 2.2 with a variety of numbers of change points. Across the 50 simulations, the dimension of the change points varied from 1 to 14.

For each simulation, a sample from the posterior distribution of the change points was obtained by 5,000 iterations of the RJMCMC sampler with a burn-in of 1,000 iterations, with the hyperparameters set to their true values and with $k = 0$ as the starting value. We observed that, in this set-up, the number of iterations chosen was large enough for the Markov chains to converge. The estimate of k was chosen to be the mode of the dimension of the change points in the sample, that is the maximum a posteriori probability (MAP) estimate. The change points, $\tau_{1:k}$, were finally estimated to be the mode of the change points in the sample conditioned on the number of change points being equal to its MAP estimate.

Figure 4.3 shows the different steps of the estimation procedure for one simulation corresponding to a process with 11 change points. We observe that the mode of the dimension of the change point in the sample obtained with the RJMCMC sampler corresponds to the true dimension of the change points. The trace of the change points shows the Markov chain converges to true change points. Finally, the true and the estimated change points are displayed on the data showing they coincide.

Figure 4.2 displays the results of the estimation for each simulation. One observes that the estimated dimension of the change points is equal or close to the true dimension, and when k is correctly estimated, the elements of $\tau_{1:k}$ are correctly estimated too. Hence, the simulation study illustrates that the RJMCMC sampler may be used to successfully infer the number and the position of the change points.

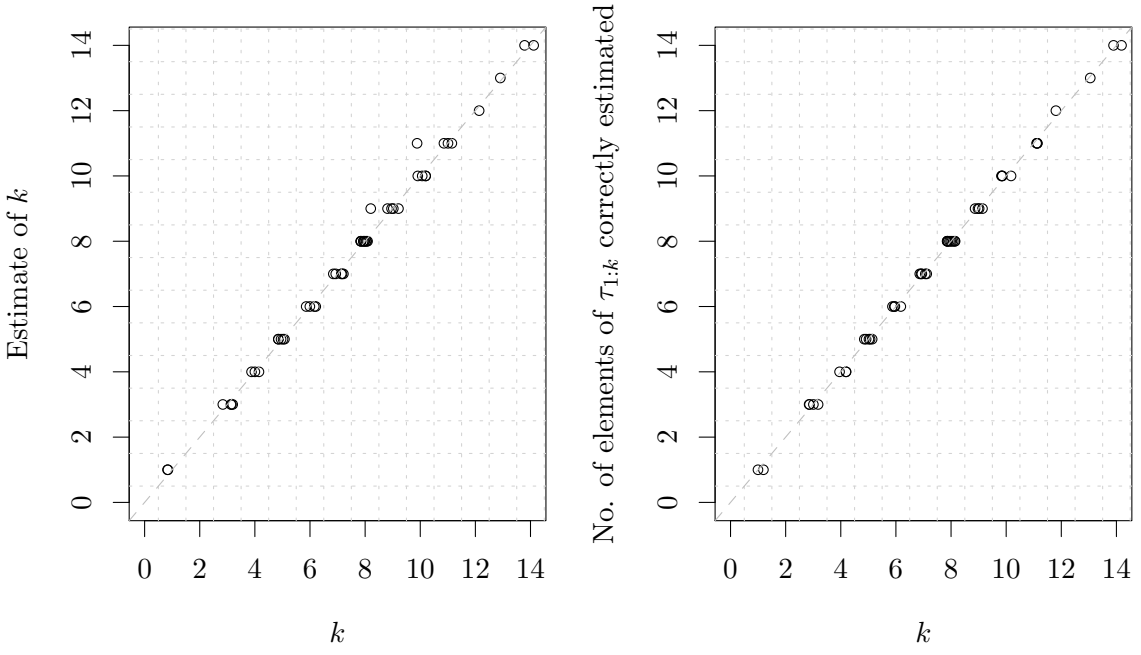

Figure 4.2: Left: Estimated number of change points versus the true number of change points for each simulation. Right: Number of elements of $\tau_{1:k}$ correctly estimated versus the dimension of the change points for each simulation where k was successfully estimated. The points are horizontally jittered within the box corresponding to their coordinates. Note these plots were chosen rather than heat maps because they may be easily extended to display results from the more flexible model discussed in the next chapter.

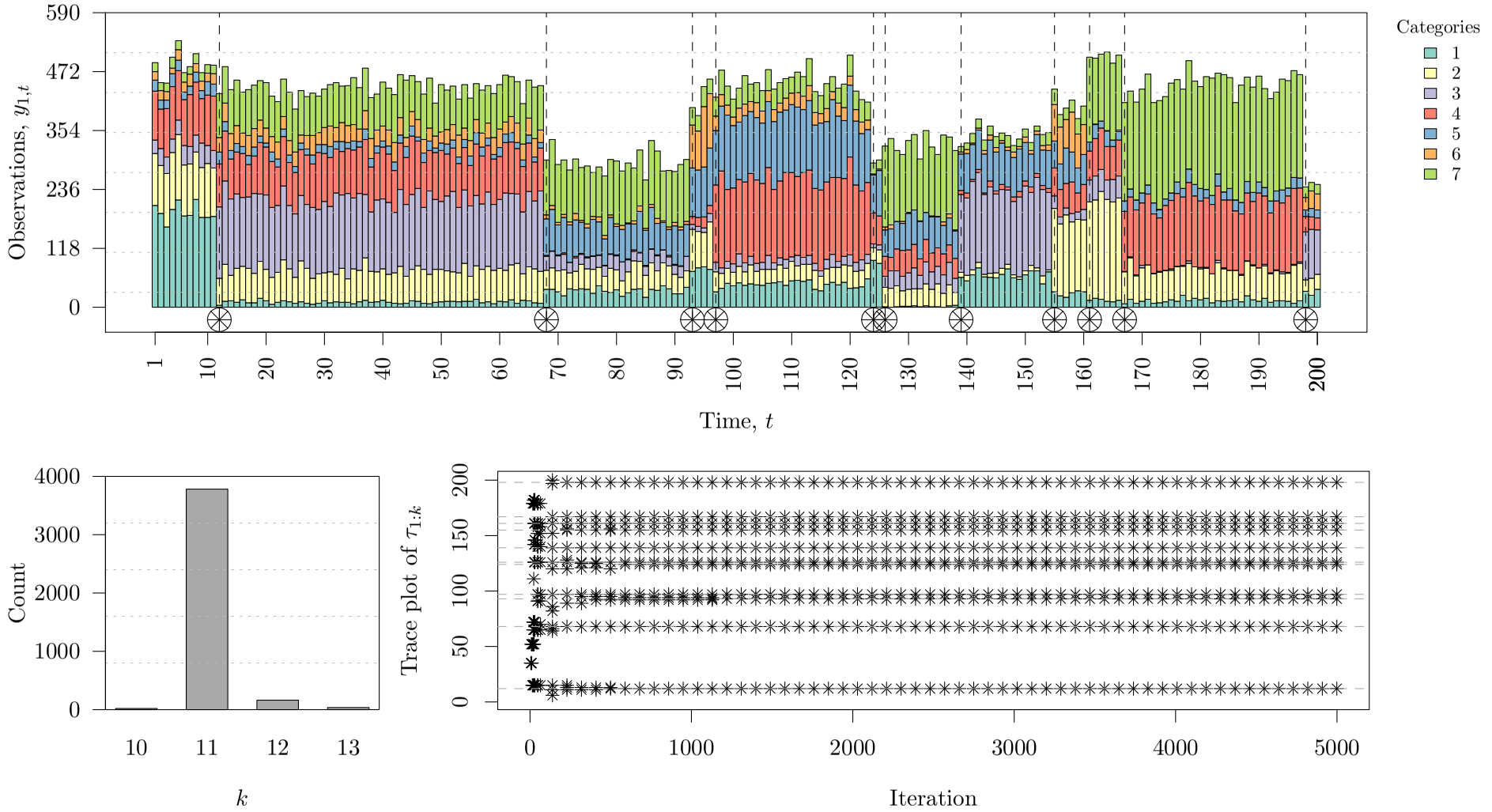

Figure 4.3: Bottom left: Number of occurences for each dimension of the change points observed in the sample for a simulation with $k = 11$. Bottom right: Trace plot of the change points in the sample, where the positions of the true change points are shown by horizontal dashed lines. Only 1 every 80 iterations are displayed to improve readability. Top: Data from the process of interest, where the true change points are identified by ○ and the estimated ones by ✳.

Chapter 5

Extension to Binary Marked Change Point Analysis

The statistical framework described in Chapter 4 detects changes in the joint distribution of sequences of multivariate count statistics, which may provide a holistic representation of the behaviour of an enterprise computer network. It relies on the standard change point model, which assumes that at each change point the parameters governing the distribution of the multiple processes change. Bolton and Heard (2018) provide an extension to the standard change point model for multiple processes, where change points may differently affect the processes of interest. A binary marked vector is associated to each change point to indicate which processes are affected by the change point. In this chapter, we describe how the statistical framework described in Chapter 4 may be extended to a binary marked change point analysis. As described in Section 2.3, when monitoring for changes in the behaviour of a network to detect attacks, it is crucial to interpret the nature of the changes in order to distinguish attacks from normal changes and to provide context to the change in the case of an attack, such that it may be counteracted quickly. Hence, the extension discussed in this chapter is of particular interest because it provides a statistical framework to monitor changes in the behaviour of the network from multiple sources, such that the context of each change may be inferred: it determines which multivariate counting processes are affected and whether it stems from a change in the distribution of the overall count, $n_{\ell,t}$, or in the distribution of the relative importance of the categories, $y_{\ell,t}$ given $n_{\ell,t}$, for each multivariate process affected by the change point. In Section 5.3, results from simulation studies assessing the accuracy of the method are provided.

5.1 Binary marked change point model

In this section, we discuss the Bayesian binary marked change point model for discrete time multivariate counting processes.

5.1.1 Binary marked change points

For the change point model introduced in Chapter 4, it is assumed in (4.2) and (4.3) that the parameters for each of the P processes change at each change point. That is, for each change point $\tau_j \in \tau_{1:k}$, for all $\ell = 1, \ldots, P$, we have that $\lambda_{\ell,j+1} \neq \lambda_{\ell,j}$ and $\theta_{\ell,j+1} \neq \theta_{\ell,j}$. We consider

a more flexible model introduced in Bolton and Heard (2018) where the change points may differently affect the parameters governing each process. The change point parameters $(k, \tau_{1:k})$ defined in (3.1) are extented to $(k, \tau_{1:k}, I_{1:k})$, with $I_{1:k} = (I_1, \ldots, I_k)$, where I_j is a $P \times 2$ matrix for each change point index $j = 1, \ldots, k$, with ℓ-th row $I_{j,\ell} \in \{0,1\}^2$ corresponding to the ℓ-th process, where $I_{j,\ell,1} = 1$ iff $\lambda_{\ell,j+1} \neq \lambda_{\ell,j}$ and $I_{j,\ell,2} = 1$ iff $\theta_{\ell,j+1} \neq \theta_{\ell,j}$. Hence, for all $j = 1, \ldots, k$, the matrix I_j determines which processes are affected by the j-th change point, τ_j, and in particular, for all $\ell = 1, \ldots, P$, the ℓ-th element of the first column indicates if the distribution of n_ℓ is affected and the ℓ-th element of the second column indicates if the distribution of y_ℓ given n_ℓ is affected.

For all $\ell = 1, \ldots, P$, let $k_{\ell,i} = \sum_{j=1}^{k} I_{j,\ell,i}$ and denote by $\tau(\ell, i)$ the $k_{\ell,i}$-vector of the ordered positions of the change points, which affect the distribution of n_ℓ for $i = 1$ and of y_ℓ given n_ℓ for $i = 2$, with $\tau(\ell, i, j)$ defined to be the j-th element of $\tau(\ell, i)$. Moreover, let k_ℓ be the dimension the change points that affect the ℓ-th process, which we denote by $\tau(\ell) = \tau(\ell, 1) \cup \tau(\ell, 2)$.

We note that if for some $j \in \{1, \ldots, k\}$, we have $I_{j,\ell,i} = 0$ for all $\ell = 1, \ldots, P$ and $i = 1, 2$, then the change point τ_j does not affect any process. Using the terminology of Bolton (2016), such a change point is coined to be ineffective, and we define the effective number of change points, k^{eff}, to be the cardinality of the set of the effective change points:

$$k^{\text{eff}} = \left| \{ j \in \{1, \ldots, k\} : \exists \ell \in \{1, \ldots, P\} \; \exists i \in \{1, 2\} \quad \text{s.t.} \quad I_{j,\ell,i} = 1 \} \right|. \tag{5.1}$$

Figure 5.1 shows data observed from the model described above in the special case where there are $k = k^{\text{eff}} = 3$ change points $\tau_{1:3} = (\tau_1, \tau_2, \tau_3)$, there are $P = 2$ processes, with $m_1 = 4$ categories for the first process and $m_2 = 6$ categories for the second process, and the binary marked matrices $I_{1:k} = (I_1, I_2, I_3)$ are

$$I_1 = \begin{pmatrix} I_{1,1,1} & I_{1,1,2} \\ I_{1,2,1} & I_{1,2,2} \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad I_2 = \begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \quad \text{and} \quad I_3 = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix},$$

such that, for example for the first process, we have $k_1 = 2$, $k_{1,1} = 1$ and $k_{1,2} = 1$.

5.1.2 Prior distribution

A priori there is no reason to believe the position of a change point has any influence on how the processes are affected, so we factorise the prior distribution of the model parameters as follows

$$\pi(k, \tau_{1:k}, I_{1:k}, \boldsymbol{\psi}) = \pi(k, \tau_{1:k}, \boldsymbol{\psi}) \pi(I_{1:k} | k), \tag{5.2}$$

where $\pi(k, \tau_{1:k}, \boldsymbol{\psi})$ is built as in (4.3). Independently, each element of $I_{1:k} \in \{0,1\}^{kP2}$ is assumed to be the realisation of a Bernoulli random variable. Some processes may be more susceptible to change than others. Hence, for all $\ell, \ldots, P$ and $i = 1, 2$, a separate Bernoulli parameter $\omega_{\ell,i}$ is assigned to the mark k-vector $I_{\cdot,\ell,i}$ that encodes which change points affect the distribution of n_ℓ if $i = 1$ or the distribution of y_ℓ given n_ℓ if $i = 2$. Formally, we assume that for all $\ell = 1, \ldots, P$ and $i = 1, 2$, independently for all $j = 1, \ldots, k$, we have $I_{j,\ell,i} \sim \text{Bernoulli}(\omega_{\ell,i})$,

for some $\omega_{\ell,i} \sim \text{Beta}(\eta, \upsilon)$, so that

$$\pi(I_{1:k}|k) = \prod_{\ell=1}^{P} \prod_{i=1}^{2} \pi(I_{\cdot,\ell,i}|k), \tag{5.3}$$

where the probability mass function of $I_{\cdot,\ell,i}$ is obtained by conjugacy of the Beta distribution with the Bernoulli distribution as follows

$$\pi(I_{\cdot,\ell,i}|k) = \int \omega_{\ell,i}^{k_{\ell,i}} (1-\omega_{\ell,i})^{k-k_{\ell,i}} \frac{\Gamma(\eta+\upsilon)}{\Gamma(\eta)\Gamma(\upsilon)} \omega_{\ell,i}^{\eta-1} (1-\omega_{\ell,i})^{\upsilon-1} \mathrm{d}\omega_{\ell,i} \tag{5.4}$$

$$= \frac{\Gamma(\eta+\upsilon)}{\Gamma(\eta)\Gamma(\upsilon)} \frac{\Gamma(\eta+k_{\ell,i})\Gamma(\upsilon+k-k_{\ell,i})}{\Gamma(\eta+\upsilon+k)}, \tag{5.5}$$

for all $\ell = 1, \ldots, P$ and $i = 1, 2$.

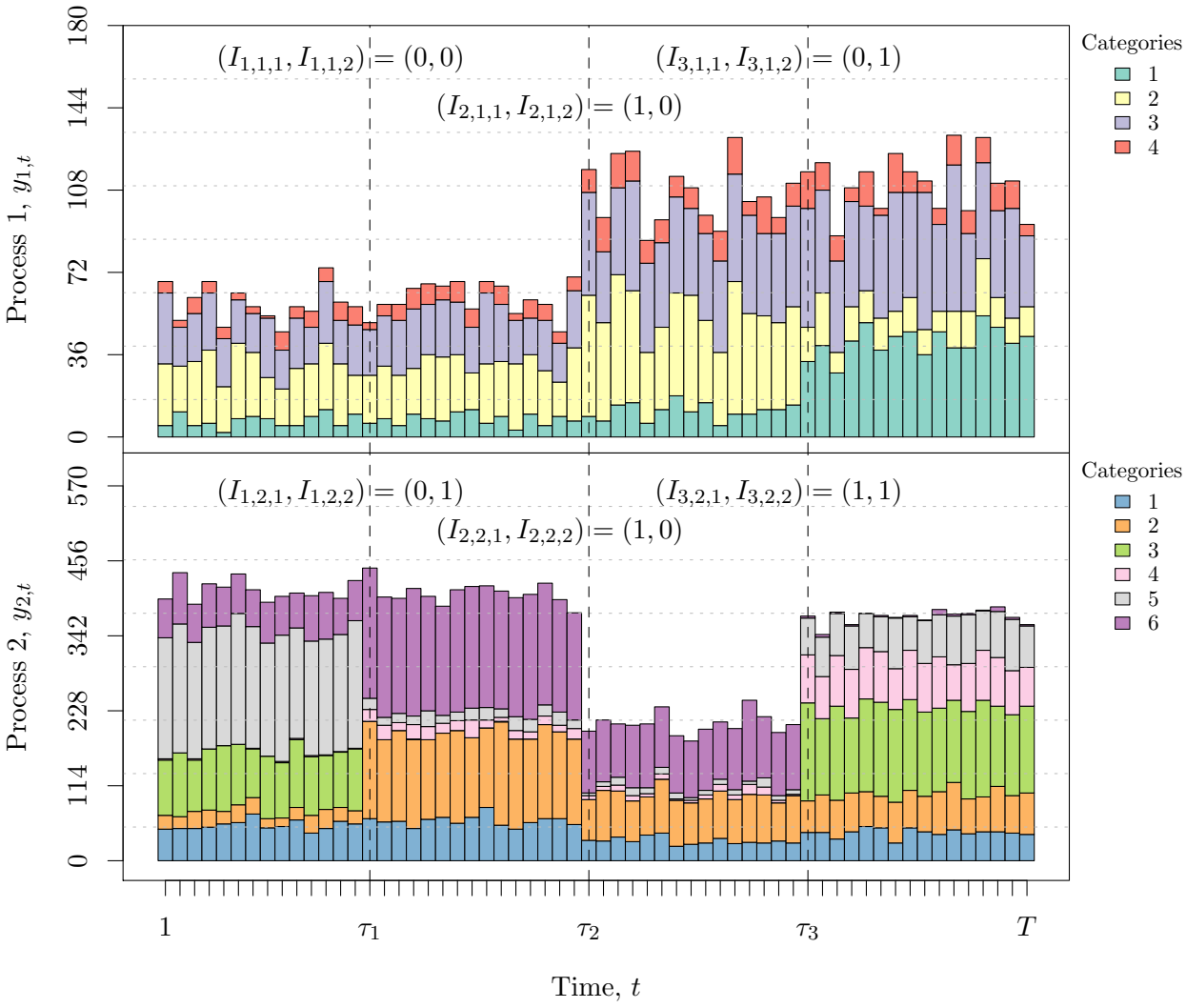

Figure 5.1: Data from a binary marked change point model for discrete time multivariate counting processes

5.1.3 Marginal posterior distribution of the binary marked change points

Similar arguments to those given in the sub-Section 4.1.3 lead to the following expression of the marginal likelihood of the data conditional on k, $\tau_{1:k}$ and $I_{1:k}$

$$\pi(k, \tau_{1:k}, I_{1:k}|\boldsymbol{y}) \propto \pi(k, \tau_{1:k}, I_{1:k}) f(\boldsymbol{y}|k, \tau_{1:k}, I_{1:k}) \tag{5.6}$$

$$= \pi(k, \tau_{1:k}, I_{1:k}) \prod_{\ell=1}^{P} f(y_\ell|k, \tau_{1:k}, I_{1:k}), \tag{5.7}$$

where, for each $\ell = 1, \ldots, P$, we have $f(y_\ell|k, \tau_{1:k}, I_{1:k}) = f(y_\ell|n_\ell, k, \tau_{1:k}, I_{1:k}) f(n_\ell|k, \tau_{1:k}, I_{1:k})$, with

$$f(n_\ell|k, \tau_{1:k}, I_{1:k}) = \left(\prod_{t=1}^{T} \Gamma(n_{\ell,t} + 1) \right)^{-1} \left(\frac{\beta_\ell^{\delta_\ell}}{\Gamma(\delta_\ell)} \right)^{k_{\ell,1}+1} \left(\prod_{j=1}^{k_{\ell,1}+1} \frac{\Gamma(n_\ell^{(1,j)} + \delta_\ell)}{(\beta_\ell + s_{\ell,1,j})^{n_\ell^{(1,j)}+\delta_\ell}} \right), \tag{5.8}$$

where $s_{\ell,1,j} = \tau(\ell, 1, j) - \tau(\ell, 1, j-1)$ is the number of time intervals contained in the j-th segment resulting from $\tau(\ell, 1)$, and $n_\ell^{(1,j)} = \sum_{t=\tau(\ell,1,j-1)}^{\tau(\ell,1,j)-1} n_{\ell,t}$ is the number of observations made on the j-th segment resulting from the effective change points $\tau(\ell, 1)$; and

$$f(y_\ell|n_\ell, k, \tau_{1:k}, I_{1:k}) = \left(\prod_{t=1}^{T} \frac{\Gamma(1 + n_{\ell,t})}{\prod_{r=1}^{m_\ell} \Gamma(1 + y_{\ell,t,r})} \right) \left(\prod_{j=1}^{k_{\ell,2}+1} \frac{\Gamma(\alpha_\bullet^\ell) \prod_{r=1}^{m_\ell} \Gamma(y_{\ell,\cdot,r}^{(2,j)} + \alpha_{\ell,r})}{\Gamma(n_\ell^{(2,j)} + \alpha_\bullet^\ell) \prod_{r=1}^{m_\ell} \Gamma(\alpha_{\ell,r})} \right), \tag{5.9}$$

where $\alpha_\bullet^\ell = \sum_{r=1}^{m_\ell} \alpha_{\ell,r}$ as previously defined in (4.7), and $y_{\ell,\cdot,r}^{(2,j)} = \sum_{t=\tau(\ell,2,j-1)}^{\tau(\ell,2,j)-1} y_{\ell,t,r}$ denotes the number of times the r-th category is observed amongst the $n_\ell^{(2,j)} = \sum_{t=\tau(\ell,2,j-1)}^{\tau(\ell,2,j)-1} n_{\ell,t}$ observations made on the j-th segment resulting from the effective change points $\tau(\ell, 2)$.

5.2 RJMCMC sampling

We discuss how to sample from the posterior distribution of the binary marked change points in the set-up given above. It is an application of the procedure introduced in Bolton and Heard (2018) and described in more details in Bolton (2016).

5.2.1 The procedure

As in Section 4.2, the dimension of the parameter of interest, $(k, \tau_{1:k}, I_{1:k})$, is not fixed, so we consider the Reversible Jump MCMC procedure described in Section 3.3.2. For each model M_k, k indicates the number of change points and the parameter space is $\Phi_k = \mathbb{T}_k \times \mathcal{I}_k$, where for each k, $\mathbb{T}_k$ is the space of the change point positions as defined as in (3.1) and $\mathcal{I}_k = \{0, 1\}^{kP2}$ is the space of the binary marked vectors. In order to explore $\bigcup_k \{k\} \times \mathbb{T}_k \times \mathcal{I}_k$, the support of the marginal posterior distribution of the change points, we consider four types of moves as in Bolton and Heard (2018):

- Birth: The addition of a change point;
- Death: The deletion of one of the change points;

- Shift: A change in the position of one the change points;

- Resample a mark matrix: The resampling of the binary marked matrix corresponding to a randomly chosen change point whose position is not changed;

Given that the latest element of the chain corresponds to k change points, let r_k^b, r_c^d, $r_k^{s_1}$ and $r_k^{s_2}$ denote the probabilities of the birth, death, shift and resampling of mark matrix moves, respectively, such that $r_k^b + r_k^d + r_k^{s_1} + r_k^{s_2} = 1$, for all k and $r_{T-1}^b = r_0^c = r_0^{a_1} = r_0^{a_2} = 0$.

The RJMCMC algorithm given in Algorithm 2 in Section 3.3.2, considered with the four types of moves given above and by noting that $\Phi_k = \mathbb{T}_k \times \mathcal{I}_k$ for all k, yields a Markov chain with stationary distribution $\pi(k, \tau_{1:k}, I_{1:k}|\boldsymbol{y})$. For each type of moves, a description and the acceptance probability corresponding to the lines 7 to 9 of the Algorithm 2 are given below.

5.2.2 The acceptance probabilities

First, we introduce a notation which will be used in the subsection. For all processes $\ell = 1, \ldots, P$, we define the following Bayes' factors

$$B_1^\ell[\boldsymbol{\tau}, \boldsymbol{\tau'}] = \frac{f(n_\ell|k', \boldsymbol{\tau'})}{f(n_\ell|k, \boldsymbol{\tau})} \quad \text{and} \quad B_2^\ell[\boldsymbol{\tau}, \boldsymbol{\tau'}] = \frac{f(y_\ell|n_\ell, k', \boldsymbol{\tau'})}{f(y_\ell|n_\ell, k, \boldsymbol{\tau})}, \tag{5.10}$$

for all change points $\boldsymbol{\tau}$ and $\boldsymbol{\tau'}$ where k and k' denote their respective dimensions.

Suppose we are at the $(t+1)$-th iteration of the algorithm, with $(k, \tau_{1:k}, I_{1:k})$ being the element of the chain at time t.

Birth: Addition of a change point

If a birth move is considered, $(k^*, \tau_{1:k^*}^*)$ is proposed from $(k, \tau_{1:k})$ as described in Section 4.2.2 by adding τ_h' to $\tau_{1:k}$. The difference resides in the necessity to sample I_h', the $P \times 2$ matrix of binary marks associated with the new change point, which is then added to $I_{1:k}$ to obtain $I_{1:k^*}^*$. Following the approach introduced in Bolton and Heard (2018), the columns of I_h' are sampled from the full conditional distribution, q, such that for all $\ell = 1, \ldots, P$ and $i \in \{1, 2\}$

$$I'_{h,\ell,i} \sim \text{Bernoulli}\left(\frac{(\eta + k_{\ell,i})B_i^\ell[\tau(\ell,i), \tau'(\ell,i)]}{(\eta + k_{\ell,i})B_i^\ell[\tau(\ell,i), \tau'(\ell,i)] + (\upsilon + k - k_{\ell,i})}\right), \tag{5.11}$$

where the effective change points $\tau'(\ell, i)$ are obtained by adding τ_h' to the effective change points $\tau(\ell, i) \subseteq \tau_{1:k}$.

The probability of accepting the move is

$$\begin{aligned}\alpha &= 1 \wedge \frac{r_{k^*}^d}{r_k^b}\frac{(T-1-k)}{k^*}\frac{\pi(k^*, \tau_{1:k^*}^*)}{\pi(k, \tau_{1:k})}\frac{\pi(I_{1:k^*}^*|k^*)f(\boldsymbol{y}|k^*, \tau_{1:k^*}^*, I_{1:k^*}^*)}{\pi(I_{1:k}|k)f(\boldsymbol{y}|k, \tau_{1:k}, I_{1:k})}\frac{1}{q(I_h')} \\ &= 1 \wedge \frac{r_{k^*}^d}{r_k^b}\frac{(T-1-k)}{k^*}\frac{p}{1-p}\prod_{\ell=1}^{P}\prod_{i=1}^{2}\frac{(\eta + k_{\ell,i})B_i^\ell[\tau(\ell,i), \tau'(\ell,i)] + \upsilon + k - k_{\ell,i}}{k + \upsilon + \eta}\end{aligned} \tag{5.12}$$

since, with $E_{h,\ell,i} = \mathbb{1}\{I'_{h,\ell,i} = 1\}$, we have

$$\frac{\pi(I^*_{1:k^*}|k^*)f(\boldsymbol{y}|k^*,\tau^*_{1:k^*},I^*_{1:k^*})}{\pi(I_{1:k}|k)f(\boldsymbol{y}|k,\tau_{1:k},I_{1:k})}\frac{1}{q(I'_h)} = \prod_{\ell=1}^{P}\prod_{i=1}^{2}\frac{\pi(I^*_{\cdot,\ell,i}|k^*)}{\pi(I_{\cdot,\ell,i}|k)}\frac{\left(B^\ell_i[\tau(\ell,i),\tau'(\ell,i)]\right)^{E_{h,\ell,i}}}{q(I'_{h,\ell,i})},$$

where for all $\ell = 1,\dots,P$ and $i \in \{1,2\}$, if $I'_{h,\ell,i} = 0$ then via (5.11) and (5.5)

$$q(I'_{h,\ell,i}) = \frac{\upsilon + k - k_{\ell,i}}{(\eta + k_{\ell,i})B^\ell_i[\tau(\ell,i),\tau'(\ell,i)] + \upsilon + k - k_{\ell,i}} \quad \text{and} \quad \frac{\pi(I^*_{\cdot,\ell,i}|k^*)}{\pi(I_{\cdot,\ell,i}|k)} = \frac{\upsilon + k - k_{\ell,i}}{k + \upsilon + \eta},$$

whilst if $I'_{h,\ell,i} = 1$ then

$$q(I'_{h,\ell,i}) = \frac{(\eta + k_{\ell,i})B^\ell_i[\tau(\ell,i),\tau'(\ell,i)]}{(\eta + k_{\ell,i})B^\ell_i[\tau(\ell,i),\tau'(\ell,i)] + \upsilon + k - k_{\ell,i}} \quad \text{and} \quad \frac{\pi(I^*_{\cdot,\ell,i}|k^*)}{\pi(I_{\cdot,\ell,i}|k)} = \frac{\eta + k_{\ell,i}}{k + \upsilon + \eta}.$$

Death move: Deletion of a change point

If a death move is considered, $(k^*, \tau^*_{1:k^*})$ is proposed from $(k, \tau_{1:k})$ as described in Section 4.2.2 by removing τ_h from $\tau_{1:k}$, and by removing I_h, the $P \times 2$ matrix of binary marks corresponding to the change point τ_h, from $I_{1:k}$ to obtain $I^*_{1:k^*}$. The move may be reversed by a birth move from $(k^*, \tau^*_{1:k^*}, I^*_{1:k^*})$ to $(k, \tau_{1:k}, I_{1:k})$. It follows that the probability of accepting the move is

$$\alpha = 1 \wedge \frac{r^b_{k^*}}{r^d_k}\frac{k}{(T-1-k^*)}\frac{\pi(k^*,\tau^*_{1:k^*})}{\pi(k,\tau_{1:k})}\frac{\pi(I^*_{1:k^*}|k^*)f(y|k^*,\tau^*_{1:k^*},I^*_{1:k^*})}{\pi(I_{1:k}|k)f(y|k,\tau_{1:k},I_{1:k})}\frac{q(I_h)}{1},$$

that is,

$$\begin{aligned}\alpha = 1 \wedge &\frac{r^b_{k^*}}{r^d_k}\frac{k}{(T-1-k^*)}\frac{1-p}{p} \\ &\times \prod_{\ell=1}^{P}\prod_{i=1}^{2}\frac{\upsilon + \eta + k^*}{(\eta + k_{\ell,i} - E_{h,\ell,i})B^\ell_i[\tau^*(\ell,i),\tau'(\ell,i)] + (\upsilon + k^* - (k_{\ell,i} - E_{h,\ell,i}))},\end{aligned} \tag{5.13}$$

where $E_{h,\ell,i} = \mathbb{1}\{I_{h,\ell,i} = 1\}$ and the effective change points $\tau'(\ell,i)$ are obtained by adding τ_h to the effective change points $\tau^*(\ell,i)$ of $\tau^*_{1:k^*}$ for all $\ell = 1,\dots,P$ and $i = 1,2$.

Shift move and resampling of a binary mark matrix

If a shift move is proposed, then $k^* = k$, and τ_h, one element of $\tau_{1:k}$, is replaced by τ'_h, an element randomly picked from $\{\tau_{h-1}+1,\dots,\tau_{h+1}-1\}$. The probability of accepting the move follows from previous calculations:

$$\alpha = 1 \wedge \prod_{\ell=1}^{P}\prod_{i=1}^{2}\frac{(\eta + k_{\ell,i} - E_{h,\ell,i})B^\ell_i[\tau^o(\ell,i),\tau''(\ell,i)] + (\upsilon + k - 1 - (k_{\ell,i} - E_{h,\ell,i}))}{(\eta + k_{\ell,i} - E_{h,\ell,i})B^\ell_i[\tau^o(\ell,i),\tau'(\ell,i)] + (\upsilon + k - 1 - (k_{\ell,i} - E_{h,\ell,i}))}, \tag{5.14}$$

where $\tau^o = \tau_{1:k}\backslash\tau_h$, $E_{h,\ell,i} = \mathbb{1}\{I_{h,\ell,i} = 1\}$, and $\tau'(\ell,i)$ and $\tau''(\ell,i)$ are obtained by adding τ_h and τ'_h to $\tau^o(\ell,i)$, respectively, for all $\ell = 1,\dots,P$ and $i = 1,2$. Note that if $\tau_h \in \tau(\ell,i)$, then $\tau^o(\ell,i) = \tau(\ell,i)\backslash\tau_h$ and otherwise $\tau^o(\ell,i) = \tau(\ell,i)$. We acknowledge that it is only after implementing the method and thanks to a comment from one of our supervisors, that we

realised the binary marked matrix associated to the change point, whose position is changed, is not resampled in Bolton and Heard (2018).

Finally, if a binary marked matrix I_h is resampled, since the position of the corresponding change point τ_h is not changed so that τ_h' is τ_h, the probability of accepting the move is 1.

5.3 Simulation studies

Two simulation studies were performed to assess the accuracy of the inference method discussed in this chapter. The first experiment shows that the RJMCMC algorithm presented in Section 5.2 may be used to estimate the binary marked change points of the model defined in Section 5.1.1. The second experiment investigates how drastic the changes must be in order to be detected.

5.3.1 Estimation of binary marked change points

We simulated 50 independent discrete time multivariate counting processes of length $T = 200$ subject to binary marked change points as defined in (5.3). For each simulation, one 5-variate counting process and one 7-variate counting process were generated, that is we considered the case where $P = 2$, $m_1 = 5$ and $m_2 = 7$, using the terminology of Section 5.1.1. For each simulation, the change points $(k, \tau_{1:k})$ were generated from a Bernoulli(8/200) process, and for each process the binary marked vectors were sampled as specified in (5.3) with Bernoulli parameters drawn from a Beta(1,1) distribution. The segment parameters for the Poisson distribution were generated from the Gamma(4, 0.01) and Gamma(4, 0.006) for the first and second process, respectively, whilst the segment parameters for the multinomial distribution were generated from the Dirichlet($\mathbf{1}_5$) and Dirichlet($\mathbf{1}_7$) distributions for the first and second process, respectively. That is, for each simulation, the parameters $(k, \tau_{1:k}, I_{1:k}, \boldsymbol{\psi})$ were generated from the prior distribution given in (5.2) with hyperparameters: $p = 8/200$, $\eta = 1$, $\upsilon = 1$, $\delta_1 = 4$, $\delta_2 = 4$, $\beta_1 = 0.01$, $\beta_2 = 0.006$, $\alpha_1 = \mathbf{1}_5$ and $\alpha_2 = \mathbf{1}_7$. The hyperparameters were chosen to obtain data similar to the data from LANL displayed in Figure 2.2 with a variety of numbers of change points. Across the 50 simulations, the dimension of the effective change points, k^{eff}, varied from 3 to 16, the dimension of the effective change points for the first process, k_1, from 2 to 16 and the dimension of the effective change points for the second process, k_2, from 0 to 16.

For each simulation, a sample from the posterior distribution of the binary marked change points was obtained by 5,000 iterations of the RJMCMC sampler with a burn-in of 1,000 iterations, with the hyperparameters set to their true values and with $k = 0$ as the starting value. We observed that, in this set-up, the number of iterations chosen was large enough for the Markov chains to converge. We computed the MAP estimate for the effective number of change point, k^{eff}, and conditional on this value, the MAP estimate for the binary marked change points was computed.

Figures 5.2 and 5.3 display the results for each estimation. In Figure 5.2, we observe that the estimated dimensions of the effective change points are equal or close to the true dimensions, and when the dimension is correctly estimated, all or most of the effective change points are successfully estimated. Similarly, in Figure 5.3, we observe that for each process

$\ell = 1, 2$, the estimated dimensions of the change points affecting the process ℓ, denoted by k_ℓ, the estimated dimensions of the change points affecting the distribution of n_ℓ, denoted by $k_{\ell,1}$, and the estimated dimensions of the change points affecting the distribution of y_ℓ given n_ℓ, that is $k_{\ell,2}$, are equal or close to the true dimensions. Moreover, when the dimension of the effective change points is correctly estimated, all or most of the effective change points are successfully estimated in most cases. Hence, the simulation study shows that the RJMCMC sampler may be used to successfully infer the binary marked change points.

Figure 5.4 displays the MAP binary marked change points on the data corresponding to one simulation for which $k^{\text{eff}} = 9$, $k_1 = 9$, $k_2 = 4$, $k_{1,2} = 6$, $k_{2,1} = 2$ and $k_{2,2} = 3$ were correctly estimated but $k_{1,1}$ was estimated to be 8 although the true value is 9. The posterior distribution of the effective number of change points is given in Table 5.1, and the trace plots of the effective change points are given in Figure A.1, where we observe the Markov chain converges. We observe in Figure 5.4 that although the estimation detected the change in the distribution of y_1 given n_1 at time $t = 15$, it failed to detect the change in the distribution of n_1 occuring at $t = 15$ because the change is a very small. Although the model of interest assumes the parameters change from one segment to another, since the segment parameters are drawn from the same prior distribution in (4.3), it is possible to obtain two consecutive segment parameters which are very close to each other. In the next study, we investigate how drastic a change must be in order to be detected.

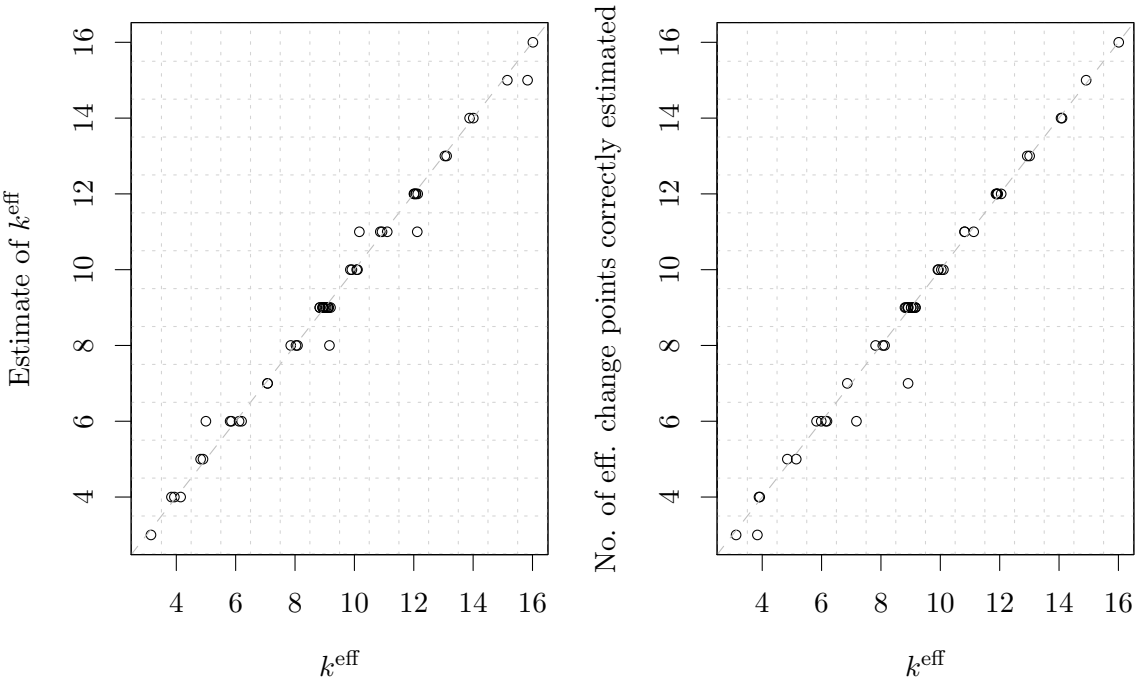

Figure 5.2: Left: Estimation of k^{eff} versus k^{eff} for each simulation. Right: The number of elements of the effective change points correctly estimated versus k^{eff} for each simulation where k^{eff} was correctly estimated. The points are horizontally jittered within the box corresponding to their coordinates.

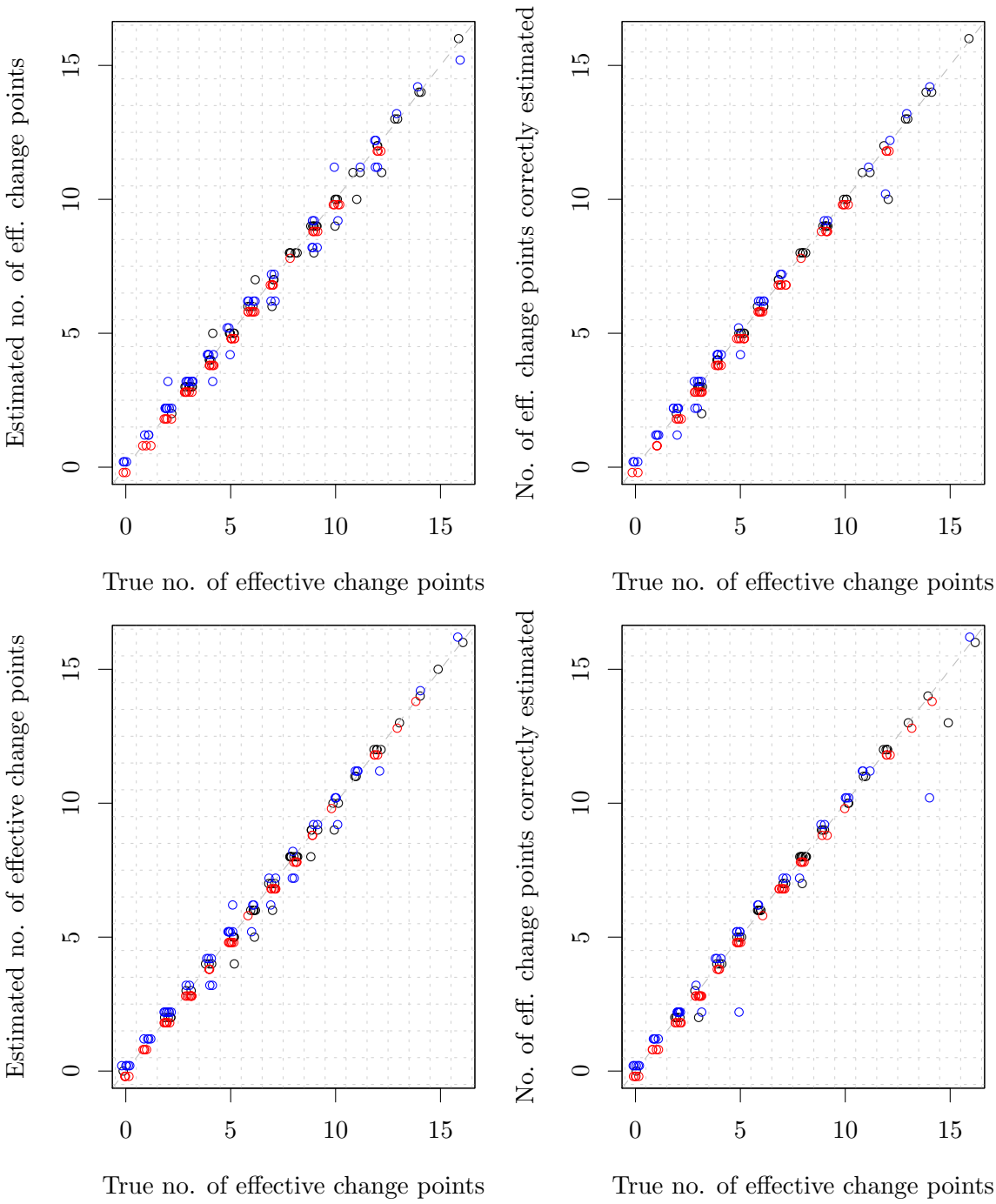

Figure 5.3: Top row: Process $\ell = 1$. Bottom row: Process $\ell = 2$. Left: Estimations of $\boldsymbol{k}_{\boldsymbol{\ell}}$, $k_{\ell,1}$ and $k_{\ell,2}$ versus the true values for each simulation. Right: Number of elements of $\boldsymbol{\tau}(\boldsymbol{\ell})$, $\tau(\ell, 1)$ and $\tau(\ell, 2)$ correctly estimated versus their true dimensions for each simulation where the true dimensions were correctly estimated. The points are horizontally jittered within the box corresponding to their coordinates.

k^{eff}	8	9	10	11	12
$\pi(k^{\text{eff}}\|\boldsymbol{y})$	0.01	0.88	0.10	0.01	0.00

Table 5.1: Posterior distribution of the number of effective change points for one simulation with $k^{\text{eff}} = 9$.

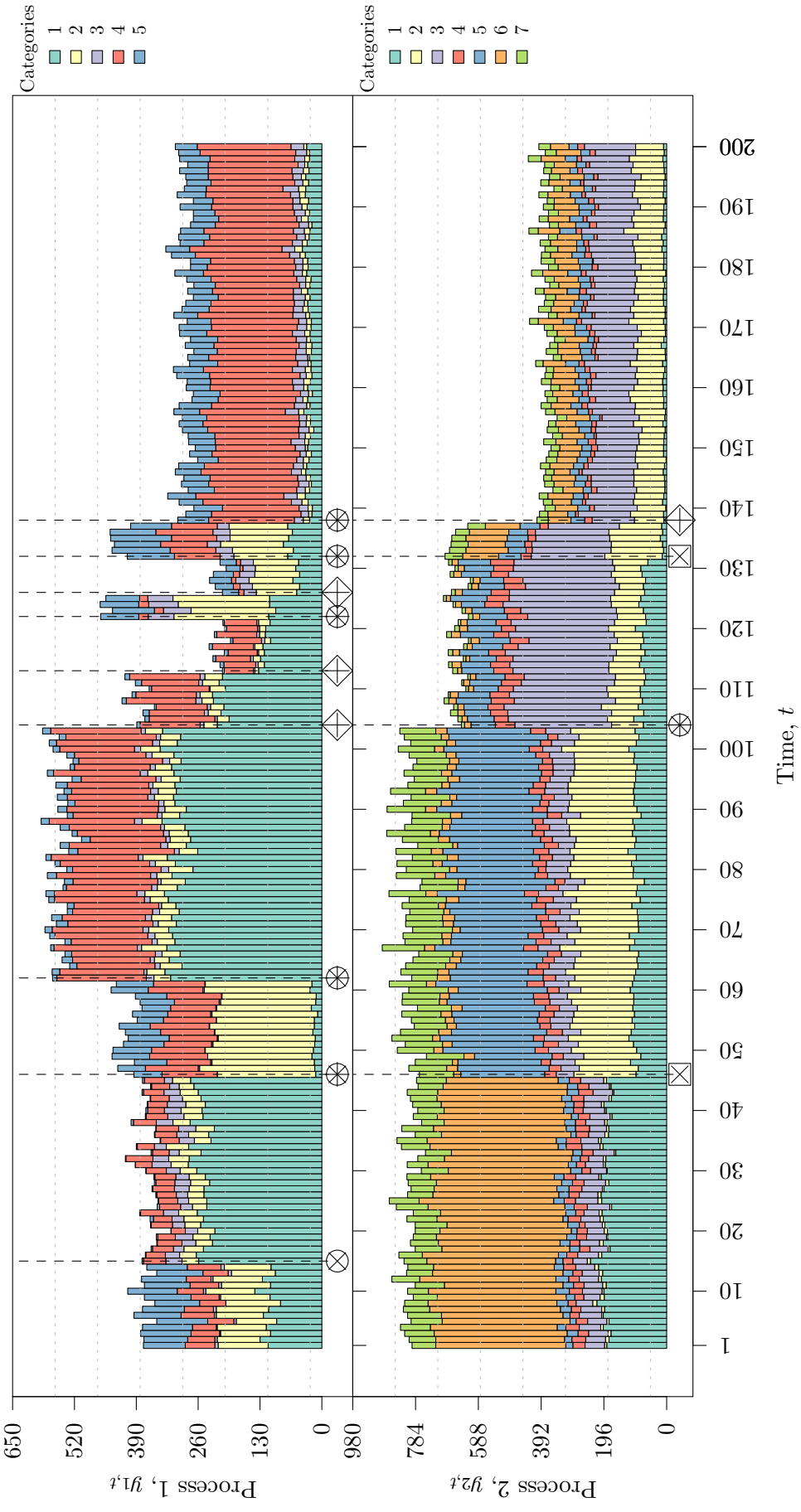

Figure 5.4: Output of the simulation of interest. For each process $\ell = 1, 2$, the true elements of $\tau(\ell, 1) \setminus \tau(\ell, 2)$, $\tau(\ell, 2) \setminus \tau(\ell, 1)$ and $\tau(\ell, 1) \cap \tau(\ell, 2)$ are indicated by $\diamond$, $\square$ and $\bigcirc$, respectively, whilst the estimated elements are indicated by $+$, $\times$ and $\ast$, respectively.

5.3.2 Detection threshold

If a change in the distribution of a process of interest is not drastic enough, it might not be detected as we witnessed in the previous section. In the context of cyber security, where an attack changes the behaviour of the network for a limited amount of time, it is relevant to investigate how drastic and persistent a change must be in order to be detected by our statistical framework. We illustrate how to determine the detection threshold of the methodology.

We restricted ourselves to the scenario where only $P = 1$ multivariate counting process of length $T = 100$ with $m_1 = 5$ categories is subject to $k = k^{\text{eff}} = 2$ effective change points, $\tau_{1:k} = (\tau_1, \tau_2)$, such that $\tau_1 - 1 = T + 1 - \tau_2$, with $\lambda_{1,1} = \lambda_{1,3}$ and $\theta_{1,1} = \theta_{1,3}$. Between the two change points, the behaviour of the process of interest deviates from the base behaviour, which is common to the first and the third segments resulting from the change points. We measure the deviation from the base behaviour by the distance from $\lambda_{1,1}$ to $\lambda_{1,2}$, which we define to be $|\lambda_{1,2} - \lambda_{1,1}|$, the Kullback–Leibler divergence of $\theta_{1,2}$ from $\theta_{1,1}$, denoted by $D_{KL}(\theta_{1,1}||\theta_{1,2})$ which is a measure of how different the probability distribution $\theta_{1,2}$ is from $\theta_{1,1}$, and the duration of the deviation, which we denote by $d = \tau_2 - \tau_1$. Note that the definition of the Kullback–Leibler divergence as well as the expression of $D_{KL}(\theta_{1,1}||\theta_{1,2})$ are given in Appendix A.4.

We considered three experiments. In the first experiment, we investigated the situation where the change points only affect the parameter $\lambda_{1,2}$ so that $\tau(1,1) = (\tau_1, \tau_2)$ and $\tau(1,2) = \emptyset$, by estimating the probability of detecting the change points with respect to $|\lambda_{1,2} - \lambda_{1,1}|$ and d, when $D_{KL}(\theta_{1,1}||\theta_{1,2}) = 0$. In the second experiment, we estimated the probability of detecting the change points with respect to $D_{KL}(\theta_{1,1}||\theta_{1,2})$ and d when $|\lambda_{1,2} - \lambda_{1,1}| = 0$, that is when the change points only affect the parameter $\theta_{1,2}$. Finally, in the third experiment, we investigated whether the detection threshold with respect to $|\lambda_{1,2} - \lambda_{1,1}|$, $D_{KL}(\theta_{1,1}||\theta_{1,2})$ and d is lowered when the change points affect both $\theta_{1,2}$ and $\lambda_{1,2}$.

The value of the parameters for the experiments were chosen as follows. We fixed $\lambda_{1,1}$ to be the mean of the Gamma(δ_1, β_1) distribution with $\delta_1 = 4$ and $\beta_1 = 0.01$, that is $\lambda_{1,1} = 400$. From a number of draws from Gamma(δ_1, β_1), we selected 6 potential values of $\lambda_{1,2}$, denoted by $(\lambda^1, \ldots, \lambda^6)$, such that their distances from $\lambda_{1,1}$ were 20, 30, 40, 50, 60 and 70, respectively. Similarly, we fixed $\theta_{1,1}$ to be the mean of the Dirichlet(α_1) distribution with $\alpha_1 = \mathbf{1}_5$, that is $\theta_{1,1} = (0.2, \ldots, 0.2)$. From a number of draws from Dirichlet($\mathbf{1}_5$), we selected 6 potential values of $\theta_{1,2}$, denoted by $(\theta^1, \ldots, \theta^6)$, such that their Kullback–Leibler divergences from $\theta_{1,1}$ were 0.01, 0.03, 0.09, 0.14, 0.26 and 0.52, respectively. Finally, the following 6 potential values for d, denoted by $(d^1, \ldots, d^6)$, were considered: 2, 8, 14, 20, 26 and 32.

A grid of values for the parameters $(\lambda_{1,2}, \theta_{1,2}, d)$ was built for each experiment: $\{(\lambda^i, \theta_{1,1}, d^j)\}_{i,j=1,\ldots,6}$ for the first experiment; $\{(\lambda_{1,1}, \theta^i, d^j)\}_{i,j=1,\ldots,6}$ for the second experiment and $\{(\lambda^i, \theta^i, d^j)\}_{i,j=1,\ldots,6}$ for the third experiment. For each experiment, 5 simulations[1] were performed for each element of the grid. Figure 5.5 displays two simulations we obtained for the third experiment; the change is hardly visible in the first example, whilst it is clear in the second example that a change occurs, both in the distribution of n_ℓ and in the distribution of y_ℓ given n_ℓ.

For each simulation, the binary marked change points were estimated via a sample obtained by 5,000 iterations of the RJMCMC sampler with a burn-in of 1,000 iterations, with the

[1] More simulations should have been performed to control the Monte Carlo errors but we were constrained by the computational cost of the estimation of the change points.

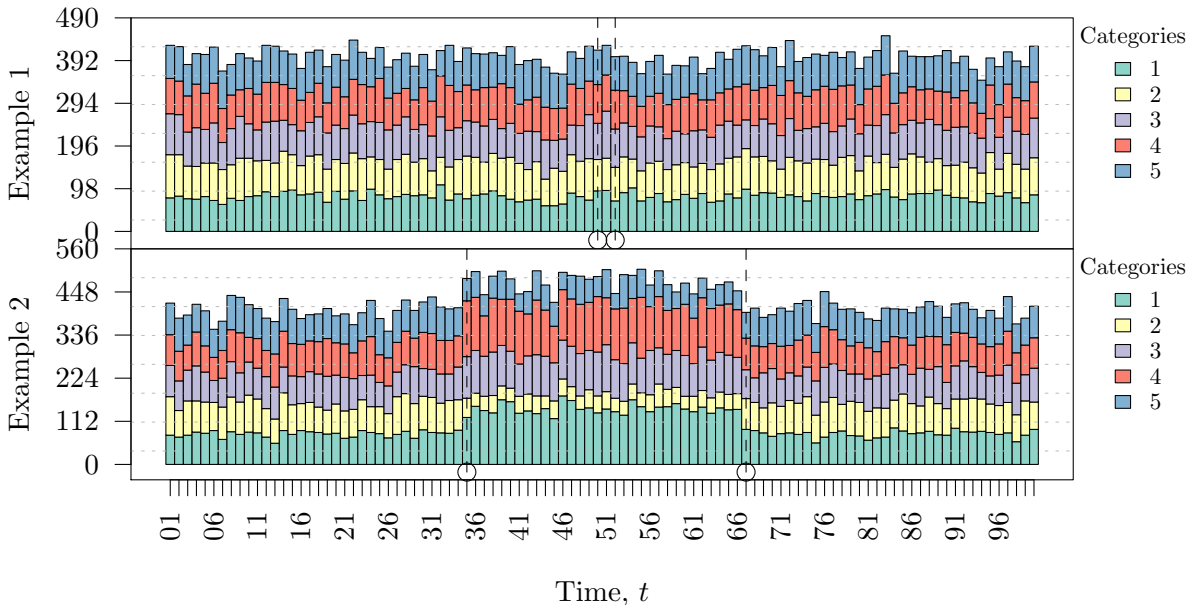

Figure 5.5: Two simulations performed for the third experiment. The change points are indicated by ○. Top: for the first example, $d = 2$, $\lambda_{1,2} - \lambda_{1,2} = 20$, and $D_{KL}(\theta_{1,1}||\theta_{1,2}) = 0.01$. Bottom: for the second example, $d = 32$, $\lambda_{1,2} - \lambda_{1,1} = 70$, and $D_{KL}(\theta_{1,1}||\theta_{1,2}) = 0.52$.

hyperparameters set to their true values and with $k = 0$ as the starting value, as discussed in Section 5.3.1. In the first experiment, each estimation was considered successful if the change points affecting $\lambda_{1,2}$, namely $\tau(1,1)$, were successfully estimated. In the second experiment, each estimation where $\tau(1,2)$ was successfully estimated was considered successful. In the third experiment, in order to compare the results with the first and the second experiments, for each simulation, whether $\tau(1,1)$ was successfully estimated or not and whether $\tau(1,2)$ was successfully estimated or not were considered separately. Finally, for each experiment, the probability of successfully detecting the change generated by each element the grid was estimated to be the proportion of successful estimations.

Results for the first and the second experiments are given in Figure 5.6. For the first experiment, the probability of detecting the change points $\tau(1,1)$ increases with d and $|\lambda_{1,2} - \lambda_{1,1}|$, and there is a threshold, which may be defined in terms of d and $|\lambda_{1,2} - \lambda_{1,1}|$, under which the change is not detected. Similarly, for the second experiment, the probability of detecting the change points $\tau(1,2)$ increases with d and $D_{KL}(\theta_{1,1}||\theta_{1,2})$, and there is a threshold, which may be defined in terms of d and $D_{KL}(\theta_{1,1}||\theta_{1,2})$, under which the change is not detected. Results for the third experiment are given in Figure 5.7. Interestingly, the probabilities of detecting $\tau(1,1)$ and $\tau(1,2)$ show the same trends as in the first experiment and in the second experiment, respectively, but the detection thresholds are lower in each case. More simulations should be performed in order to control the Monte Carlo errors, yet these preliminary results suggest that our statistical framework may be able to detect less drastic changes when the changes affect both the distribution of n_ℓ and the distribution of y_ℓ given n_ℓ.

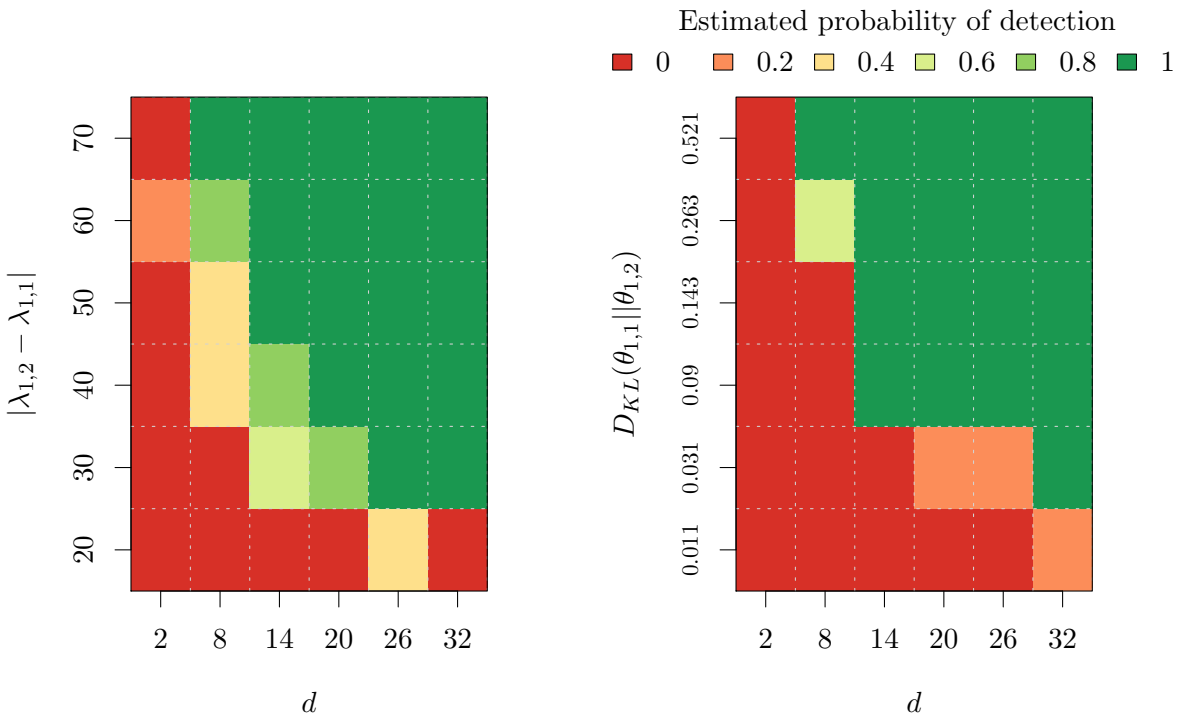

Figure 5.6: Left: Results for the first experiment. Estimated probabilities of successfully estimating $\tau(1,1)$. Right: Results for the second experiment. Estimated probabilities of successfully estimating $\tau(1,2)$.

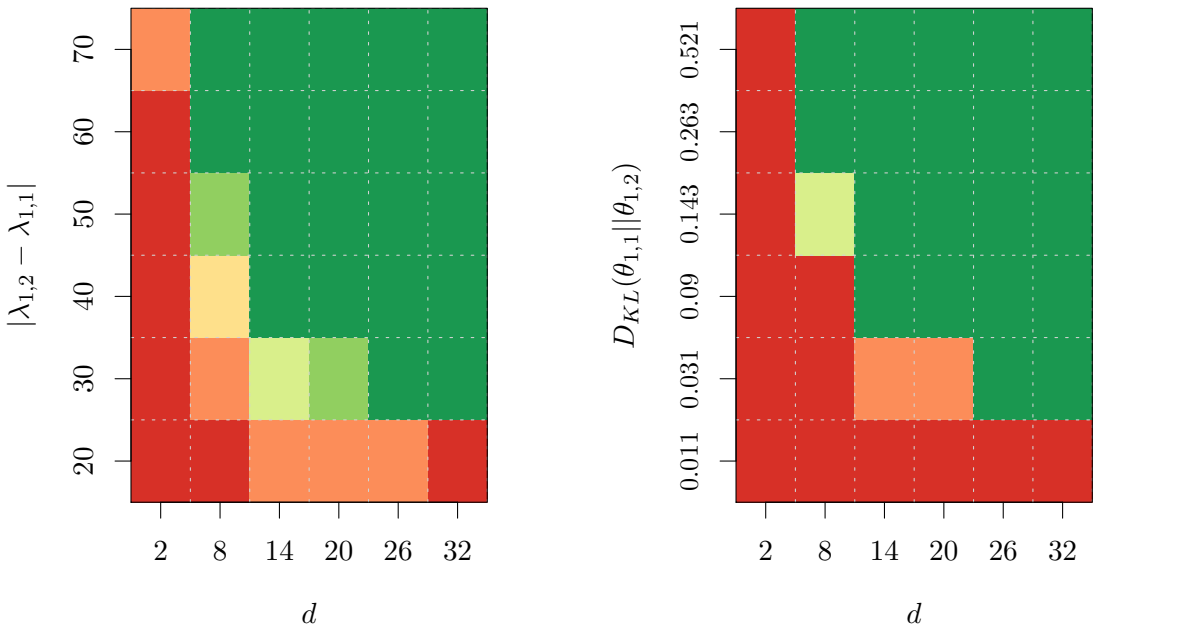

Figure 5.7: Results for the third experiment. Left: estimated probabilities of successfully estimating $\tau(1,1)$. Right: estimated probabilities of successfully estimating $\tau(1,2)$.

Chapter 6

Application to the LANL data

In this chapter, we show that the binary marked change point analysis discussed in Chapter 5 is a pertinent statistical framework to monitor the behaviour of a computer in a network. Firstly, we apply the change point detection method to some data from LANL displayed in Figure 2.2, which represent both the internal activity of a host computer and its communication activity in the network over the course of one week. We discuss how the results of the analysis may be interpreted. Then, we perform some prior sensitivity analysis to illustrate how sensitive the method might be to the choice of the hyperparamaters, but also to discuss how a practioner may encode his prior beliefs to influence the change detection results. Finally, since the data are not labelled for cyber attacks, we modify the data to simulate evidence for a WannaCry ransomware attack on the computer. The binary marked change point model based analysis is performed on the modified data and successfully identifies the evidence for the attack.

6.1 Monitoring the behaviour of a computer in LANL network

In this section, we illustrate how the behaviour of a computer in a network can be monitored thanks to the statistical framework discussed in Chapter 5. Consider the data from LANL displayed in Figure 2.2, which consist of the number of communication events per destination port[1] and of the number of event records per *EventID* for each hour over the course of one week for the host computer *Comp847396*. As explained in Chapter 2, the data are pertinent to simultaneously monitor the network and the internal activity of the computer. Moreover, to detect anomalous behaviours of the computer in the network, it is relevant to seek significant temporal changes in the joint distribution of the two multivariate counting processes. In order to detect the changes and to determine how they affect the two processes, the binary marked change point model for multivariate counting processes described in Chapter 5 is assumed for the data, which we denote by $\boldsymbol{y}$.

Using the notations of Chapter 4 and 5, the data consist of the realisations of $P = 2$ independent processes y_1 and y_2, defined as in Section 4.1.1, where, for each hour t of the $T = 168$ hours of the week of interest, $y_{1,t,r}$ denotes the number of communications made by the computer to the r-th destination port of the $m_1 = 10$ destination ports being monitored, and $y_{2,t,r}$ denotes the number of events recorded on the computer corresponding to the r-th

[1] Only the 10 most recurrent destination ports in the records over the week are monitored.

EventID of the $m_2 = 12$ *EventIDs* observed over the week.

The parameter p of the Bernoulli process assumed a priori on the change points was set to $p = 10/T$ to encode our a priori expectation to observe 10 change points in the week because the behaviour of a computer typically changes at the beginning and at the end of each working day. The prior distribution for the Bernoulli parameter of the prior distribution for each of the binary marked vectors was chosen to be the uninformative Beta(1, 1) distribution; that is, the hyperparameters η and υ defined in (5.3) were set to 1. The prior distributions of the segment parameters of the multinomial distribution were chosen to be the uninformative Dirichlet($\mathbf{1}_{m_1}$) and Dirichlet($\mathbf{1}_{m_2}$) distributions for the first and second process, respectively. Finally, for each process, the parameters of the conjugate prior Gamma distribution for the segment parameters of the Poisson were chosen to be the maximum likelihood estimates on one week of data *preceding* the week of interest, so that the hyperparameters δ_1, δ_2, β_1 and β_2 defined in (5.3) were set to 1.2882, 0.9565, 0.0037 and 0.0093, respectively.

The RJMCMC sampler was run with a burn-in of 2,000 iterations followed by 10,000 iterations to obtain a sample from the posterior distribution of the binary marked change points $(k, \tau_{1:k}, I_{1:k})$. The posterior distribution for the effective number of change points, k^{eff}, is given in Table 6.1, where one observes that the maximum a posteriori (MAP) effective number of change points is 22. The trace plots of the effective change points are given in Figure A.2. The MAP binary marked change points are displayed on the data in Figure 6.1, where we use notations introduced in Section 5.3.1.

We observe in Figure 6.1 that our change detection method successfully detected the changes in the behaviour of the computer at the beginning and at the end of the three working days when the computer was active: day 17, day 18 and day 19. In particular, at the 8th hour of day 17, the method detected the increase in the overall internal activity of the computer, the increase in the overall communication activity of the computer and the change in the relative importance of the destination ports, such as the increase in the relative importance of port 443 versus port 80, which may be associated with the start of human activity at the beginning of the day. Moreover, the four major peaks of activity in the internal activity of the computer occuring around noon on the days 15, 17, 18 and 21 were detected. Interestingly, the change in the relative importance of the *EventIDs* at the peak occuring at noon on day 18 was detected: the sharp increase in the relative importance of the EventID 4625, which indicates an account failed to logon, was flagged. Such a change may be evidence for an intrusion attempt.

k^{eff}	20	21	22	23	24	25	26	27
$\pi(k^{\text{eff}}\|\boldsymbol{y})$	0.00	0.04	0.45	0.27	0.18	0.05	0.01	0.00

Table 6.1: Posterior distribution of the effective number of change points.

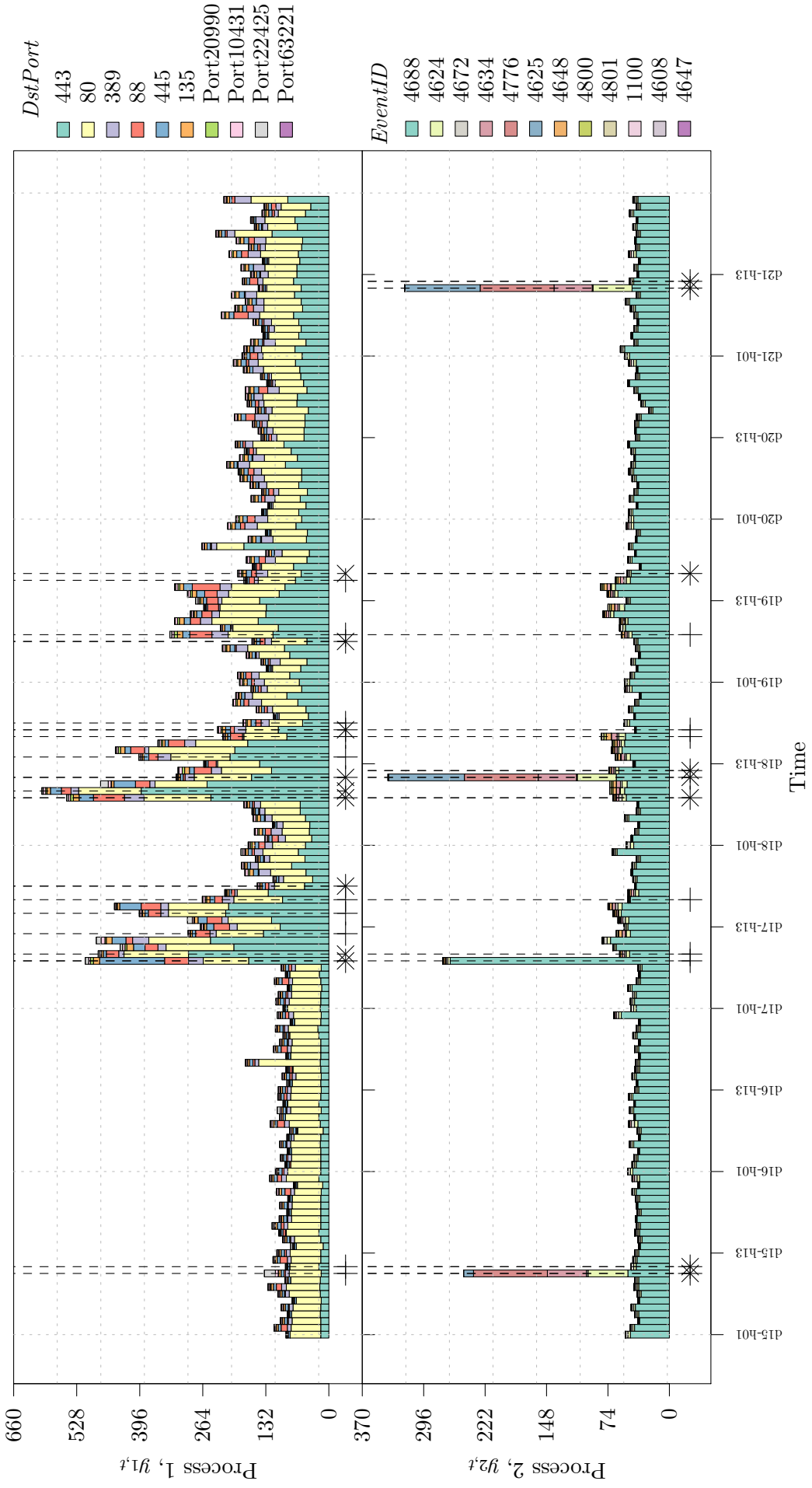

Figure 6.1: Number of communication events per destination port (top) and number of event records per event identifier (bottom) for each hour over the course of one week for the computer *Comp847396* with the estimated binary marked change points. For each process, the elements of estimated effective change points $\tau(\ell, 1)$ and $\tau(\ell, 2)$ are indicated by $+$ and $\times$, respectively, such that the elements of $\tau(\ell, 1) \cap \tau(\ell, 2)$ are indicated by $\ast$.

p	$0.01/T$	$2/T$	$4/T$	$6/T$	$8/T$	$10/T$	$12/T$	$14/T$	$16/T$	$18/T$	$100/T$
MAP k^{eff}	16	21	21	21	22	22	22	24	23	24	32

Table 6.2: MAP number of effective change points for various values of p.

6.2 Prior sensitivity analysis

In order to assess the sensitivity of the analysis to the choice of the hyperparameters, we performed some prior sensitivity analysis. The results of the sensitivity analysis showed that the analysis in Section 6.1 is robust to small variations of the hyperparameters. For the sake of brevity, in this report we focus only on the results we obtained for the influence of the hyperparameter p of the Bernoulli process assumed a priori on the change points, which we defined in (4.3). The hyperparameter p encodes our a priori belief on the number of change points. For example, setting p to $10/T$, as we did in the analysis, suggests we believe a priori that 10 change points affect the process. The analysis of Section 6.1 was performed for a variety of values for p with all the other hyperparameters unchanged. The resulting estimates for the number of effective change points are given in Table 6.2. We first note that the estimations are identical or very similar for values of p in the neighbourhood of the value we chose for the analysis in Section 6.1. Nevertheless, as one would expect, the MAP number of effective change points tends to increase with p, and in particular, setting p to a very low or high value relative to the value we chose, such as $0.01/T$ or $100/T$, leads to a significant increase or decrease in the MAP number of effective change points. Hence, the parameter p may be chosen by the practitioner to control the MAP number of effective change points.

6.3 Detection of a WannaCry ransomware attack

In this section, we show that the binary marked change point analysis discussed in Chapter 5 could detect a WannaCry ransomware attack and allow cyber analysts to investigate the nature of the attack to limit the damage on the network. CERT-EU (2017) gives a technical description of the WannaCry ransomware attack of May 2017, which affected more than 200,000 computers across 150 countries. Prominent organisations such as the NHS were infected. The attack exploited a vulnerability in port 445 of machines running the Microsoft Windows operating system to gain access to the enterprise network via the Internet and then to propagate through the internal network from one vulnerable computer to another. Once a computer was infected, multiple processes were initiated on the computer to encrypt files and to demand ransom payments in exchange for the decryption key.

Consider the set-up described in Section 6.1. We modified the data, which is displayed in Figure 6.1, to simulate evidence for a two hour WannaCry ransomware attack on the computer starting at 8:00 am on the day 20. The modified data are displayed in Figure 6.2, where one can observe that we multiplied by 8 the number of communication events for the destination port 445 and that we multiplied by 2 the number of event records with *EventID* 4688, which corresponds to the start of a process, for the two hours of the attack.

The MAP binary marked change points for the modified data were obtained by repeating the estimation procedure discussed in Section 6.1, and they are displayed on the modified data

k^{eff}	22	23	24	25	26	27	28	29
$\pi(k^{\text{eff}}\|\boldsymbol{y})$	0.04	0.06	0.13	0.18	0.40	0.15	0.04	0.01

Table 6.3: Posterior distribution of the number of effective change points.

in Figure 6.2. The posterior distribution for the number of effective change points, k^{eff}, is given in Table 6.3, where we observe that the maximum a posteriori (MAP) number of effective change points is 26. The trace plots of the effective change points are given in Figure A.3.

In Figure 6.2, we observe that the attack was successfully detected. The change in the overall number of communications, the change in the relative importance of the destination ports caused by the peak in activity of port 445, and the change in the overall number of process activity were detected both at the beginning and at the end of the attack. The output of the analysis flagged the anomaly caused by the attack and provided context on the change in the behaviour of the computer, which would have been helpful for a cyber analyst to understand the nature of the attack.

Finally, we note that using the methodology discussed in sub-Section 5.3.2 one could estimate how drastic the change in the data caused by the WannaCry attack should be in order to be detected.

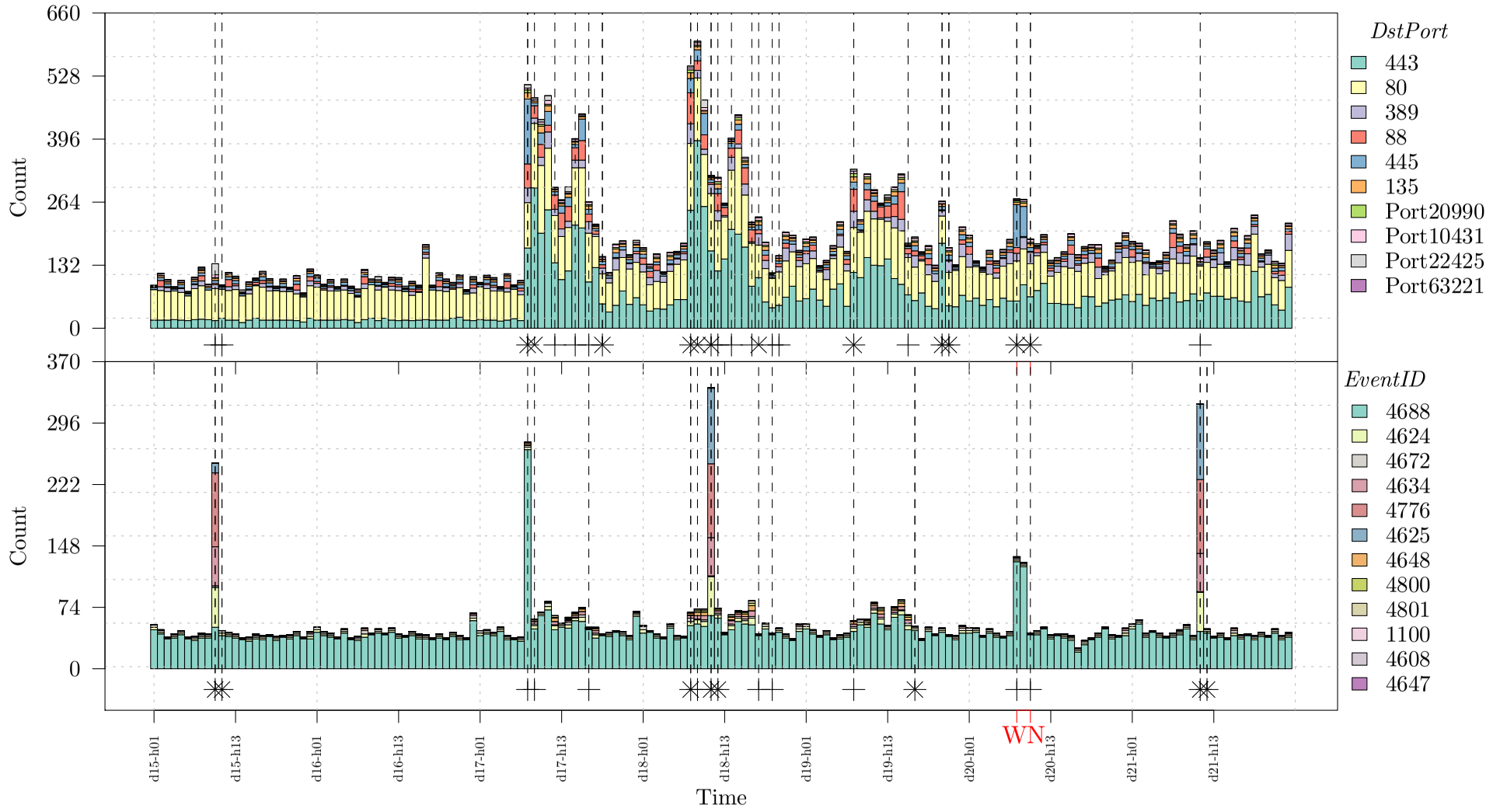

Figure 6.2: Data with evidence for a WannaCry ransomware attack (WN). Number of communication events per destination port (top) and number of event records per event identifier (bottom) for each hour over the course of one week for the computer *Comp847396* with the estimated binary marked change points. For each process, the elements of estimated effective change points $\tau(\ell, 1)$ and $\tau(\ell, 2)$ are indicated by $+$ and $\times$, respectively, such that the elements of $\tau(\ell, 1) \cap \tau(\ell, 2)$ are indicated by $\ast$.

Chapter 7

Conclusion

In this report we proposed a Bayesian model based change point detection method to monitor an enterprise computer network for the purpose of cyber security. By considering data collected from the network environment at LANL (Turcotte et al., 2017), we argued that in the context of cyber security, various important aspects of a computer network may be modelled by discrete time multivariate counting processes, whose joint distribution undergoes changes when an attack occurs. Hence, by detecting change points in the joint distribution of multiple processes, one may detect cyber attacks. We also noted that when a change is detected, it is crucial to provide an explanation for the change, such that cyber analysts can gain situational awareness to quickly limit the impact of the attack, but also distinguish false alerts from attacks. In order to build a statistical method to infer from the data the times when the changes occur, we reviewed the standard Bayesian model based change point analysis framework which requires computational simulation techniques such as Reversible Jump MCMC (Green, 1995). Based on this review, we detailed how to perform a Bayesian model based change point analysis for discrete time multivariate counting processes. We showed via a simulation study that the method successfully infers the number and the positions of the changes. By considering an extension from change points to binary marked change points introduced in Bolton and Heard (2018), the model was then extended to allow the changes to differently affect the multiple processes. As a result, we obtained a statistical framework to detect changes in the behaviour of a computer network from multiple sources, such that the context of each change may be inferred. Via a simulation study, we demonstrated the method may be used to detect an a priori unknown number of change points, their positions and how the processes are affected. We also performed an experiment to determine the detection threshold of the method. Finally, the change point detection method was applied to data from LANL, which represent both the internal activity and the communication activity in the network of a host computer. Interesting changes in the behaviour of the computer were detected. Since the data are not labelled for cyber attacks, we modified the data to simulate evidence for a WannaCry ransomware attack on the computer. We showed that our change detection method successfully detected the attack and provided context on the change in the behaviour of the computer, which would have been helpful for a cyber analyst to limit the damage of the attack on the network.

As future work, we believe it would be interesting to apply the method discussed in this report to other data from LANL in order to further show the method's strengths and limitations.

For example, multiple computers, which would have be chosen to be peers in some sense, could be monitored jointly. Moreover, the detection method we proposed could be refined to detect changes in the behaviour of the computer network as the data arrive. This could be achieved by considering the the Sequential Monte Carlo sampler for change point models described in Bolton (2016).

Appendix A

Appendix

A.1 Trace plots for the RJMCMC of sub-Section 5.3.1

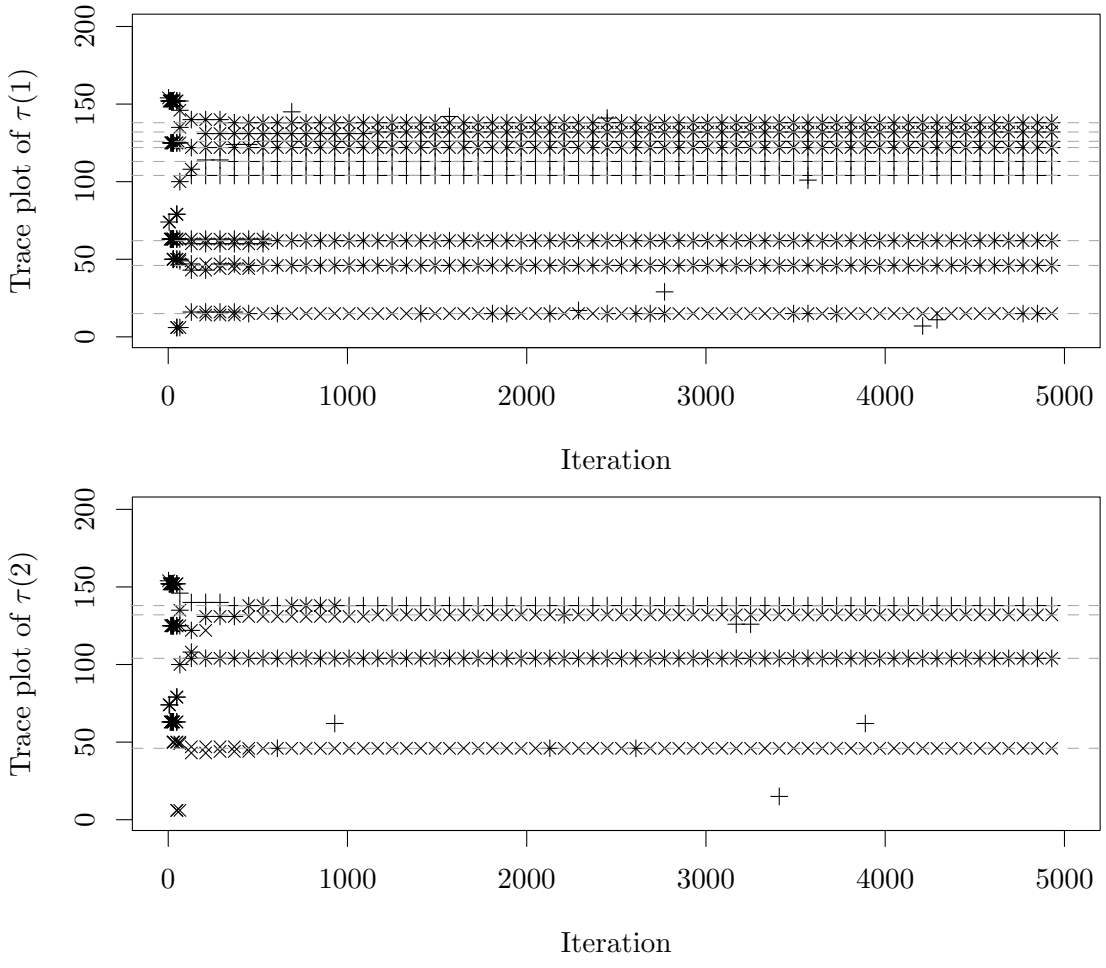

Figure A.1: Top row: Process 1. Bottom row: Process 2. Trace plots of the effective change points in the sample obtained by RJMCMC for the experiment in sub-Section 5.3.1, where the positions of the true change points are shown by horizontal dashed lines. Note that the estimated elements of $\tau(\ell, 1) \setminus \tau(\ell, 2)$, $\tau(\ell, 2) \setminus \tau(\ell, 1)$ and $\tau(\ell, 1) \cap \tau(\ell, 2)$ are indicated by $+$, $\times$ and $\ast$, respectively, for each process.

A.2 Trace plots for the RJMCMC of Section 6.1

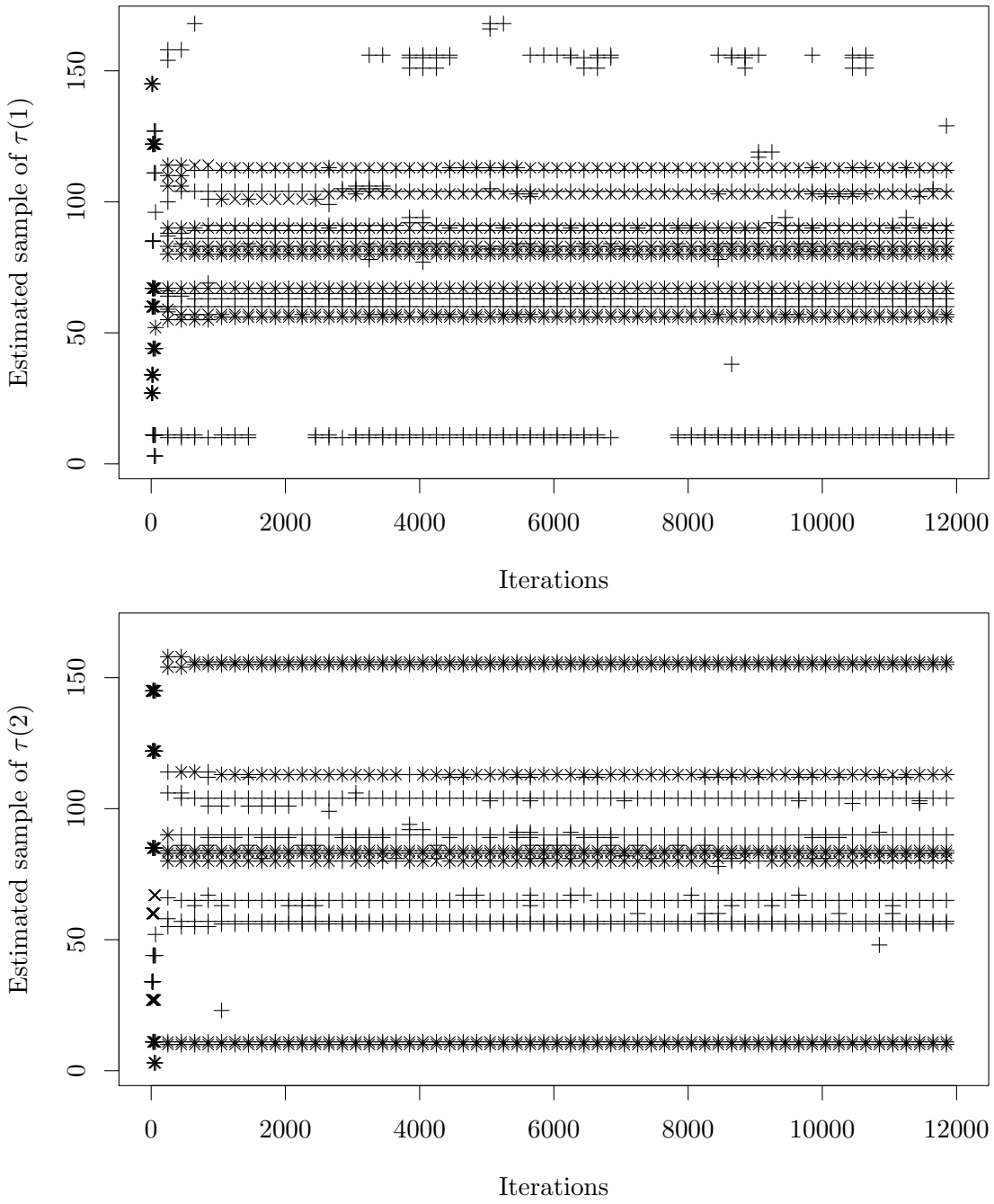

Figure A.2: Top row: Process 1. Bottom row: Process 2. Trace plots of the effective change points in the sample obtained by RJMCMC for the estimation in Section 6.1. Note that the estimated elements of $\tau(\ell,1) \setminus \tau(\ell,2)$, $\tau(\ell,2) \setminus \tau(\ell,1)$ and $\tau(\ell,1) \cap \tau(\ell,2)$ are indicated by $+$, $\times$ and $\ast$, respectively, for each process.

A.3 Trace plots for the RJMCMC of Section 6.3

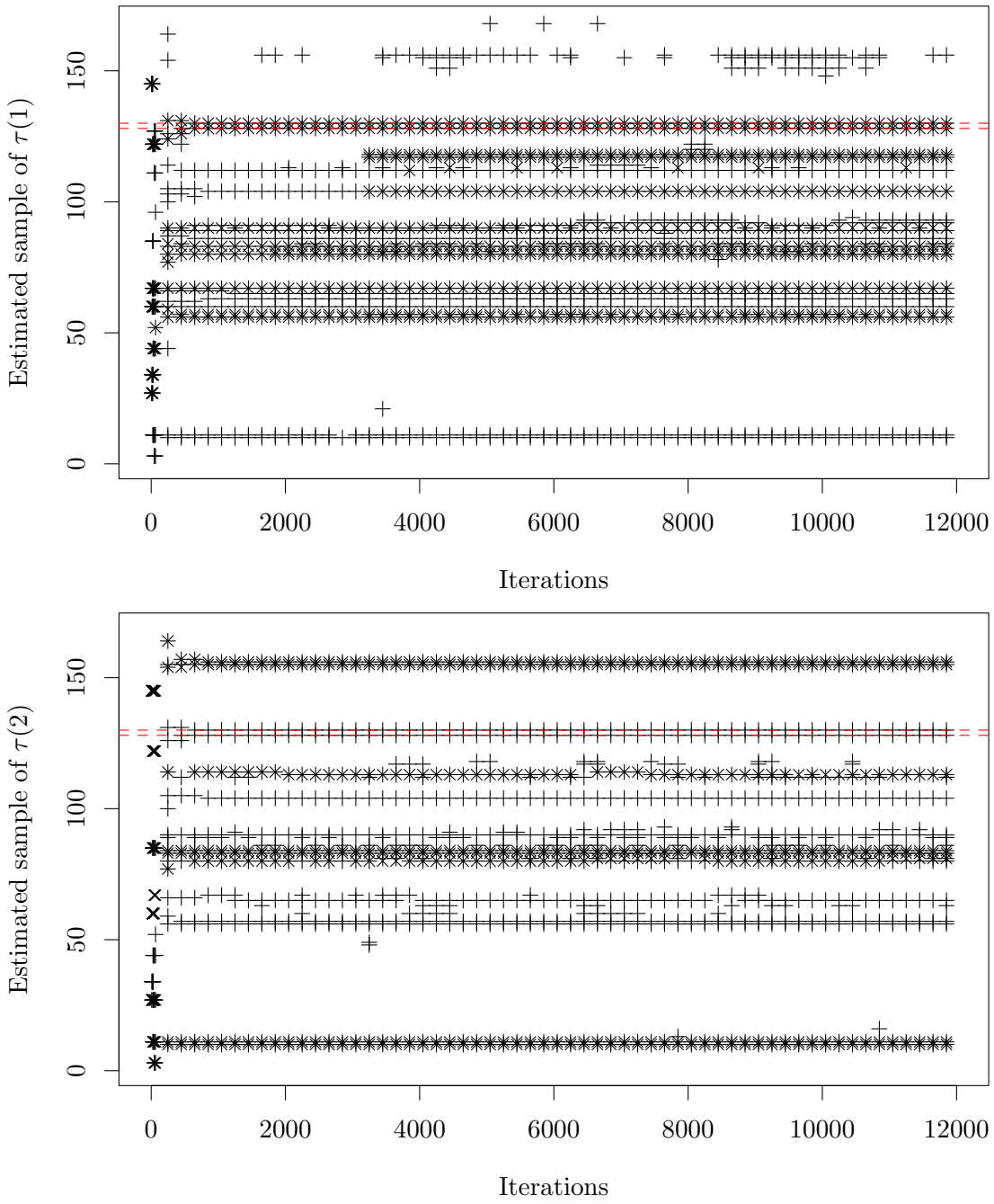

Figure A.3: Top row: Process 1. Bottom row: Process 2. Trace plots of the effective change points in the sample obtained by RJMCMC for the estimation in Section 6.3, where the positions of the change points corresponding to the attack are shown by horizontal red dashed lines. Note that the estimated elements of $\tau(\ell,1) \setminus \tau(\ell,2)$, $\tau(\ell,2) \setminus \tau(\ell,1)$ and $\tau(\ell,1) \cap \tau(\ell,2)$ are indicated by $+$, $\times$ and $\ast$, respectively, for each process.

A.4 Kullback-Leibler divergence

Introduced in Kullback and Leibler (1951), the Kullback-Leibler divergence of the discrete probability distribution Q from the discrete probability distribution P, denoted by $D_{KL}(P||Q)$, is defined to be

$$D_{KL}(P||Q) = \sum_x P(x) \log\left(\frac{P(x)}{Q(x)}\right). \tag{A.1}$$

It is a measure of how different Q is from P but it is not a metric since it is not symmetric. However, it is true that $D_{KL}(P||Q) = 0$ iff P and Q are the same distribution.

In particular, if P is the Dirichlet distribution with parameters $\alpha = (\alpha_1, \ldots, \alpha_K)$ and Q is the Dirichlet distribution with parameters $\beta = (\beta_1, \ldots, \beta_K)$, we have from Kurt (2013) that

$$D_{KL}(P||Q) = \log\left(\frac{\Gamma(\alpha_\bullet)}{\Gamma(\beta_\bullet)}\right) + \sum_{k=1}^{K} \log\left(\frac{\Gamma(\beta_k)}{\Gamma(\alpha_k)}\right) + (\alpha_k - \beta_k)(\psi(\alpha_k) - \psi(\alpha_\bullet)), \tag{A.2}$$

where $\alpha_\bullet = \sum_{k=1}^{K} \alpha_k$, $\beta_\bullet = \sum_{k=1}^{K} \beta_k$ and ψ is the digamma function.

Bibliography

Bolton, A. D. (2016). *Bayesian change point models for regime detection in stochastic processes with applications in cyber security.* PhD thesis, Imperial College London.

Bolton, A. D. and Heard, N. A. (2018). Malware family discovery using Reversible Jump MCMC sampling of regimes. *Journal of the American Statistical Association*, pages 1–13. https://dx.doi.org/10.1080/01621459.2018.1423984.

CERT-EU (2017). WannaCry ransomware campaign exploiting SMB vulnerability. https://cert.europa.eu/static/SecurityAdvisories/2017/CERT-EU-SA2017-012.pdf. Accessed: 2018-08-27.

Claise, B. (2004). Cisco systems NetFlow services export version 9. RFC 3954, Internet Engineering Task Force. https://www.ietf.org/rfc/rfc3954.txt.

Green, P. J. (1995). Reversible Jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82(4):711–732.

Heard, N. and Rubin-Delanchy, P. (2016). Network-wide anomaly detection via the Dirichlet process. In *IEEE Conference on Intelligence and Security Informatics (ISI)*. IEEE. http://dx.doi.org/10.1109/ISI.2016.7745478.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22(1):79–86.

Kurt, B. (2013). Kullback-Leibler divergence between two Dirichlet (and Beta) distributions. http://bariskurt.com/kullback-leibler-divergence-between-two-dirichlet-and-beta-distributions. Accessed: 2018-08-30.

Morgan, M., Sexton, J., Neil, J., Ricciardi, A., and Theimer, J. (2016). Network attacks and the data they affect. In Adams, N. and Heard, N. A., editors, *Dynamic graphs and cyber-security*, chapter 1, page 1–36. Imperial College Press, London.

Neil, J., Hash, C., Brugh, A., Fisk, M., and Storlie, C. B. (2013). Scan statistics for the online detection of locally anomalous subgraphs. *Technometrics*, 55(4):403–414.

Page, E. S. (1954). Continuous inspection schemes. *Biometrika*, 41(1):100–115.

Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, 51(12):3448 – 3470.

Robert, C. P. and Casella, G. (2005). *Monte Carlo Statistical Methods (Springer Texts in Statistics)*. Springer-Verlag New York, Inc., Secausus, NJ, USA.

Shiryaev, A. N. (1963). On optimum methods in quickest detection problems. *Theory of Probability and Its Applications*, 8(1):22–46.

Turcotte, M. (2014). *Anomaly detection in dynamic networks.* PhD thesis, Imperial College London.

Turcotte, M. J. M., Kent, A. D., and Hash, C. (2017). Unified Host and Network Data Set. *ArXiv e-prints*. 1708.07518.

Wang, H., Zhang, D., and Shin, K. (2004). Change-point monitoring for the detection of DoS attacks. *IEEE Transactions on Dependable and Secure Computing*, 1(4):193–208.

World Economic Forum (2018). The global risks report, 13th edition. *World Economic Forum*. http://www3.weforum.org/docs/WEF_GRR18_Report.pdf. Accessed: 2018-08-27.