A Case Study in Robust Quickest Detection for Hidden Markov Models

by

Aliaa Atwi

BEng, American University of Beirut (2008)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Masters of Science in Electrical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

September 2010

Author .....
Department of Electrical Engineering and Computer Science
September 3, 2010

Certified by .....
Munther A. Dahleh
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by .....
Terry P. Orlando
Chairman, Department Committee on Graduate Theses

A Case Study in Robust Quickest Detection for Hidden Markov Models

by

Aliaa Atwi

Submitted to the Department of Electrical Engineering and Computer Science
on September 3, 2010, in partial fulfillment of the
requirements for the degree of
Masters of Science in Electrical Engineering

Abstract

Quickest Detection is the problem of detecting abrupt changes in the statistical behavior of an observed signal in real-time. The literature has focused much attention on the problem for i.i.d. observations. In this thesis, we assess the feasibility of two HMM quickest detection frameworks recently suggested for detecting rare events in a real data set. The first method is a dynamic programming based Bayesian approach, and the second is a non-Bayesian approach based on the cumulative sum algorithm. We discuss implementation considerations for each method and show their performance through simulations for a real data set. In addition, we examine, through simulations, the robustness of the non-Bayesian method when the disruption model is not exactly known but belongs to a known class of models.

Thesis Supervisor: Munther A. Dahleh
Title: Professor of Electrical Engineering and Computer Science

Acknowledgments

First and foremost, I thank God for all the experiences He put in my way, including the ones I was too short-sighted to understand and appreciate until much later.

I would like to thank my advisor Professor Munther Dahleh for giving me the opportunity to join his group, and for being extremely patient and supportive thereafter. I appreciate the lessons I learned from you, both technical and otherwise. Your insight allowed me to experience a very rewarding research journey, much needed for a rookie like me.

I would also like to thank Dr. Ketan Savla, for going above and beyond the call of duty on several occasions to help me with this work. Your feedback and criticism are highly appreciated, and I look forward to more.

My collegues in LIDS have each contributed to making my time more enjoyable. I would especially like to thank Mesrob Ohannessian for supporting me in difficult times and for his willingness to go through philosophical discussions that no one else would agree to have with me. I would also like to thank Parikshit Shah for expanding my foodie horizons and for all the fun times we had together. Yola Katsargiri for being a constant source of positive energy and enthusiasm for me and a lot of people in LIDS. Michael Rinehart for the long enjoyable chats, tons of advice, and the told-you-so's that followed. Giancarlo Baldan for being an awesome office mate, friend and source of encouragement. Sleiman Itani for introducing me to LIDS and being a valuable friend ever since. I would also like to thank Giacomo Como, Rose Faghih, Mitra Osqui, Ermin Wei, AmirAli Ahmadi, Ali Parandegheibi, Soheil Feizi, Arman Rezaee, Ulric Ferner, Noah Stein for the conversations, help and support they have given me.

Finally, I would like to thank the dearest people to my heart, my family. Mum and dad for their unconditional love and countless sacrifices. My grandparents for keeping me in their dawn prayers year after year. And Sidhant, for being my bestfriend and my rock. I dedicate my thesis to all of you.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quickest Detection is the problem of detecting abrupt changes in the statistical behavior of an observed signal in real-time. Designing optimal quickest detection procedures typically involves a tradeoff between two performance criteria; one being a measure of the delay between the actual change point and the time of detection, and the other being a measure of the frequency of false alarms [15].

The literature has focused much attention on the case of i.i.d. observations before and after the change occurs. However, real applications often involve complex structural interdependencies between data points which may be modelled using hidden markov models (HMMs). HMMs have been successfully used in wide array of real applications including speech recognition, economics, digital communications, etc. A recent paper by Dayanik and Goulding provides a Bayesian dynamic programming based framework to study the quickest detection problem when observations are drawn from a hidden Markov model. In addition, Chen and Willett suggested a non-Bayesian framework for studying the problem in [5]. In this thesis, we assess the feasibility of the suggested methods in detecting disruptions in a real data set, consisting of a time series of normalized communication network traffic from Rome.

A recent trend in urban planning is to view cities as cyber-physical systems (CPS) that acquire data from different sources and use it to makes inferences about the states of cities and provide services to their inhabitants. Cell phone data is available at little to no cost to city planners, and can be used to detect emergencies that require a rapid

response. A timely response is crucial in such an application to avert catastrophes, and false alarms may lead to unneeded costly measures. Thus, the problem of using cell phone data to detect abnormal states in a city can be suitably formulated as a quickest detection problem. In addition, the nature of cell phone data as data that reflects periodic human activity indicates that HMMs would provide a more accurate description of the data than i.i.d. alternatives.

An important consideration in our application is robustness when the disruption model is not exactly known. This is a realistic assumption to make about rare events (abnormal states). A recent paper by Unnikrishnan et al [8] addresses this problem for the case of i.i.d. observations. In this thesis, we examine the issue of robustness for the non-bayesian framework suggested in [5] through simulations.

The rest of this thesis is oraginized as follows:

- In Chapter 2, we describe the classical quickest detection problem where the observations before and after the change are i.i.d. samples from two independent distributions. We outline both Bayesian and non-Bayesian formulations often used to describe the optimal tradeoff between detection delay and false alarm. We also describe the minimax robustness criterion in [5] for the i.i.d. case.

- In chapter 3, we present the Bayesian DP-based formulation suggested in [6] for solving the problem when the observations are sampled from HMMs. We describe the optimal solution, which is based on infinite horizon dynamic programming with imperfect state knowledge.

- In chapter 4, we discuss the implementation aspects of the Bayesian DP-based quickest detection algorithm in [6] to detect disruptions in the real data set at hand. More especifically, we describe the data set and the procedure we follow for modeling it with HMMs. We then describe the approximations required to address several computational challenges in the dynamic program describing the optimal solution. Finally, we present the results of our simulations for the Rome data.

- In chapter 5, we present the non-Bayesian formulation suggested in [5] which extends the CUSUM algorithm often used for the i.i.d. case to the case of observations from HMM. We explain the behavior of the suggested procedure through simulations, and proceed to examine the performance of the procedure when the disruption model is not entirely known, but is known to belong to a class of models.

- Finally, in chapter 6, we summarize our findings and suggest directions for future work.

Chapter 2

Quickest Detection for I.I.D. Observations

The problem of detecting abrupt changes in the statistical behavior of an observed signal has been studied since the 1930s in the context of quality control. In more recent years, the problem was studied for a wide variety of applications including climate modeling, econometrics, enviromental and public health, finance, image analysis, medical diagnosis, navigation, network security, neuroscience, remote sensing, fraud detection and counter-terrorism, and even the analysis of historical texts. In many of those applications, the detection is required to occur online (in real time) as soon as possible after it happens. Some examples of situations that require immediate detection are: the monitoring of cardiac patients, detecting the onset of seismic events that precede earthquakes, and detecting security breaches. [15]

This type of problem, known as the "quickest detection problem", will be the focus of this chapter. In particular, we will address the case when the observed data is i.i.d. before the change, and continues to be i.i.d. thereafter under a different model.

2.1 Problem Statement

Consider a sequence $Z_1, Z_2, \ldots$ of random observations, and suppose that there is a change time $T \geq 1$ such that, $Z_1, Z_2, \ldots, Z_{T-1}$ are i.i.d. according to the known

marginal distribution Q_0, and $Z_T, Z_{T+1}, \ldots$ are i.i.d. according to a different known marginal distribution Q_1. For a given T, the sequence $Z_1, Z_2, \cdots, Z_{T-1}$ is independent of $Z_T, Z_{T+1}, \ldots$ Our aim is to detect the change in the model describing the observations as quickly as possible after it happens while minimizing the frequency of false alarms. A sequential change detection procedure is characterized by a stopping time τ with respect to the observation sequence. Essentially, τ is a random variable defined on the space of observation sequences, where $\tau = k$ is equivalent to deciding that the change time T for a certain sequence has occurred at or before time k. We denote by Δ the set all of allowable stopping times.

2.1.1 Non-Bayesian Formulations

In Non-Bayesian quickest detection, the change time T is taken to be a fixed but unknown quantity. Several formulations have been proposed for the optimal tradeoff between false alarm and detection delay.

The minimax formulation suggested by Lorden in [10] is based on minimizing the worst-case detection delay (over all possible change points T and all possible realizations of the pre-change sequence) subject to a false alarm constraint. The worst-case delay is defined as,

$$WDD(\tau) = \sup_{T \geq 1} \operatorname{ess\,sup} E_T^Q[(\tau - T + 1)^+ | Z_1, Z_2, \ldots, Z_T]$$

where E_T^Q refers to the expectation operator when the change happens at T and the pre-change and post-change sequences are sampled from Q_0 and Q_1 respectively. The false alarm rate is defined as,

$$FAR(\tau) = \frac{1}{E_\infty^Q[\tau]}$$

where $E_\infty^Q[\tau]$ can be interpreted as the mean time to false alarm. The objective can then be expressed as:

$$\min WDD(\tau) \text{ s.t. } FAR(\tau) \leq \alpha \tag{2.1}$$

The optimal solution to [10] was shown by Moustakides [12] to be Page's decision rule [13]:

$$\tau_c = \inf\{n \geq 1 : \max_{1 \leq k \leq n} \sum_{i=k}^{n} L^Q(Zi) \geq \beta\} \tag{2.2}$$

where L^Q is the log-likelihood ratio between Q_1 and Q_0 and β is chosen such that, $E_\infty^Q(Z_i) \geq \frac{1}{\alpha}$.

Another formulation was suggested by Pollak in [14]. Worst-case average delay was used instead of WDD(τ) to quantify delay.

2.1.2 Bayesian Formulations

In Bayesian quickest detection, the change time T is assumed to be a random variable with a known prior distribution. Let C_D be a constant describing the cost for each observation we take past the change time T, and C_F be the cost of false alarm. The performance measures are the average detection delay:

$$ADD(\tau) = E[(\tau - T)^+]$$

where expectation is over τ and and all possible observation sequences. The probability of false alarm:

$$PFA(\tau) = P(\tau < T)$$

The objective is to seek $\tau \in \Delta$ that solves the optimization problem:

$$\inf_{\tau \in \Delta} ADD(\tau) \text{ s.t } PFA(\tau) \leq \alpha$$

Equivalently, the optimization problem can be expressed as:

$$\inf_{\tau \in \Delta} C_F.PFA(\tau) + C_D.ADD(\tau)$$

where $C = C_F/C_D$ is chosen to guarantee $PFA(\tau) \leq \alpha$.

Taking the prior on T to be geometric, the optimal policy is given by Shiryaev's

test[18]??:

$$\tau_{opt} = \inf\{k \geq 0 \mid \pi_k \geq \pi^*\}$$

where π_k is the posterior probability that $T \leq k$ given the observation sequence up to time k, and π^* is a threshold that depends on the ratio $C = \frac{C_D}{C_F}$. When $C = 1$, $\pi^* = 0$.

2.2 Robustness

The policies outlined above are optimal assuming that we have exact knowledge of Q_0 and Q_1. Several applications, however, involve imperfect knowledge of these distributions. [8] provides robust versions of the quickest detection problems above when the pre-change and post-change distributions are not known exactly but belong to known uncertainty classes of distributions P_0 and P_1. For the Bayesian criterion, the version of the (minimax) robust problem suggested in [8] is:

$$\min \sup_{p_0 \in P_0, p_1 \in P_1} E\{(\tau - T + 1)^+\}$$

$$\text{s.t.} \sup_{p_0 \in P_0} \Pr(\tau < T) \leq \gamma$$

For uncertaintly classes that satisfy specific conditions, least favorable distributions (LFDs) can be identified such that the solution to the robust problem is the same as the solution for the non-robust problem designed for the LFDs. To describe the conditions, we need to introduce the notion of "joint stochastic boundedness".

Notation: $\mu \succ \mu'$ means that if $X \sim \mu_0$ and $X' \sim \mu_1$, then $\Pr(X \geq m) \geq \Pr(X' \geq m)$, for all real m.

Definition (Joint Stochastic Boundedness): Let $(\overline{v_0}, \underline{v_1}) \in Q_0 \times Q_1$ be a pair of distributions such that $\underline{v_1}$ is absolutely continuous with respect to $\overline{v_0}$. Let L^* denote the log likelihood ratio between $\underline{v_1}$ and $\overline{v_0}$. For each $v_j \in Pj$, let μ_j denote the distribution of $L^*(X)$ when $X \sim v_j$, $j = 0, 1$. $\overline{\mu_0}$ and $\underline{\mu_1}$ denote the distribution of $L^*(X)$ when $X \sim \overline{v_0}$ and $X \sim \underline{v_1}$ repectively. The pair (Q_0, Q_1) is said to be jointly

stochastically bounded by $(\overline{v_0}, \underline{v_1})$ if for all (v_0, v_1) in (Q_0, Q_1), $\overline{\mu_0} \succ \mu_0$ and $\mu_1 \prec \underline{\mu_1}$.

Under certain assumptions on Q_0 and Q_1, the pair $(\overline{v_0}, \underline{v_1})$ are LFDs for the robust quickest detection problem. Loosely speaking, the LFD from one uncertainty class is the distribution that is "nearest" to the other uncertainty class.

The conditions on Q_0 and Q_1 for the Bayesian robust quickest detection problem are:

- Q_0 contains only one distribution v_0, and the pair (Q_0, Q_1) is jointly stochastically bounded by $(v_0, \underline{v_1})$.
- The prior distribution of T is geometric.
- $L^*(.)$ is continuous over the support of v_0.

Chapter 3

DP-Based Bayesian Quickest Detection for Observations Drawn from a Hidden Markov Model

3.1 Setup and Problem Statement

Consider a finite-state Markov chain $M = \{M_t; t \geq 1\}$ with d states, and suppose that the initial state distribution and the one-step transition matrix of M change suddenly at some unobservable random time T. Conditioned on the change time, M is time-homogenous before T with initial state distribution μ and one-step transition matrix W_0 and is time-homogenous thereafter with initial distribution ρ and one-step transition matrix W_1.

The change time T is assumed to have a zero-modified geometric distribution with parameters θ_0 and θ, meaning that

$$T = \left\{ \begin{array}{ll} 0, & \text{w.p. } \theta_0 \\ t, & \text{w.p. } (1-\theta_0)(1-\theta)^{t-1}\theta \end{array} \right\}$$

Let the process $X = \{X_t; t \geq 1\}$ denote a sequence of noisy observations of M. The probability distribution of X_t is a function of the current state M_t and whether

or not the change has occured by time t. For instance, X_t can be assumed to have a poisson distribution with parameter λ_{ij}, where $i \in 1, ..., d$ refers to the value of M_t, and $j = 1_{\{t \geq T\}}$ is 0 before the change and 1 thereafter.

We would like to use the noisy observation sequence X to detect the change in the underlying unobservable sequence M as soon as possible while minimizing false alarms.

The schemes in chapter 2 were developed assuming independence between observations conditioned on the change time. The Markov assumption of the setup in this chapter necessitates a different approach for solving the problem.

3.2 Bayesian Framework

The framework presented in this section was presented in [6]. It proceeds as follows:

Define the process Y by $Y_t = \left(M_t, 1_{\{t \geq T\}}\right)$ for $t \geq 1$. $Y_t = (d, 0)$ has the interpretation that $M_t = d$ and the change has not occured yet $(t < T)$. Similarly, $Y_t = (d, 1)$ means that the change has occured and $M_t = d$. The state space of the process is

$$\mathcal{Y} = \{(0,0), (1,0), ..., (d,0), (0,1), (1,1), ..., (d,1)\}$$

and is partitioned into

$$\mathcal{Y}_0 = \{(0,0), (1,0)..., (d,0)\} \text{ and } \mathcal{Y}_1 = \{(0,1), (1,1), ..., (d,1)\}$$

First, we observe that Y is a Markov process with initial distribution $\eta = ((1-\theta_0)\mu, \theta_0\rho)$ and one-step transition matrix $P = \begin{bmatrix} (1-\theta)W_0 & \theta W_1 \\ 0 & W_1 \end{bmatrix}$. We can also see that Y_1 forms a recurrent class, and the states in Y_0 are transient. Therefore, the change time T of M is the time till absorption of Y in $\mathcal{Y}_1$:

$$T = \min\{t \geq 1; Y_t \notin \mathcal{Y}_0\}$$

The quickest detection problem for HMMs thus reduces to the problem of using noisy observation to detect absorption in an unobservable Markov chain as quickly as possible with a false alarm constraint.

The desire to detect changes quickly is reflected in a cost a paid for every observation taken after T without detecting a change. False alarms are penalized by a cost b for declaring a change before T. [1] The Bayes' risk associated with a certain decision rule τ is thus:

$$\mu(\tau) = a.E\left[(\tau - T)^+\right] + b.\Pr(\tau < T) \tag{3.1}$$

where expectation is taken over all possible sequences X and Y and all change times T.

The objective is to solve the following optimization problem:

$$\inf_{\tau \in \Delta} \mu(\tau) \tag{3.2}$$

3.3 Solution

For every $t \geq 0$, let $\Pi_t = (\Pi_t(y), y \in \mathcal{Y})$ be the row vector of posterior probabilities

$$\Pi_t(y) = \Pr\{Y_t = y \mid X_1, X_2, ...X_t\},\ y \in \mathcal{Y}$$

that the Markov chain Y is in state $y \in \mathcal{Y}$ at time t given the history of the observation process X.

The process $\{\Pi_t, t \geq 0\}$ is a Markov process on the probability simplex state space $\mathcal{P} = \{\pi \in [0,1]^{|Y|}; \sum_{y \in Y} \pi(y) = 1\}$ with

$$\Pi_{t+1} = \frac{\Pi_t P diag(f(X_{t+1}))}{\Pi_t P f(X_{t+1})} \tag{3.3}$$

where $f(X_{t+1})$ is the row vector of emission probabilities of X_{t+1} under $y \in \mathcal{Y}$, and

[1] The framework suggested in [6] allows different states of Y to be associated with different costs for delay and false alarm; however, in this thesis, we assume the cost to be uniform across all states.

$diag(f(X_{t+1}))$ is the diagonal matrix formed using the elements of $f(X_{t+1})$.

Let $g(\pi)$ be the expected delay cost for the current sample over all possible underlying $y \in \mathcal{Y}$ given the past. Similarly, define $h(\pi)$ as the expected cost of false alarm over $y \in \mathcal{Y}$ given our knowledge of the past if change is declared at the current sample. The two quantities are given by:

$$g(\pi) = \sum_{y \in \mathcal{Y}_\infty} a\pi(y)$$

$$h(\pi) = \sum_{y \in Y_0} b\pi(y)$$

The Bayes' risk can then be expressed as

$$\mu(\tau) = E\left[\sum_{t=0}^{\tau-1} g(\Pi_t) + h(\Pi(\tau))\right], \text{ for all } \tau \in \Delta$$

where the expectation is taken over all possible observation sequences X and underlying markov chains Y.

[6] suggests that the optimal cost of the problem (3.2) is a function of η, the initial state probability distribution of Y. More especifically,

$$\mu^* = \inf_{\tau \in \Delta} \mu(\tau) = J(\eta)$$

where

$$J(\pi) = \inf_{\tau} E_\pi \left[\sum_{t=0}^{\tau-1} g(\Pi_t + h(\Pi_\tau)\right]$$

$J(\pi)$ is the value function of an optimal stopping problem over the Markov process Π, and E_π is the expected value over all sequences X and Y given that $\Pi_0 = \pi$.

[6] proceeds to prove that the value function satisfies the following Bellman equation:

$$J(\pi) = \min\{h(\pi), g(\pi) + E\left[J(\pi') \mid \pi\right]\} \tag{3.4}$$

where π' is obtained from π through the update equation (3.3), and expectation is taken over Π'. This will be explained in more detail in the next subsection.

3.3.1 An Infinite Horizon Dynamic Program with Imperfect State Information

Equation (3.4) captures the tradeoff between immediate and future costs at the heart of dynamic programming(DP). Dynamic programming is concerned with decisions made in stages, where each decision poses an immediate cost, and influences the context in which future decisions are made in way that can be predicted to some extent in the present. The goal is to find decision making policies that minimize the cost incurred over a number of stages. In DP formulations, a discrete-time dynamic system with current state i is assumed to transition to state j with probability $p_{ij}(u)$ dependent on the control u. A transition from i to j under u poses a cost $pc(i,u,j)$ in the present and a cost $J^*(j)$ in the future, where $J^*(j)$ is refered to as the optimal cost-to-go of state j over all remaining stages. The costs to go can be shown to satisfy a form of Bellman's equation:

$$J^*(i) = \min_u E\left[pc(i,u,j) + J^*(j) \mid i,u\right],, \text{ for all i,} \tag{3.5}$$

where the expectation is taken over j.[4]

The DP formulation just described assumes perfect knowledge of the system's states. Our setup, however, grants us access only to noisy observations of the states Y_t, making this a DP problem with imperfect state information. [3] shows that this type of problems can be reduced to a perfect-state-knowledge DP where the state space consists of the partial information available at every time. In our application, the partial information at time t is the vector of posterior probabilities π_t obtained from the observation sequence $X_1, X_2, ...X_t$ and the initial state distribution π_0.

Equation (3.4) is clearly a DP Bellman equation of the same form as (3.5) where the state space is the 2d-dimensional simplex $\mathcal{P}$. The decision u is either to"declare a change and stop" or "continue sampling". If we choose to continue sampling, we incure a cost $pc(\pi, continue, \pi') = g(\pi)$ in the present representing the expected current sample cost (over the possible underlying states of Y given the history so far). In addition we incur $E\left[J(\pi') \mid \pi\right]$ in the future where expectation is taken over

the next posterior Π' (a function of the next random noisy observation). On the other hand, if we choose to declare a change, we incur a cost $pc(\pi, stop, \pi') = h(\pi)$ in the present representing the expected cost of false alarm, and no future costs.

The Bellman equation (3.4) has an infinite horizon making it computationally infeasible. An approximate solution can be obtained by solving the problem for only $N(\varepsilon)$ stages, where ε corresponds to a tolerable error margin. Let $\mu_{N(\varepsilon)}$ be the optimal cost for the finite horizon problem with $N(\varepsilon)$ stages. [6] indicates that for every positive ε, there exists $N(\varepsilon)$ such that $\mu_{N(\varepsilon)} - \mu^* \leq \varepsilon$. More especifically,

$$N(\varepsilon) = \left\lceil \frac{b}{\varepsilon} \left(\frac{b}{a} + \sum_{y,y' \in Y_0} I - (1-\theta) {W_0}^{-1}(y, y') \right) \right\rceil$$

This expression is consistent with the intuition that higher false alarm costs, lower sampling costs, higher precision, and higher expected time of change value each call for a longer horizon. $\mu_{N(\varepsilon)}$ is the N-stage value function evaluated at $\pi_0 = \eta$. The N-stage finite horizon value function for a starting posterior probability π_t can be obtained through the following recursion:

$$J^{k+1}(\pi_t) = \min \left\{ h(\pi_t), g(\pi_t) + E_{X_{t+1}} \left(J^k(\Pi_{t+1}) \mid \Pi_t = \pi_t \right) \right\} \tag{3.6}$$

where $j = 1, 2, \cdots, N(\varepsilon)$ and $J^0 := h$, and π_{t+1} can easily be calculated from π_t for a given value of X_{i+1} using (3.3). Simply stated, this equation uses the optimal cost-to-go calculated for a horizon k to calculate the optimal cost-to-go for a horizon $k+1$.

3.3.2 ε-Optimal Alarm Time and Algorithm

We recap that the problem is to detect a change in the underlying Markov model through observing noisy samples of it. Every sample past the change point where the change goes undetected poses a cost a, and false alarms pose a cost b. The algorithm for detecting the change suggested in [6] is as follows:

Starting with observation X_1 and $\pi_0 = \eta$, calculate the updated posterior π_1 from π_0 and X_1 using (3.3). If π_1 belongs to the region $\Gamma_{N(\varepsilon)} = \{\pi \in \mathcal{P}; J^{N(\epsilon)}(\pi) = h(\pi)\}$, then declare that a change has occured and stop sampling. Otherwise, repeat for $t = 2$ by sampling X_2, and calculating π_2 from π_1 and X_2, and so on.

The region $\Gamma_{N(\varepsilon)}$ is calculated offline only once for a set of model parameters and costs. [6] shows that the region is a non-empty closed convex subset of the 2d-dimensional simplex $\mathcal{P}$ that decreases with increasing $N(\epsilon)$, and converges to Γ (the infinite horizon optimal stopping region) as $N \to \infty$. Calculating this region in practice poses problems that will be addressed in the next chapter.

Chapter 4

Implementation of the Bayesian DP-based HMM Quickest Detection Algorithm for Detecting Rare Events on a Real Data Set

In this chapter, we propose a framework for detecting distruptions in real correlated data sequences based on the HMM quickest detection algorithm described in chapter 3. We proceed to test the framework on a real life data set representing cell phone traffic in Rome.

4.1 Description of Data Set

A recent trend in urban planning is to view cities as cyber-physical systems that acquire data from different sources and use it to make inferences about the states of cities and provide services for their inhabitants. Wireless network traffic data is available at little to no cost to city planners [7], and can be used to detect emergencies that require a rapid response. A timely response is crucial in such an application to avert catastrophes, and false alarms may lead to unnecessary costly measures. Thus the problem of using network traffic to detect disruptions in a city can be suitably

formulated as a quickest detection problem. In addition, the nature of this kind of data as one that reflects periodic human activity indicates that HMMs would provide a more accurate description of the data than i.i.d. alternatives.

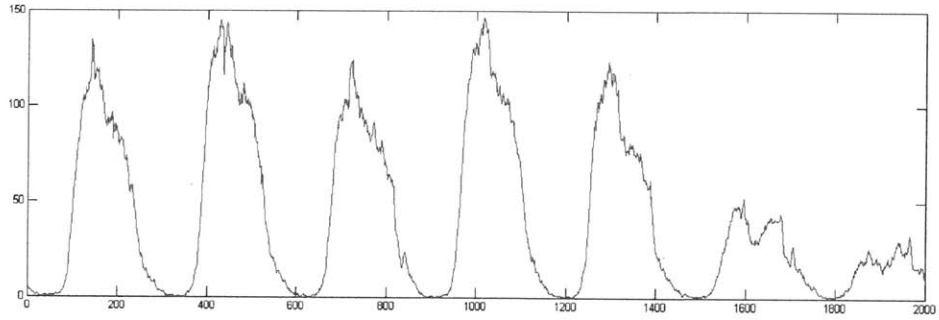

Figure 4-1: An Average Week of Network Traffic

In 2006, a collaboration between Telecom Italia (TI) and MIT's SENSEable City Laboratory allowed unprecedented access to aggregate mobile phone data from Rome. The data consists of normalized numbers of initiated calls in a series of 15 minute intervals spanning 3 months around Termini, Rome's business subway and railway station. The data exhibits differences between weekdays, fridays and weekends, with the latter exhibiting lower overall values. All days, however, share a pattern of rapid increase in communication activity between 6 and 10a.m. followed by a slight decrease and another increase after working hours. Night time exhibits the lowest level of traffic[7]. This activity is depicted in figure 4-1. .

A disruption of the periodic pattern is known to have occured in week 8 of the interval covered, leading to a surge in communication traffic, as shown in figure 4-2. In the following section we propose a method to detect such disruptions in correlated data using the HMM quickest detection algorithm in [6]. We test the performance of the algorithm on the Rome data assuming perfect knowledge of the business as usual and disruption models. This unrealistic assumption will be subsequently adressed in chapter 5.

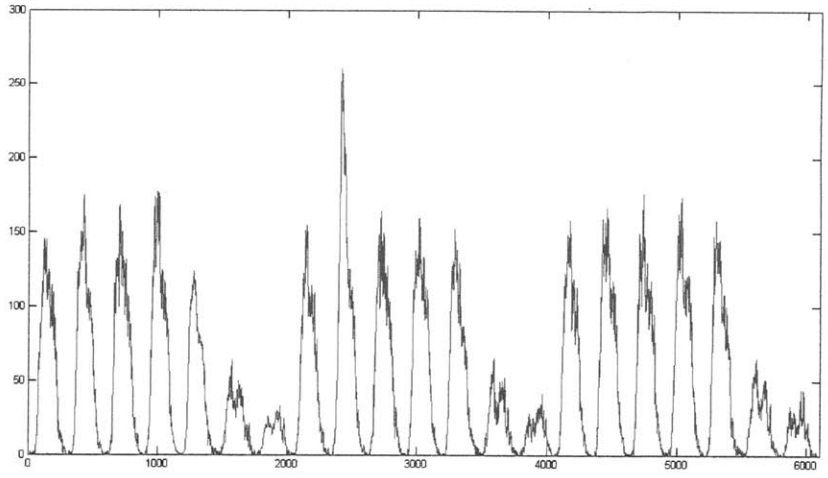

Figure 4-2: Three Weeks of Data from Termini with a Surge in Cellphone Network Traffic around Termini on Tuesday of the Second Week

4.2 Proposed Setup

Consider the two-state Markov chain $S = \{S_t; t \geq 0\}$, and let S_t denote the state of Rome at time t where $S_t \in \{0, 1\}$ for all t. $S_t = 0$ indicates that at time t, the data is in "business as usual", and $S_t = 1$ indicates a disruption. Assume that the initial probability of being in state $S_t = 1$ is θ_0, and that the transition matrix is

$$U = \begin{bmatrix} 1 - \theta_a & \theta_a \\ \theta_b & 1 - \theta_b \end{bmatrix}$$

Assume without a loss of generality that the chain starts with state $S_1 = 0$. The time T until the chain enters state 1 for the first time is a zero-modified geometric random variable with parameters θ_0 and θ_a. Detecting a disruption as fast as possible is equivalent to applying a quickest detection algorithm on real observations where the change time is T. The model parameters W_0 , W_1 and the emission probabilities for the pre/post-change models are obtained from training data. Emission probability distributions describing the normalized initiated call counts for different states of the city will be assumed Poisson. Upon detecting a disruption, the algorithm is restarted while switching the pre-change and post-change HMMs to detect the end of the

disruption and return of business as usual. The new change time is a zero-modified geometric random variable with parameters θ_1 and θ_b, where θ_1 is the probability that the first observation after declaring a disruption is a business as usual observaton.

4.3 Modeling the Data Set using HMMs

The quickest detection algorithm in [6] assumes that the HMM model describing the observation sequence is known before and after the change. Eventhough this assumption may be unrealistic for the disruption state, the description for business as usual state can be obtained fairly accurately through fitting an HMM in long training sequences from the past. In this section, we focus on the implementation aspects of fitting an HMM in the business as usual Rome data obtained from Termini.

For a given number of underlying states N_s, the parameters that describe a hidden markov model with Poisson distributions are [17]:

- The transition probability matrix A of the underlying Markov chain, where $a_{ij} = Pr\left[s_{t+1} = j \mid s_t = i\right]$ and s_t is the underlying state at time t.
- The emission probability vector B such that $B(i)$ is the parameter of the Poisson random variable describing the observations under state i of the underlying Markov chain.
- The vector I of initial probabilities of the underlying states.

The joint probability and defining property of an HMM sequence is:

$$Pr(s_1, s_2, \ldots, s_t, x_1, x_2, \ldots, x_t) = I \left[\prod_{k=1}^{t-1} a_{s_k s_{k+1}}\right] \left[\prod_{l=1}^{t} Poiss(x_l, s_l)\right]$$

where $Poiss(x_l, s_l)$ refers to the Poisson pdf at x_l with parameter $B(s_l)$.

We would like to solve for the model $\Omega = (A, B, I, N_s)$ that best describes a finite training sequence of business as usual data O for a fixed number of states N_s. There is no known way to analytically solve for the model Ω with maximizes $Pr(O \mid \Omega)$ [17]. However, starting with an intial "guess" model $\Omega_1 = (A_1, B_1, I_1, N_s)$, we can choose

a model $\Omega_2 = (A_2, B_2, I_2, N_s)$ that locally maximizes $\Pr(O \mid \Omega)$ using an iterative procedure like Baum-Welch [9] (a variant of the EM Algorithm [11]). In this work, we used functions from the "mhsmm" package in the statistical computing language *R* [16] to implement the Baum-Welch algorithm for Poisson observations. Note that this approach assumes the number of states to be fixed by the user and provides locally optimal solutions.

The choice of initial model parameters in the Baum-Welch algorithm has a significant effect on the optimality of its final estimate. In particular, the choice of emission probability parameters are essential for rapid and proper convergence of the re-estimation formulas used [17]. In our implementation, we made the following choices for the initial model:

- All underlying states are initially equally likely.
- All transitions are allowed and equally likely.
- The parameters of the different Poisson emission probabilities were obtained by sorting the observation sequence, dividing it into N_s bins of equal length, and averaging the observations in each bin resulting in B.

To determine the number of states N_s that best describes our data sequence O, we ran the Baum-Welch algorithm on O for increasing number of states N_s. $Pr(O \mid \Omega(N_s))$ where $\Omega(N_s)$ is the locally optimal model for N_s states, was taken as a measure of accuracy of fit.

The probability of an observation sequence O of length *len* under a given HMM Ω=(A,B,I) can be calculated using the Forward Algorithm [2][1]. Consider the forward variable defined by:

$$\alpha_t(i) = Pr[O_1 O_2 ... O_t, s_t = i \mid \Omega]$$

We can find $\alpha_t(i)$ recursively, as follows:

- Initialization: $\alpha_1(i) = I(i).Poiss(O_1, B(i))$ for $1 \leq i \leq N_s$.
- Induction: $\alpha_{t+1}(j) = \left(\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right) Poiss(O_{t+1}, B(j))$ for $1 \leq j \leq N_s$ and $1 \leq t \leq len$.

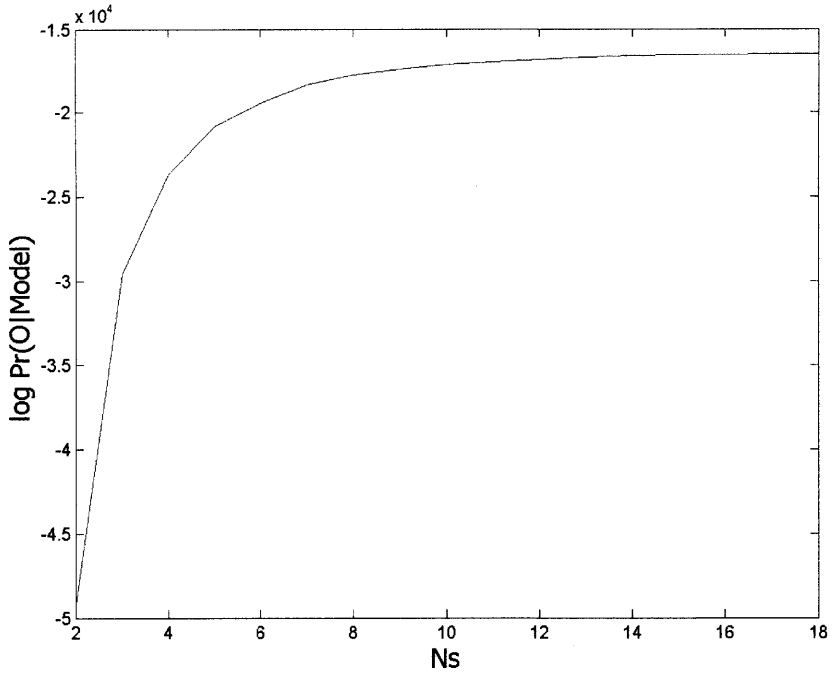

Figure 4-3: $\log\{Pr[O \mid \Omega(N_s)]\}$ for Increasing Number of States of Underlying Model

- Termination: $Pr(O|\Omega) = \sum_{i=1}^{N} \alpha_{len}(i)$

In practice, the forward variable "underflows" for larger t, meaning that it heads exponentially to 0 exceeding the precision range of essentially any machine [17]. A standard procedure to tackle this issue is to scale $\alpha_t(i)$ by a factor independent of i:

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)}$$

Then we can calculate the desired probability as:

$$log[Pr(O \mid \Omega)] = -\sum_{t=1}^{len} log(c_t)$$

Figure 4-3 is a plot of $\log\{Pr(O \mid \Omega(N_s))\}$ for N_s between 2 and 18. Note that increasing the number of states initially results in increasing accuracy, but beyond $N_s = 6$ there is no significant gain in increasing N_s.

In many real life scenarios, the training data available for HMM fitting has missing

samples creating the need for interpolation. The data set available to us had missing samples spanning days at a time, making interpolation an unfavorable option, especially knowing the pattern of variation within each day. To preserve the periodic pattern of the data described in [7], we chose to address the issue by replacing missing samples with the average value corresponding to available samples in the same day of the week (on a different week) and time of the day as the missing sample.

4.4 Implementation of the Quickest Detection Algorithm

For a given HMM description of the data, the quickest detection algorithm cosists of two parts: one that needs to be executed only once and offline to calculate the region $\Gamma^{N(\varepsilon)}$ for the given model parameters, and one that uses $\Gamma^{N(\varepsilon)}$ to detect changes in different data sequences from the model in real time.

Despite the favorable properties of region $\Gamma^{N(\varepsilon)}$ (namely convexity and closedness), its boundary cannot generally be expressed in closed form [6]. The offline determination of $\Gamma^{N(\varepsilon)}$ involves computing $J^{N(\varepsilon)}(\pi)$ for all $\pi \in \mathcal{P}$. Since $\mathcal{P}$ is the 2d-dimensional probability simplex, this compuation is intractable. To get around this problem, we resort to a simple form of cost-to-go function approximation, where the optimal cost-to-go is computed only for a set of representative states.

The representative states are chosen through a discretization of the 2d-dimensional probability simplex. The optimal cost to go is then calculated offline for all posterior probability vectors in the resulting grid. However, for a posterior probability vector π in the grid, the calculation requires knowing the value of the optimal cost-to-go function at all states (posterior probability vectors) accessible from π. Nearest neighbour interpolation is used to estimate the cost-to-go values corresponding to accessible states outside the grid.

Determining $\Gamma^{N(\varepsilon)}$ poses another challenge relating to the calculation of the Q-factor at stage k+1: $E_{X_{t+1}}\left(J^k(\Pi_{t+1}) \mid \Pi_t = \pi_t\right)$. Even when the values of $J^k(\Pi_{t+1})$

for different Π_{t+1}'s are known, the expectation is taken over X_{t+1} that can take countably infinite values. Conditioned on the underlying state Y_{t+1}, X_{t+1} is a Poisson random variable with a parameter $\lambda(y)$ that depends on the value y of Y_{t+1}. For that reason, we approximate the Q-factor for a certain $\Pi_t = \pi_t$ by the weighted average of $J^k(\Pi_{t+1})$ evaluated at a well-chosen set S_A of values of X_{t+1}. More specifically, the approximate Q-factor is:

$$Q^k_{app}(\pi) = \frac{\sum_{x \in S_A} Pr(X_{t+1} = x \mid \pi_t) J^k(\pi_{t+1})}{Pr(X_{t+1} \in A \mid \pi_t)}$$

The set A is chosen to include points where most of the Poisson p.d.f. is centered for parameters $\lambda(y)$ where $y \in \mathcal{Y}$.

4.5 Simulation Results

In this section, we show through simulation that the algorithm outlined in [6] can indeed be utilized to detect the disruption shown in figure 4-2 despite several approximations taken to facilitate its running. However, the computational complexity of the algorithm renders it extremely impractical, especially for rich data sets whose description requires a high number of markov states.

In section 4.3, we showed that the number of underlying states required to model the data set with "reasonable" accuracy is $N_s = 6$. However, the discretization step described in 4.4 results in exponential computational complexity in the number of states N_s. For simplicity of computations, we start by modeling the data with a three state HMM, hoping that the model captures enough of the properties of the business as usual and disruption states to detect the disruption. The number of underlying states is then increased if needed in order to achieve better detection.

A locally optimal three state HMM for the Termini data is shown in table 4.1. The initial model for the HMM fitting was obtained as per the guidelines in section 4.3 .

We run the change detection algorithm in the beginning of the disruption week.

Model Parameter	Business As Usual Model			Disruption Model		
Inital Probability	[1 0 0]			[1 0 0]		
Transition Matrix	0.9851	0.0149	0	0.9522	0.0478	0.0000
	0.0167	0.9737	0.0096	0.0905	0.8000	0.1095
	0	0.0089	0.9911	0.0000	0.1191	0.8809
Emission λ's	[3 27 107]			[197 232 249]		

Table 4.1: Model Parameters of Business As Usual and Disruption States for $N_s = 3$

The disruption week consists of 2016 samples, where the disruption starts on Tuesday around sample 406 (with a value of 181).

False Alarm Cost	Horizon $N(\varepsilon)$	Detection Sample
0.5	31	404
1	301	405
10	3034	405
20	6143	405

Table 4.2: Detection Time and Horizon for Different False Alarm Costs when $N_s = 3$

With a discretization step size of 0.25, sample cost of 1, error margin ε=3 (equivalent to detecting on average three samples after a disruption has started, if no false alarms are allowed), and a prior $\theta = 1/300$ (equivalently expressed as expected time till disruption being 300 samples starting from business as usual), we obtain the detection times in table(4.2) for different false alarm costs.

Model Parameter	Business As Usual Model				Disruption Model			
Inital Probability	[1 0 0 0]				[1 0 0 0]			
Transition Matrix	0.9869	0.0131	0	0	0.0727	0.9273	0.0000	0.0000
	0.0153	0.9683	0.0164	0	0.0000	0.9502	0.0498	0.0000
	0	0.0279	0.9510	0.0210	0.0000	0.0907	0.7992	0.1101
	0	0	0.0127	0.9873	0	0.0000	0.1217	0.8783
Emission λ's	[2 19 58 117]				[181 198 232 250]			

Table 4.3: Model Parameters for Business As Usual and Disruption for $N_s = 4$

The number of states of the underlying model was then increased to $N_s = 4$, keeping all other parameters constant, resulting in the model in table 4.3, and the

False Alarm Cost	Horizon $N(\varepsilon)$	Detection Sample
0.5	134	404
1	267	406
10	2700	406
20	5467	406

Table 4.4: Detection Time and Horizon for Different False Alarm Costs when $N_s = 4$

detection times in table 4.4.

For $N_s = 3$, we observe that a false alarm occured for all values of C_F/C_D chosen, resulting in a cost of 20. The value observed at 405 is 171, which is generally assumed to be a business as usual value for weekday peak time traffic. However, loosely speaking, this value is considered to be at the boundary of allowable business as usual observations, since the highest λ in the business as usual state for $N_s = 3$ is 107. Meanwhile the lowest λ in the disruption state is 197. It is then understandable that for $N_s = 3$ and for a cost of false alarm in the range we chose, this boundary case is resolved by detecting a disruption. Note that running the algorithm for C_F/C_D high enough to result in correct detection was practically infeasible on a regular machine. Coincidentally, for this particular case, the false-alarm-causing value (171) was close enough to the actual disruption that the behavior was acceptable for real applications.

Increasing N_s to 4 helped resolve the false alarm issue discussed above for a computationally practical cost C_F/C_D. We also notice that the detection accuracy was almost independent of the false alarm cost when the cost of false alarm is greater than the sample cost. This behavior is not typical of the quickest detection algorithm, which is generally expected to exhibit variation in performance with varying cost criteria. We hypothesize that this behavior was observed due to a much higher likelihood of the disruption observations under the disruption model than the business as usual model. "Much higher" is measured in terms of the difference between sample and observation cost.

The fact that we were able to detect the change with $N_s = 4$ indicates that the difference between business as usual and disruption data is significant enough that even a crude model was enough to describe it.

The computational complexity of the offline procedure with the approximations suggested is in the oder of

$$(N(\varepsilon) + N_P)\left(\frac{1}{dstep} + 1\right)^{2d-1}$$

where dstep is the probability grid step size, and N_p the number of poisson samples used to approximate the Q-Factor.

Chapter 5

NonBayesian CUSUM Method for HMM Quickest Detection

A different approach for HMM-quickest detection was proposed by Chen and Willet [5]. Their approach is non-Bayesian and mimicks the CUSUM test for i.i.d. observations. In this chapter, we present the algorithm suggested in [5], and compare it with the previously discusses approach in [6], especially from the point of view of feasibility for real data sets. We then shed light on the robustness of the algorithm when the disruption model is not entirely known.

5.1 Sequential Probability Ratio Test (SPRT)

The sequential probability ratio test (SPRT) was devised by Wald [19] to solve the following sequential hypothesis testing problem. Suppose that $X_1, X_2, \ldots, X_n$ is a sequence of observations arriving continually (i.e. n is not a fixed number), and suppose that H (K) is the hypothesis that the observations are i.i.d. according to distribution P_H (P_K). The purpose is to be able to identify the correct hypothesis as soon as possible, while minimizing decision errors.

For a given n, the likelihood ratio is

$$L(n) = \frac{P_k(X_1, \ldots, X_n)}{P_H(X_1, \ldots, X_n)} = \frac{P_K(X_1)}{P_H(X_1)} \prod_{i=2}^{n} \frac{P_K(X_i \mid X_{i-1}, \ldots, X_1)}{P_H(X_i \mid X_{i-1}, \ldots, X_1)}$$

The SPRT defines the following stopping rule:

$$N^* = \min\{n \geq 1 : L(n) \geq A \text{ or } L(n) \leq B\}$$

The decision made is H if $\{L(n) \geq A\}$ and K otherwise. The thresholds A and B are chosen to meet a set of performance criteria. The performance measures include, as in the traditional hypothesis testing problem, the probability of error (P_e) and rate of false alarm (P_f):

$$\begin{aligned} P_e &= P_K(L(N^*) \leq B) \\ P_f &= P_H(L(N^*) \geq A) \end{aligned}$$

The design parameters A and B can be determined in terms of the target Pe and P_f using Wald's approximation, giving:

$$A \approx \frac{1 - P_e}{P_f} \tag{5.1}$$

$$B \approx \frac{P_f}{1 - P_e} \tag{5.2}$$

In addition to minimizing P_e and P_f, the sequential nature of the problem poses two more performance criteria: the average run length (ARL) under H and K. The SPRT is optimal in the sense that it minimizes ARL under both H and K for fixed P_e and P_f.

5.2 Cumulative Sum (CUSUM) Procedure

In section 2.1.1, we mentioned that the optimal stopping time for the minimax formulation of the i.i.d. non-Bayesian quickest detection problem is given by Page's

decision rule (2.2).

For pre-change and post-change probability measures f_H and f_K, Page's test can be easily reformulated in terms of the following recursion, known as the CUSUM test [13]:

$$N^* = \min\{n \geq 1 : S_n \geq h\}$$

where

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

and

$$g(X_n) = \ln\left(\frac{f_K(X_n)}{f_H(X_n)}\right)$$

h is a parameter than can be chosen according to our desired tradeoff between false alarm and detection delays.

Essentially, the CUSUM test is a series of SPRTs on the two hypotheses H (observations are due to f_H) and K (observations are due to f_H) where the lower limit A is 0 and the upper limit B is h. Every time the SPRT declares hypothesis A as its decision, we can assume that the change has not occured yet, and the SPRT is restarted. This pattern continues until the first time that the SPRT detects hypothesis K (hence crossing the threshold h for the first time).

The main requirement for CUSUM-like procedures to work is the "antipodality" condition:

$$\begin{aligned} E[g(X_n) \mid H] &< 0 \\ E[g(X_n) \mid K] &> 0 \end{aligned}$$

For $g(X_n) = \ln(\frac{f_K(X_n)}{f_H(X_n)})$, those conditions follows from the non-negativity of conditional KL distance.

The only performance measures for CUSUM are the ARL under H and K, since detection is guaranteed to happen for all "closed" tests ($Pr[N^* < \infty] = 1$). CUSUM exhibits minimax optimality in the sense that, for a given constraint on the delay between false alarms, it minimizes the worst case delay to detection.

5.3 HMM CUSUM-Like Procedure for Quickest Detection

Designing CUSUM procedures involves finding a relevant function $g(X_n)$ that satisfies Page's recursion and the antipodality principle. [5] proposes using the scaled forward variable described in section 4.3 for defining such a function for quickest detection on HMM observations. For the k'th sample after the last SPRT,

$$g(n;k)(X_n) = \ln\left(\frac{f_K(X_n \mid X_{n-1},\ldots,X_k)}{f_H(X_n \mid X_{n-1},\ldots,X_k)}\right)$$

The CUSUM recursion is then

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

To calculate $f_K(X_n \mid X_{n-1},\ldots,X_k)$ and $f_H(X_n \mid X_{n-1},\ldots,X_k)$, the scaled forward variable $\hat{\alpha}_t(i)$ is used:

$$\begin{aligned} f_H(X_t \mid X_{t-1},\ldots,X_1) &= \sum_{i=1}^{N} \hat{\alpha^H}_t(i) \\ f_K(X_t \mid X_{t-1},\ldots,X_1) &= \sum_{i=1}^{N} \hat{\alpha^K}_t(i) \end{aligned}$$

The antipodality property for this procedure is proven in [5] and is again directly related to the non-negativity of conditional KL-distance. It is also shown that the average detection delay is linear in h, whereas the average delay between false alarms is exponential in h, just like the behavior exhibited for the i.i.d. CUSUM. This "log-linear" behavior is the main reason why CUSUM-like algorithms work.

More especifically, the average detection delay (D) and time till false alarm (T)

are given by:

$$\begin{aligned} D &= \frac{h}{D(f_K \,||\, f_H)} \\ T &\approx \frac{e^h}{1-\bar{B}} \end{aligned}$$

where $\bar{B}$ is the expected value of a single SPRT test statistic B knowing that a correct detection for H has occured. $\bar{B}$ can be obtained through simulation.

5.4 Simulation Results for Real Data Set

In this section, we discuss the results of running the HMM-CUSUM algorithm on the data from Termini. The algorithm does not require any priors to be given, except for the HMM before and after disruption and a threshold h. The models parameter used correspond to $N_s = 6$ and are given in the Appendix. The threshold h determines the desired tradeoff between the frequency of false alarms and average detection delay. Higher h results in exponentially lower frequency of alarms, and a linear increase in detection delay.

Figures 5-1, 5-2 and 5-3 show the alarm times obtained from running HMM-CUSUM on Termini data for different values of h. In figure 5-3, we observe that small h ($h < 1.5$) leads to false alarms. The time between false alarms, however, increases exponentially as h increases (figure 5-1), and for $1.5 \leq h \leq 1100$, HMM-CUSUM consistently detected the disruption but with linearly increasing delay (figure 5-2). Finally, beyond $h = 1100$, the alarm time went to infinity (figure 5-1), meaning that the disruption went undetected.

This behavior can be explained by plotting the trajectory of S_n, the cumulative sum, for the Termini data under different values of h from the intervals outlined above. Figure 5-4 shows the behavior of the cumulative sum in the "detection region". Initially, the data showed an overwhelming likelihood of being "business as usual" which lead to restarting the SPRT with the negative S_n reset to 0. As n approached Monday's peak traffic time, the likelihood of disruption increased (since high traffic

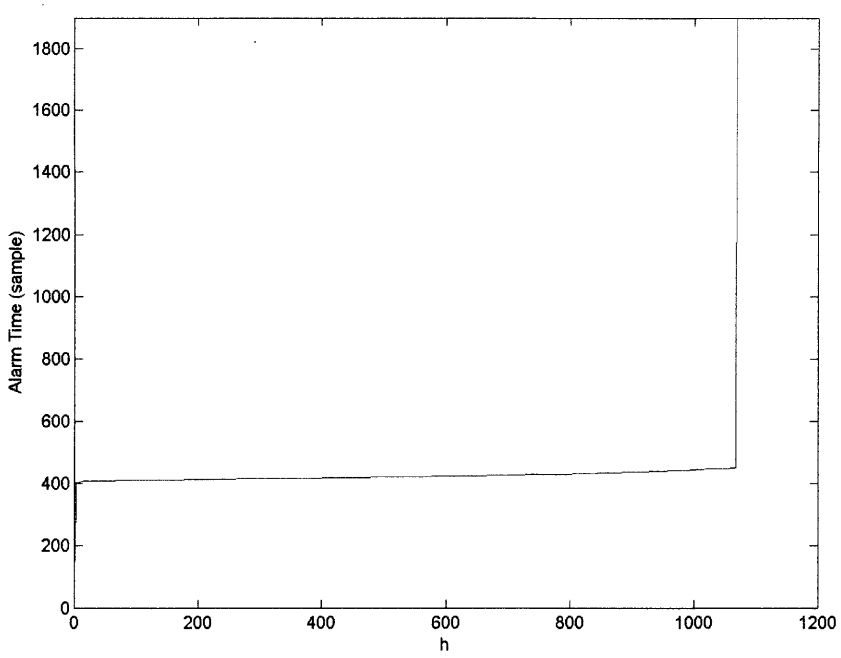

Figure 5-1: Detection Time Versus h for 1 week Termini Data Fit with a 6-state HMM: $h \in (1, 1200)$

is characteristic of the disruption) leading to a positive drift in the direction of h, but the increase was not enough to cross the threshold and raise an alarm. Eventually, Tuesday's surge in traffic raised the likelihood of disruption to the point of crossing the threshold, leading to detection. Figure 5-5 shows the behavior of S_n when h is in the false alarm region. In this case, the positive drift caused by Monday's peak traffic was enough to cross the low threshold, and cause a false alarm. Finally, when h is too high, the drift caused by 45 disruption samples is not enough for S_n to cross the threshold, and the consequent business as usual samples only serve to point the drift away from the threshold towards 0, as shown in figure 5-6.

Figure 5-7 shows an approximate average alarm time for data sampled from the Termini disruption model (Appendix) under different values of h. The average alarm time for each h was obtained through Monte Carlo simulations. To guarantee detection, on average, the value of h required is greater than 7.5, much higher than the corresponding value (≥ 1.5) required for the specific Termini sequence. One explanation for that is that the Termini sequence was used to define business as usual and

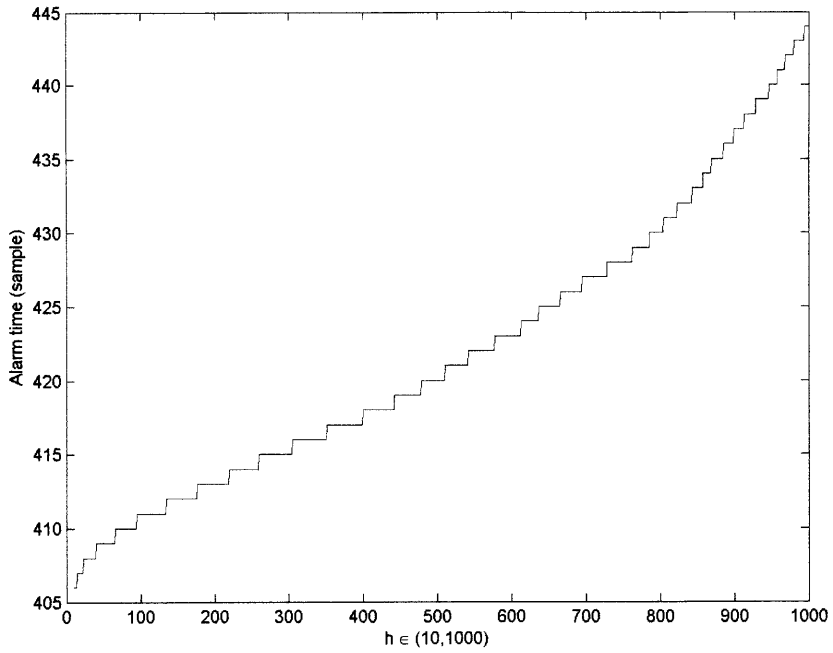

Figure 5-2: Detection Time Versus h for 1 week Termini Data Fit with a 6-state HMM: $h \in (10, 1000)$

disruption states, and is thus a "perfect fit" to the model and decision rule, clearly depicting the variation between business as usual and disruption. This behavior is not necessarily typical of all sequences from the model.

In the context of disruption detection for real data, the HMM-CUSUM algorithm exhibits several advantages as compared to the DP-based bayesian framework suggested in [6]:

- It gives instant "real time" results without the need to run lengthy computations offline.

- Its implementation was fairly simple and did not involve some of the issues encountered with the DP-based algorithm (infinite state space, infinite horizon and a cost-to-go function and Q-factor without a closed form).

- Increasing accuracy in HMM-CUSUM did not lead to any aditional computational complexity and was simply achieved by increasing the threshold h, unlike

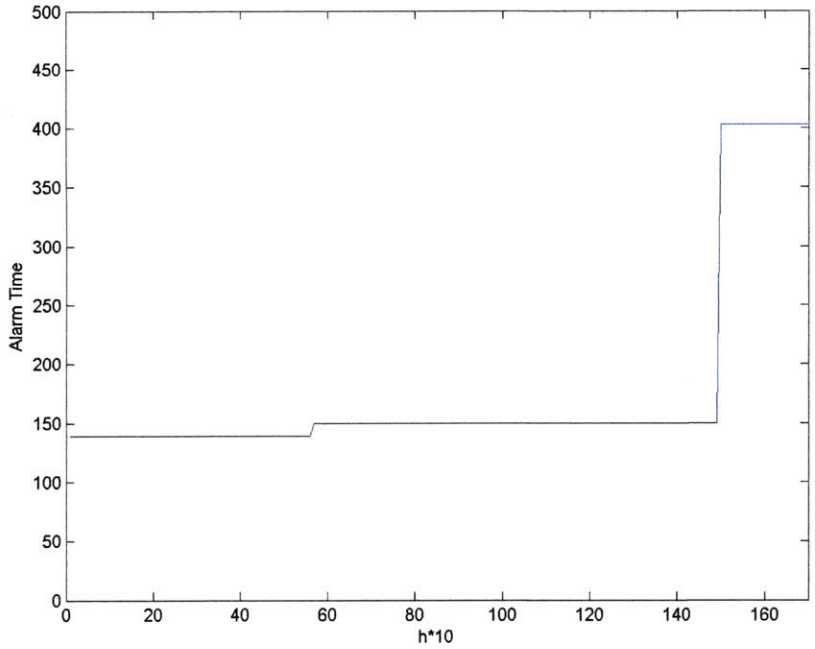

Figure 5-3: Detection Time Versus h for 1 week Termini Data Fit with a 6-state HMM: $h \in (0, 1.6)$

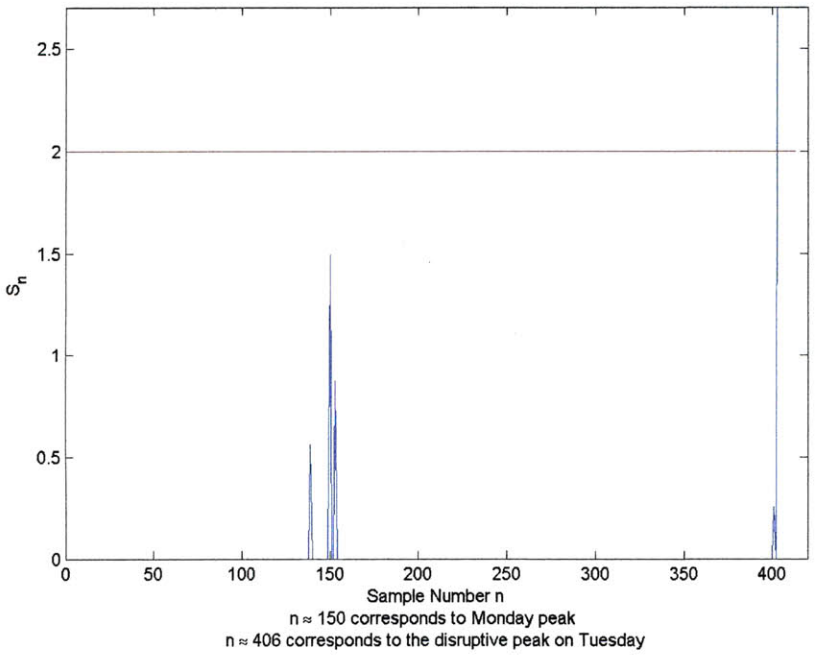

Figure 5-4: HMM-CUSUM as a Repeated SPRT: No False Alarm Case

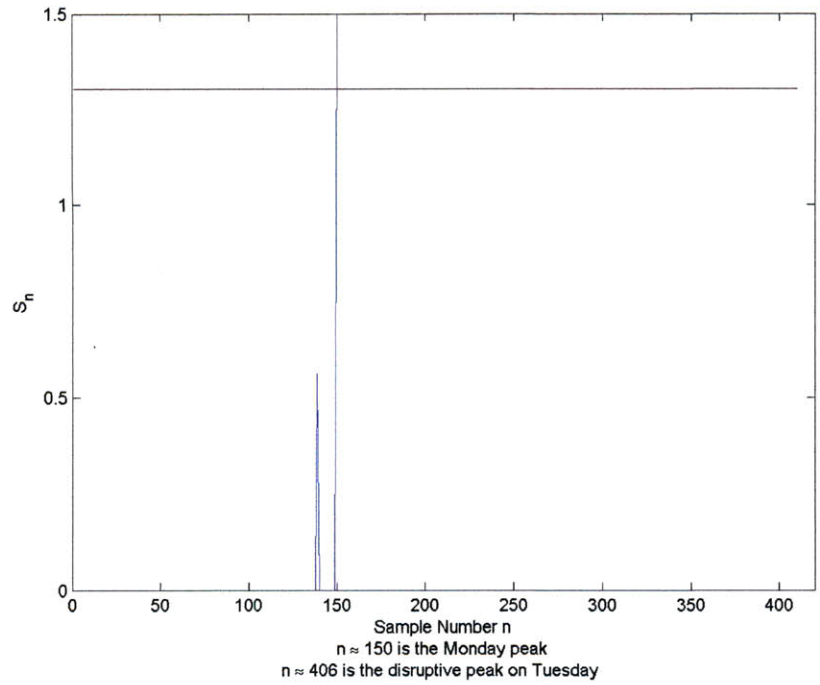

Figure 5-5: HMM-CUSUM as a Repeated SPRT: Case of False Alarm around h=150

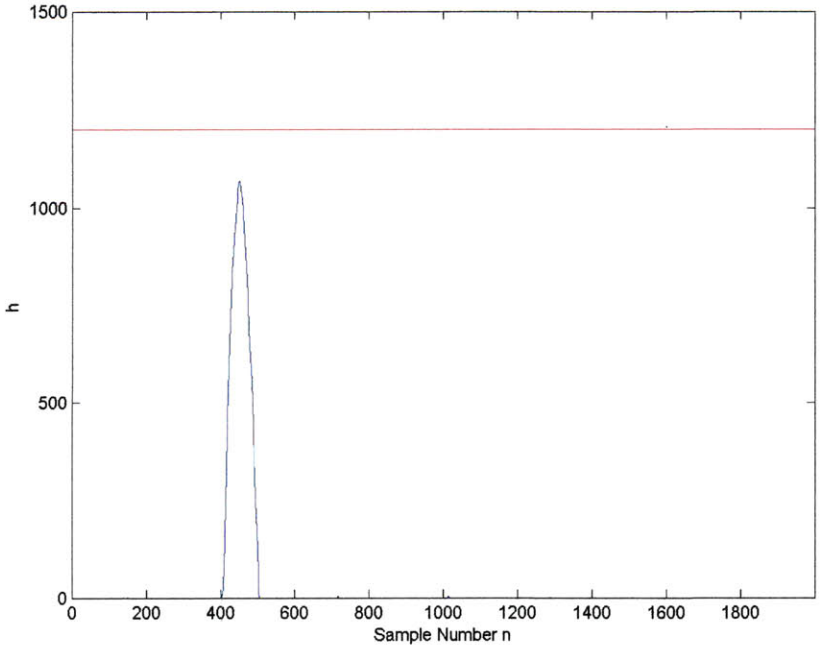

Figure 5-6: HMM-CUSUM as a Repeated SPRT: Case of No Detection Caused by Too High Threshold

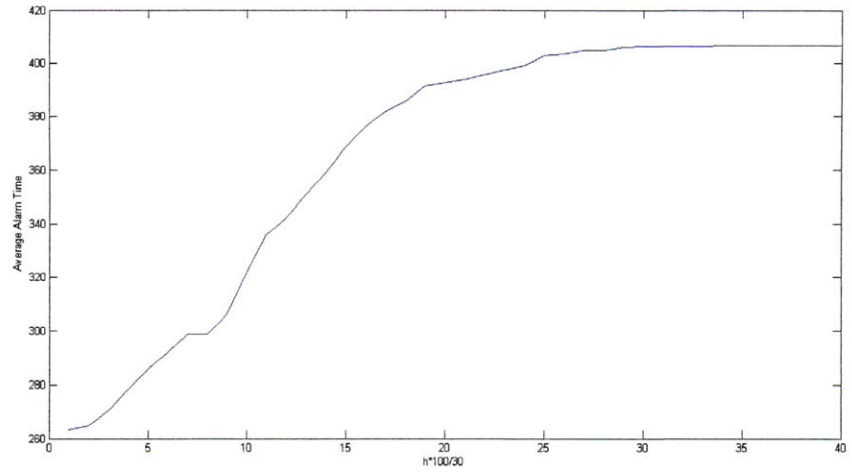

Figure 5-7: Average Alarm Time Versus h for Locally Optimal 6-State Model Describing Rome Data

in the bayesian framework where increasing C_F/C_D resulted in a longer horizon and more computations.

- The CUSUM procedure scales well with the increasing number of underlying model states. The DP-based algorithm has exponential complexity in the number of states under the proposed discretization scheme.

- The HMM-CUSUM does not require a look-up table (memory) for its implementation, as does the DP-based algorithm (for storing the decision region).

The Bayesian approach in turn has the following advantages when compared to CUSUM:

- The Bayesian approach has rigorous theoretical guarantees on its performance, whereas HMM-CUSUM's performance guarantees often include approximations.

- The user is given more parameters to control the tradeoff between false alarm and detection delay. In addition to C_F and C_D, the user can control the prior belief on the time until disruption and the probability of starting in the disruption state.

- The costs C_F and C_D are intuitively meaningful, unlike the threshold h which can represent vastly different tradeoff for different problems.

5.5 Robustness Results

In this section we examine, through simulation, the robustness of the HMM-CUSUM procedure when the disruption model is not entirely known.

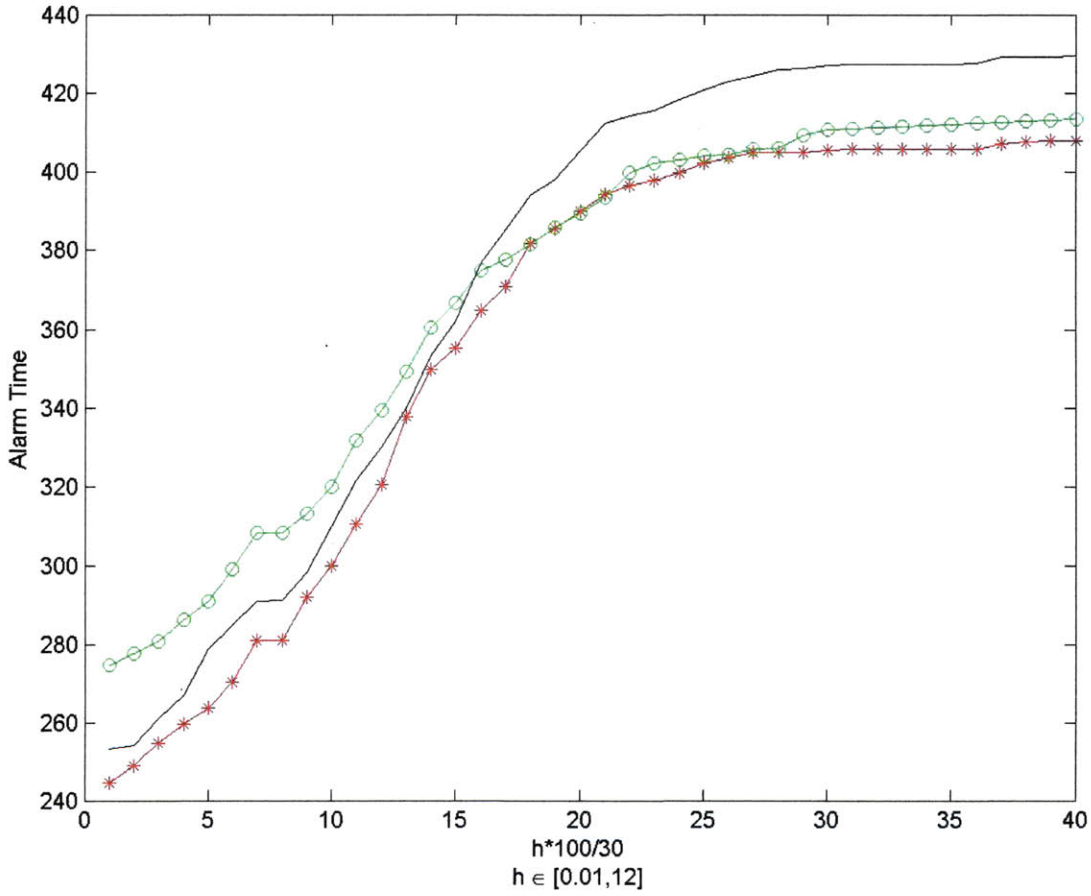

Figure 5-8: Average Alarm Time Versus h for Different Models in the Disruption Class. (* = model 1) (o = model 2) (- = model 3)

First, we study the effect of incorrectly assuming that the disruption model is the one obtained from Termini data (Appendix) when the actual disruption model is as described in table 5.1. The actual models have the same underlying transition matrix and initial state distribution as the assumed model, with progressively smaller

Model	Emission λ's
1	161 166 177 194 213 228
2	141 146 157 174 193 208
3	121 126 137 154 173 188

Table 5.1: Actual Disruption Emission λ's (Transition Matrix and Initial Probability Same as Those from Termini)

emission λs. Figure 5-8 shows the average alarm time for data drawn from those different disruption models under different values of h.

If the actual disruption model is in fact the one used to design the HMM-CUSUM procedure (Appendix), an optimal threshold of interest is $h = 7.5$. On average, the interval $h \geq 7.5$ guarantees negligible false alarm rates, and since detection delay increases with the threshold, the lowest h in the interval would in addition guarantee minimal detection delay. Hereafter, we focus our analysis on the threshold $h = 7.5$.

Figure 5-8 shows that the detection delay for $h = 7.5$ under the different disruption models is on average higher (less optimal) than the one obtained when the actual and assumed models were identical (figure 5-7). This phenomenon has a simple explanation. When the actual λs of the disruption model are less than the assumed ones, it will take more samples for S_n to show the magnitude of positive drift that would lead to crossing the threshold designed for higher λs. This leads to delayed detections on average.

Second, we attempt to find a robust HMM-CUSUM procedure that would guarantee a desired performance when the disruption is known to belong to a class $\mathcal{C}$ with $|C|$ models. The desired performance we focus on is again minimizing detection delay for negligible false alarm rate. The components defining the robust HMM procedure are the threshold h and the assumed disruption model.

We suggest the following procedure:

- For each model C_i in the class (C), plot the average alarm time when the HMM-CUSUM is designed for disruptions from C_i and the actual disruptions are drawn from C_j with $j = 1, \ldots, |C|$

Assumed Model	Threshold For False Alarm	Worst Average Alarm Time At Threshold	Worst Delay Model
Termini	9.6	437	Perturbed Model 3
Perturbed Model 1	9.6	427	Perturbed Model 3
Perturbed Model 2	10	427	Perturbed Model 3
Perturbed Model 3	6.4	43	Perturbed Model 1

Table 5.2: Actual Disruption Emission λ's (Transition Matrix and Initial Probability Same as Those from Termini)

- From each C_i plot obtained, find the lowest threshold h_i that guarantees negligible false alarm rate (average alarm time$\geq$ 406) for all C_j. In addition, find the maximum average delay (D_i) for all C_j at the chosen h_i.

- Choose the pairing of C_i and h_i that minimizes the maximum average delay D_i found in the previous step.

This procedure outlines a practical method to find an approximate minimax robust HMM-CUSUM for a desired performance tradeoff, when the assumed model is constrained to belong to $\mathcal{C}$. For the class $\mathcal{C}$ consisting of the models in table 5.1 and the original model obtained from Termini, we obtain the plots in figures 5-9, 5-10, 5-11, and 5-12. In figure 5-9, we observe that when the HMM-CUSUM is designed for the Termini disruption, the desired false alarm performance is guaranteed for all models in $\mathcal{C}$ when $h = 9.6$. For that threshold, the worst average detection delay happens for data drawn from perturbed model 3 5.1 and has a value of 31 samples (worst case average alarm time = 437). Table 5.2 summarizes the results observed for all models in the class. Based on the simulation results, the design model of choice that guarantees minimax robustness for the desired performance tradeoff is perturbed Model 1 or 2 with a worst case detection delay of 21. The tie can be resolved by taking the one that has a lower threshold, perturbed Model 1. Formalizing this procedure is left for future work.

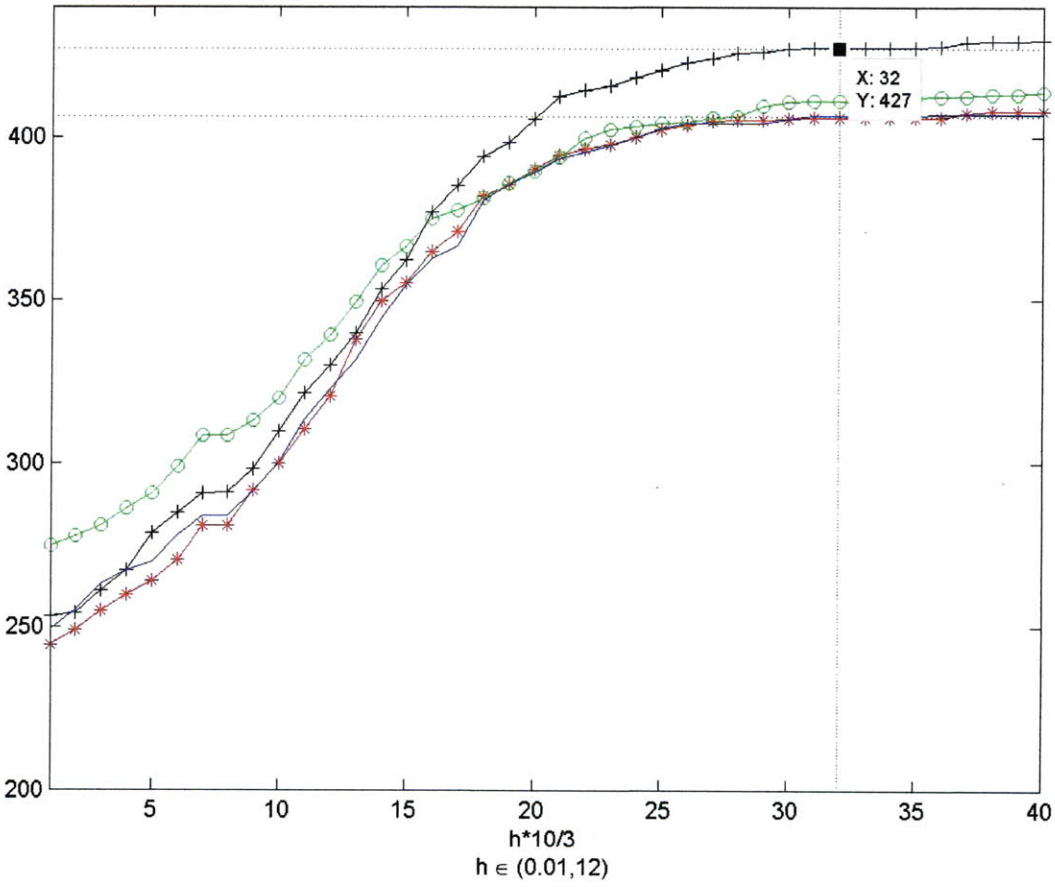

Figure 5-9: Performance of HMM-CUSUM when the Assumed Disruption Model is that Obtained from the Termini Data and the Actual Disruptions are Drawn from each Model in $\mathcal{C}$ (- = Model from Termini) (* = Perturbed Model 1) (o = Perturbed Model 2) (- = Perturbed Model 3)

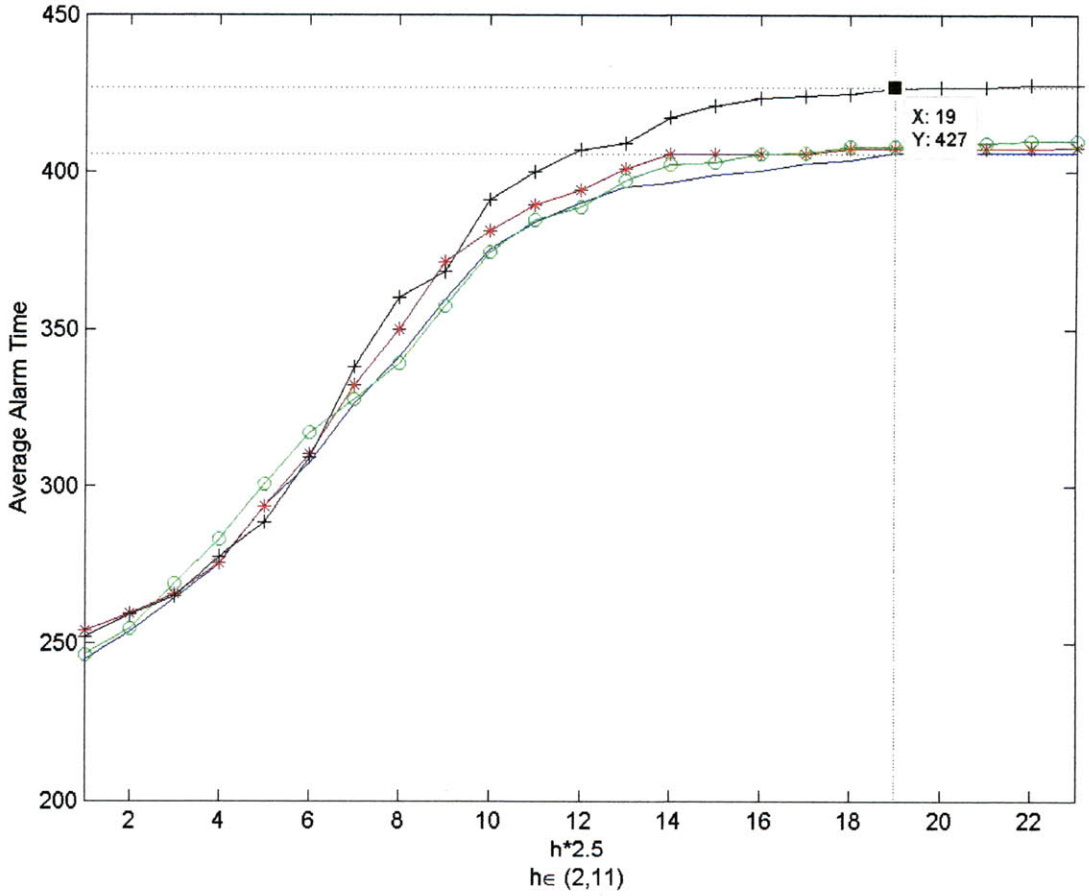

Figure 5-10: Performance of HMM-CUSUM when the Assumed Disruption Model is "Perturbed Model 1" and the Actual Disruptions are Drawn from each Model in $\mathcal{C}$ (- = Model from Termini) (* = Perturbed Model 1) (o = Perturbed Model 2) (- = Perturbed Model 3)

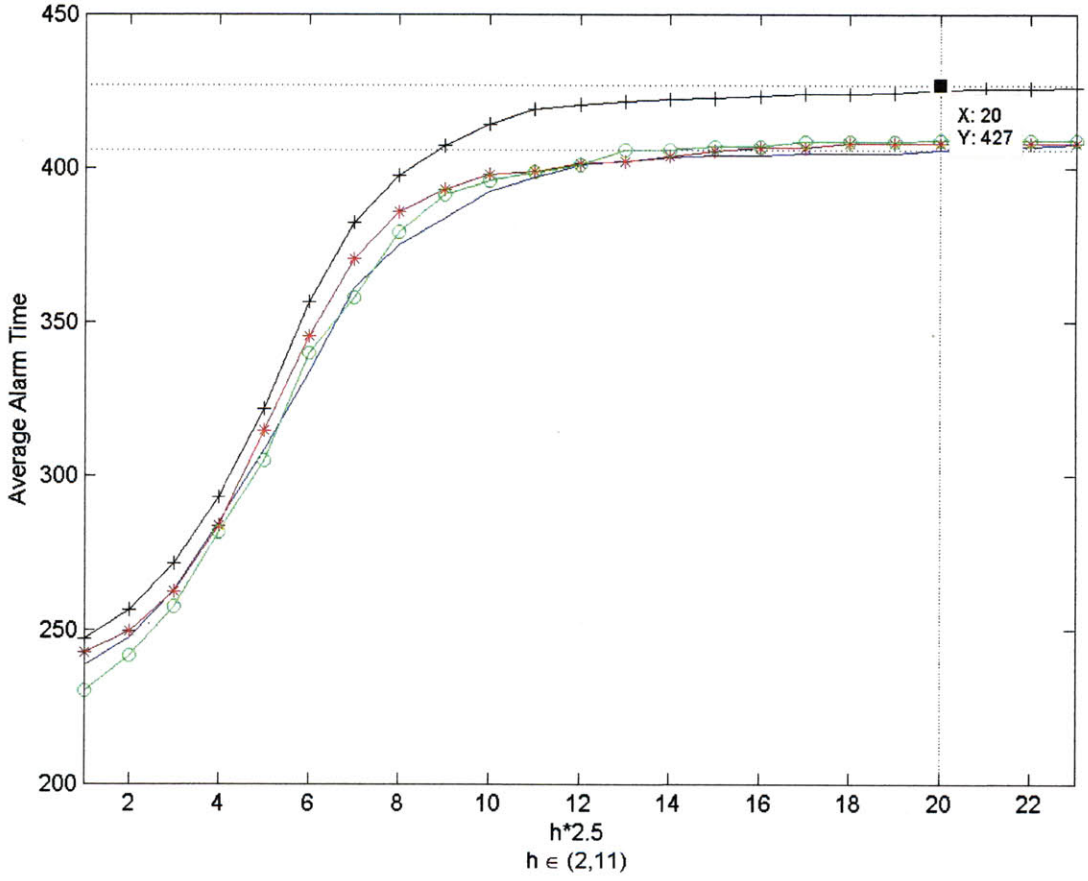

Figure 5-11: Performance of HMM-CUSUM when the Assumed Disruption Model is "Perturbed Model 2" and the Actual Disruptions are Drawn from each Model in $\mathcal{C}$ (- = Model from Termini) (* = Perturbed Model 1) (o = Perturbed Model 2) (- = Perturbed Model 3)

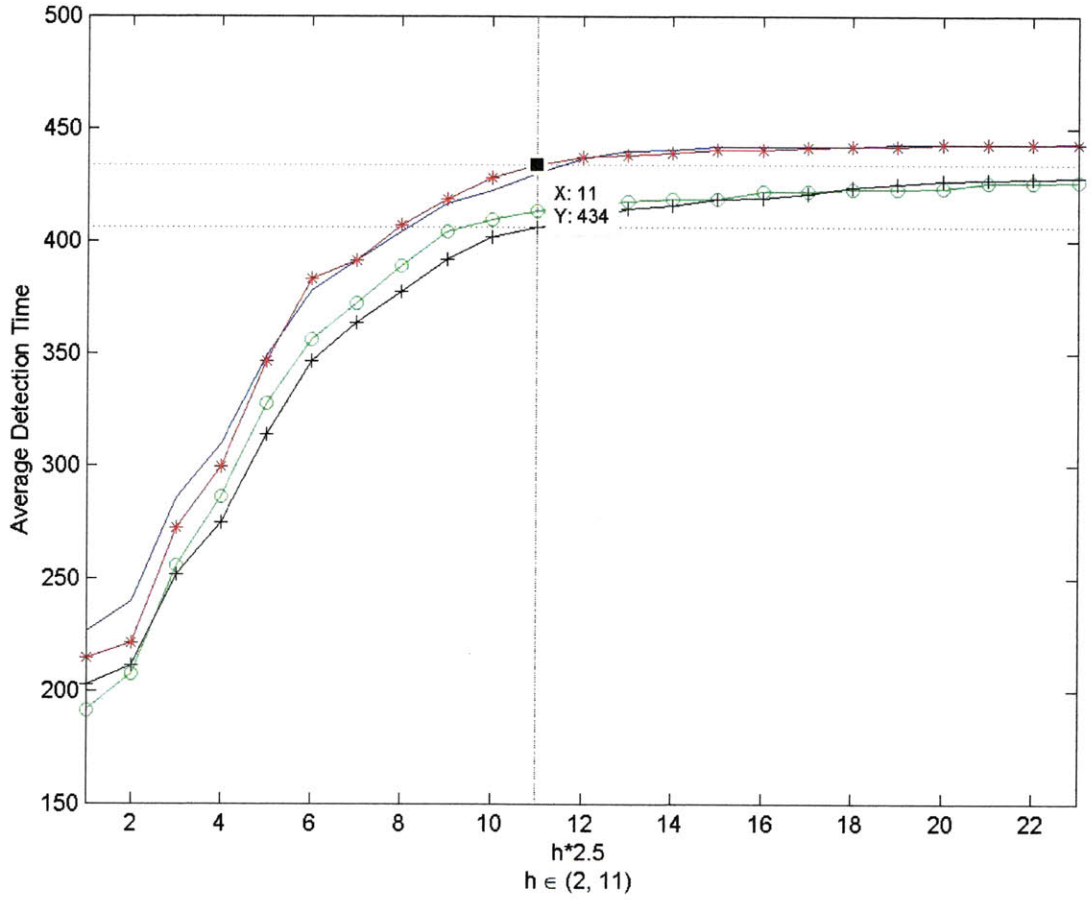

Figure 5-12: Performance of HMM-CUSUM when the Assumed Disruption Model is "Perturbed Model 3" and the Actual Disruptions are Drawn from each Model in $\mathcal{C}$ (- = Model from Termini) (* = Perturbed Model 1) (o = Perturbed Model 2) (- = Perturbed Model 3)

Chapter 6

Conclusion

In this thesis, we assessed the feasibility of two HMM quickest detection procedures for detecting disruptions in a real data sequence.

For the Bayesian procedure described in [6], we suggested approximations for finding the optimal region described by an infinite horizon dynamic program with a continuous state space. The approximations included discretizing the continuous state space and the countably infinite observation space, resulting in approximate cost-to-go function and Q-factor values for a grid of probability vectors. The suggested approximations do not scale well with increasing state-space dimension, which led us to experiment with running the algorithm on a three and four state HMM representation of the Rome data (which requires at least six states for its accurate representation). The algorithm successfully detected the disruption in the data for relatively low costs only when the number of HMM states was four. Eventhough the algorithm succeeded at detecting the disruption, our assessment was that the method is impractical for rich real data sets under the approximation methods we chose.

We proceeded to assess the feasibility of the non-Bayesian CUSUM-like procedure suggested in [5]. The procedure had the advantage of being very simple to implement, and of providing real-time detection without the use of time consuming offline calculations or memory intensive look-up tables. It scales well with the number of states of the underlying HMM models, which allowed us to to use a 6-state representation of the Rome data and find the alarm time for even the most extreme tradeoffs be-

tween false alarm frequency and detection delay. While the performance guarantees on HMM-CUSUM often involve approximations, the method is more suited for real data due to the advantages just outlined.

Finally, we examined the robustness of the HMM-CUSUM when the disruption model is not exactly known, but the belongs to a known class of HMMs. When the actual model for the disruption data is 'nearer' to the business as usual model than the assumed disruption model is, simulations show a noticeable performance drop for a fixed threshold. We then suggested an experimental approximate method for designing a CUSUM procedure that guarantees some sense of minimax robustness when the disruption belongs to a known class of models.

Future work on this topic can focus on:

- Finding optimal solutions for the robust HMM quickest detection in the minimax sense, extending the i.i.d. results in [8]

- Finding rigorous notions of distance for HMMs compatible with the quickest detection problem, especially when the initial state distributions of the pre-change and post-change sequences are not the steady state distributions of the corresponding HMMs.

- Finding more efficient approximations for the optimal region described in [6] exploiting its characteristics (convexity, closedness, etc) to increase the range of computationally feasible dimensionality.

Chapter 7

Appendix

Parameter	Business As Usual Model					
Inital Probability	[1 0 0 0 0 0]					
Transition Matrix	0.9857	0.0143	0.0000	0	0	0
	0.0259	0.9383	0.0357	0.0000	0	0
	0.0000	0.0331	0.9415	0.0254	0	0
	0	0.0000	0.0376	0.9381	0.0243	0.0000
	0	0	0	0.0158	0.9565	0.0276
	0	0	0	0.0000	0.0334	0.9666
Emission λ's	[1 10 24 50 94 133]					

Table 7.1: Model Parameters for Business As Usual for $N_s = 6$

Parameter	Disruption Model					
Inital Probability	[1 0 0 0 0 0]					
Transition Matrix	0.0000	1.0000	0.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.0000	1.0000	0.0000	0.0000
	0.0000	0.0000	1.0000	0.0000	0.0000	0.0000
	0.0000	0.0000	0.2423	0.5102	0.0000	0.2475
	0.0000	0.0000	0.0000	0.1201	0.8799	0.0000
	0.0000	0.0000	0.0000	0.0000	0.0900	0.9100
Emission λ's	[181 186 197 214 233 248]					

Table 7.2: Model Parameters for Business As Usual and Disruption for $N_s = 6$

Bibliography

[1] L. E. Baum and J. A. Egon. An inequality with applications to statistical estimation for probabilistic functions of a markov process and to a model for ecology. *Bull. Amer. Meteorol. Soc.*, 73:360–363, 1967.

[2] L. E. Baum and G. R. Sell. Growth functions for transformations on manifolds. *Pac. J. Math*, 27(2), 1968.

[3] D.P. Bertsekas. *Dynamic Programming and Optimal Control*, volume 1, chapter 8. Athena Scientific, third edition, 2007.

[4] D.P. Bertsekas and J.N. Tsitsiklis. *Neuro-Dynamic Programming*, chapter 1. Athena Scientific, 2007.

[5] B. Chen and P. Willett. Detection of hidden markov model transient signals. *IEEE Transactions on Aerospace and Electronic Systems*, 36(4):1253–1268, October 2000.

[6] S. Dayanik and C. Goulding. Sequential detection and identification of a change in the distribution of a markov-modulated random sequence. *IEEE Transactions on Information Theory*, 55(7), July 2009.

[7] A. Sevtsuk J. Reades, F. Calabrese and C. Ratti. Cellular census: Explorations in urban data collection. *IEEE Pervasive Computing*, 6(3), 2007.

[8] V. V. Venugopal J. Unnikrishnan and S. Meyn. Minimax robust quickest change detection. *IEEE Transactions on Information Theory*, 2, June 2010. Draft. Submitted Nov 2009, revised May 2010.

[9] G. Soules N. Weiss L. Baum, T. Petrie. A maximization technique occuring in the statistical analysis of probabilitic functions of markov chains. *Annals of Mathematical Statistics*, 41:164–171, 1970.

[10] G. Lorden. Procedures for reacting to a change in distribution. *Annals of Mathematical Statistics*, 42(6):1897–1908, 1971.

[11] T. Moon. The expectation maximization algorithm. *IEEE Signal Processing Magazine*, 13:47–60, 1996.

[12] G.V. Moustakides. Optimal stopping times for detecting changes in distributions. *Annals of Statistics*, 14(4):1379–1387, December 1986.

[13] E.S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[14] M. Pollak. Optimal detection of a change in distribution. *Annals of Statistics*, 13(1):206–227, 1985.

[15] V.H. Poor and O. Hadjiliadis. *Quickest Detection.* Cambridge Univeristy Press, first edition, 2009.

[16] R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2010. ISBN 3-900051-07-0.

[17] L. R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), February 1989.

[18] A.N. Shiryaev. On optimum methods in quickest detection problems. *Theory of Prob. and App.*, 8:22–46, 1963.

[19] A. Wald. *Sequential Analysis.* New York: Wiley, 1947.