2011 American Control Conference
on O'Farrell Street, San Francisco, CA, USA
June 29 - July 01, 2011

A Case Study in Robust Quickest Detection for Hidden Markov Models

Aliaa Atwi Ketan Savla Munther A. Dahleh

Abstract— **We consider the problem of detecting rare events in a real data set with structural interdependencies. The real data set is modeled using hidden Markov models (HMMs), and rare event detection is viewed as a variant of the quickest detection problem. We assess the feasibility of two quickest detection frameworks recently suggested. The first method is based on dynamic programming and follows a Bayesian approach, and the second method is a non-Bayesian approximate cumulative sum (CUSUM) 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 CUSUM-based method when the rare event model is not exactly known but belongs to a known class of models.**

I. 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 detection delay, and the other being a measure of the frequency of false alarms [14].

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 are better modelled using hidden Markov models (HMMs). This paper considers one such application in the field of urban planning. Cell phone traffic data is available at little to no cost to city planners and can be used to detect emergency states. A timely response to emergencies is crucial to avert catastrophes, and false alarms may lead to unneeded costly measures. In addition, network traffic data reflects the periodic rythm of human activity, making HMM quickest detection a more suitable choice than i.i.d. alternatives.
We focus on two frameworks recently suggested for the HMM quickest detection problem; one is a Bayesian dynamic programming (DP) based framework [6], and the other is a non-Bayesian approximate CUSUM framework [5]. In this paper, we assess the feasibility of the suggested methods for detecting disruptions in real data sets.

An important consideration is robustness when the disruption model is not exactly known, which is often the case when dealing with rare events (abnormal states). A recent paper addresses this problem for i.i.d. observations [9]. We examine through simulations the robustness of the non-bayesian CUSUM-based framework suggested in [5] and suggest an experimental procedure to design minimax-robust HMM-CUSUM algorithms when the exact disruption model is unknown but belongs to a known class of models. The procedure is demonstrated on real data.

There is abundant work in the literature on anomaly detection in general, and some work on the problem for cellphone network data in particular. To name a few, [13] proposes a one step clustering algorithm to detect anomalies in streaming cellphone network data for emergency detection. [7] uses Markov-modulated Poisson processes and machine learning techniques to detect changes in data that reflects normal human activity. Finally, the reader is refered to [4], a survey of the field of anomaly detection and the various classes of methods it employs.

The contribution of our paper is threefold. First, we use quickest detection as a method for detecting rare events in real data. Quickest detection is backed by rigorous theoretical guarantees, and to the best of our knowledge, there are few, if any, case studies that focus on its application on HMM real data. Second, we compare two novel HMM quickest detection techniques based on their feasibility for real data applications. Finally, we provide guidelines for designing HMM-CUSUM procedures that guarantee minimax robustness when the disruption model is not exactly known but belongs to a class with finitely many models.

We describe the real data set in section II. Then, in section III, we discuss the implementation issues encountered when modeling the data set with HMMs. In section IV, we outline the DP-based Bayesian algorithm and focus on implementation challenges. In section V, we outline the CUSUM-based HMM quickest detection algorithm, show its real data performance through simulations, and address its robustness when the disruption model is not exactly known.

II. DESCRIPTION OF DATA SET

The time-series data consists of measurements of cell phone network traffic from the area surrounding Termini, Rome's busiest subway station. Network traffic is measured by the number of phone calls initiated in a 15 minute interval and is normalized to maintain anonimity. The data available spans a period of approximately 3 months. Figure 1 shows an average week of network traffic. The first 5 peaks corresponds to weekdays, and the shorter peaks that follow correspond to weekends. Within any day, we observe a pattern of rapid increase in communication activity from morning until mid-day followed by decrease in traffic in the later part of the day. Night time exhibits the lowest level of activity. Figure 2 shows 3 weeks of periodic data, with a disruption of the pattern in day 2 of the second week shown. The disruption corresponds to a train accident at Termini which lead to a surge in communication traffic at the time. Further details on this data set can be found in [8]. Our

A. Atwi, K. Savla and M. A. Dahleh are with the Laboratory for Information and Decision Systems at the Massachusetts Institute of Technology, Cambridge, MA, USA.{aliaa87,ksavla,dahleh}@mit.edu.

purpose is to detect this disruption using quickest detection techniques.

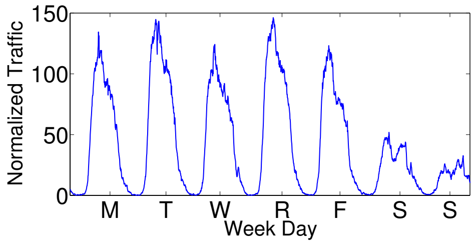


Fig. 1: An Average Week of Network Traffic

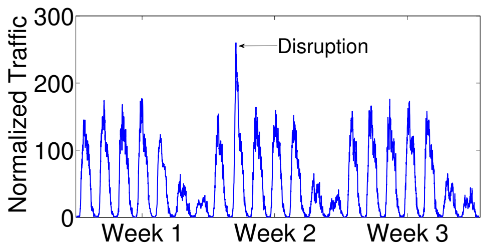


Fig. 2: Three Weeks of Data from Termini with a Surge in Cellphone Network Traffic around Termini on Tuesday of the Second Week

III. Data Mining and Statistical Modeling of Rome Data Using HMMs

In this section, we focus on implementation aspects of fitting an HMM in the business-as-usual Rome data. The reader can refer to [16] for a tutorial on HMMs.

We choose the emission probability distribution to be Poisson reflecting the nature of the data as counts of phone calls. Let A be the transition probability matrix of the underlying markov chain, N_s the number of underlying states, F the vector of initial state probabilities, and $B(i)$ the parameter of the Poisson random variable describing the observations under state i of the underlying Markov chain. Our aim is to solve for the model $\Omega = (A, B, F, N_s)$ that best describes a finite training sequence of business-as-usual data O.

There is no known way to analytically solve for the model Ω which maximizes $Pr(O \mid \Omega)$ [16]. However, for a fixed N_s and starting with an intial "guess" model $\Omega_1 = (A_1, B_1, F_1, N_s)$, we can find a model $\Omega_2 = (A_2, B_2, F_2, N_s)$ that locally maximizes $\Pr(O \mid \Omega)$ using an iterative procedure like Baum-Welch [10] (a variant of the EM Algorithm [11]). In this work, we used functions from the "mhsmm" package in the statistical computing language R [15] to implement the Baum-Welch algorithm for Poisson observations.

The choice of Ω_1 has a significant effect on the optimality of the final estimate, especially the choice of B [16]. In our implementation, we took all underlying states and transitions to be initially 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 O, apply the Baum-Welch algorithm on O for an increasing number of states N_s, yielding a locally optimal model $\Omega(N_s)$ for every N_s. $Pr(O \mid \Omega(N_s))$ can be taken as a measure of accuracy of fit for N_s states and can be calculated using the Forward Algorithm [3][2]. In practice, the forward variable "underflows" (heads exponentially to 0 for longer O) which can be resolved by scaling the forward variable by a factor independent of the underlying state, as described in [16].

Figure 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.

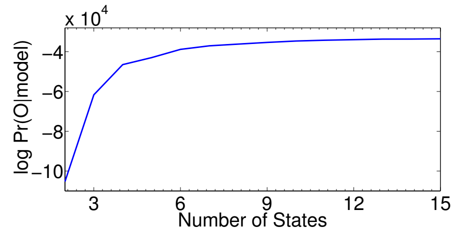


Fig. 3: $\log\{Pr[O \mid \Omega(N_s)]\}$ for Increasing Number of States of Underlying Model

The data set had missing samples spanning days at a time, making interpolation an unfavorable option, especially knowing the periodic daily pattern of variation. We addressed the issue by replacing missing samples with the average value corresponding to available samples on similar days and times of the day as the missing samples.

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

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. Here, X_t is 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 use the noisy observation sequence X to detect the change in the underlying unobservable sequence M as soon as possible while minimizing false alarms.

A. Bayesian Framework

The framework in this section was proposed 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 ($t < T$). The state space of the process

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

is partitioned into

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

First, note that Y is a Markov process with initial distribution $\eta = ((1-\theta_0)\mu, \theta_0\rho)$ and one-step transition matrix $P = \begin{bmatrix} (1-\theta)W_0 & \theta W_1 \\ 0 & W_1 \end{bmatrix}$. $\mathcal{Y}_1$ forms a recurrent class, and the states in $\mathcal{Y}_0$ are transient. The problem is to detect the time till absorption of Y in $\mathcal{Y}_1$:

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

Let C_D be the cost of each observation taken after T without detecting a change, and C_F the cost of a false alarm. The Bayes' risk associated with a decision rule τ is thus:

$$\mu(\tau) = C_D E\left[(\tau - T)^+\right] + C_F \Pr(\tau < T) \tag{1}$$

where expectation is taken over all possible sequences X and Y and change times T. The objective is to solve the following optimization problem:

$$\inf_{\text{causal } \tau} \mu(\tau) \tag{2}$$

B. Solution [6]

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

$$\Pi_{t+1} = \frac{\Pi_t P diag(f(X_{t+1}))}{\Pi_t P f(X_{t+1})} \tag{3}$$

where P is the one-step transition matrix, $f(X_{t+1})$ is the column vector of emission probabilities of X_{t+1} under $y \in \mathcal{Y}$, and $diag(f(X_{t+1}))$ 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:

$$g(\pi) = \sum_{y \in \mathcal{Y}_1} C_D \pi(y) \text{ and } h(\pi) = \sum_{y \in \mathcal{Y}_0} C_F \pi(y)$$

The optimal cost is a function of the initial state η; specifically, it is the value function evaluated at η of an optimal stopping problem over the Markov process Π. The value function J satisfies the following Bellman equation:

$$J(\pi) = \min\{h(\pi), g(\pi) + E_X\left[J(\pi') \mid \pi\right]\} \tag{4}$$

where π' is obtained from π through (3) for a given observation x. The reader can refer to [1] for the intuition behind equation (4).

An approximate solution for the infinite horizon Bellman equation 4 can be obtained by solving the problem for $N(\varepsilon)$ 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, and μ^* be the optimal cost of the infinite horizon problem. For every positive ε, there exists $N(\varepsilon)$ such that $\mu_{N(\varepsilon)} - \mu^* \leq \varepsilon$. More especifically,

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

$\mu_{N(\varepsilon)}$ 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:

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

where $k = 1, 2, \cdots, N(\varepsilon)$ and $J^0 := h$, and π_{t+1} can easily be calculated from π_t for a given value of X_{t+1} using (3).

C. ε-Optimal Alarm Time and Algorithm

The change detection algorithm suggested in [6] proceeds 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). 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. The region is a non-empty closed convex subset of the $2d$-dimensional simplex $\mathcal{P}$ that shrinks with increasing $N(\epsilon)$, and converges to Γ (the infinite horizon optimal stopping region) as $N \to \infty$ [6].

D. Implementation

Despite the favorable properties of region $\Gamma^{N(\varepsilon)}$, 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 π in the continuous higher-dimension probability simplex $\mathcal{P}$, which is computationally 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 $\Gamma^{N(\varepsilon)}$ in practice poses another challenge relating to the calculation of the Q-factor at stage k+1:

$E_{X_{t+1}}\left(J^k(\Pi_{t+1}) \mid \Pi_t = \pi_t\right)$. 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}. The countably infinite support of X_{t+1} makes the Q-factor calculation computationally intractable. 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:

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

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

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

E. Simulation Results

In section III, we found that the number of underlying states required to model the data set with "reasonable" accuracy is $N_s = 6$. But due to the exponential complexity of the algorithm in the number of states N_s, we start by modeling the data with a three state and a four state HMM. The resulting HMM parameters can be found in [1]. The change detection algorithm proceeds from the beginning of the disruption week which consists of 2016 samples. The disruption occurs at sample 406 (with a value of 181).

For $N_s = 3$, we obtain the detection times in Table I under different values of the false alarm cost (C_F). The results were obtained taking a discretization step size of 0.25, a sample cost C_D of 1, error margin of ε=3, and a prior $\theta = 1/300$. Notice how all the cost assignments considered result in a false alarm at $t = 405$. The traffic measurement at $t = 405$ was 171, a value at the boundary between between business-as-usual and disurption states. With a crude 3 state model, a higher penalty for false alarms is required to correctly resolve this boundary situation. The higher penalty, however, results in longer horizons which are computationally infeasible.

Table II shows the detection times for $N_s = 4$ under different false alarm costs C_F (keeping all other parameters constant). Notice that correct detection occured in the range of C_F considered when a more refined model ($N_s = 4$) was taken to describe the data. The fact that $N_s = 4$ was sufficient indicates that the difference between business-as-usual and disruption data in this particular data is pronounced enough that even a crude model ($N_s < 6$) was sufficient for quickest detection purposes.

For a lot of other real-world applications, the richness of the data and the subtle differences between business-as-usual and disruption necessitate a high number of underlying Markov states. The algorithm does not scale well with increasing N_s, and a different approach is needed.

TABLE I: Detection Time and Horizon for Different False Alarm Costs when $N_s = 3$

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

TABLE II: Detection Time and Horizon for Different False Alarm Costs when $N_s = 4$

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

V. NON-BAYESIAN APPROXIMATE CUSUM METHOD FOR HMM QUICKEST DETECTION

A non-Bayesian CUSUM-based approach for HMM quickest detection was suggested in [5]. It 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. CUSUM is commonly viewed as a repeated sequential probability ratio test (SPRT) with a lower threshold of zero, and an upper threshold chosen to capture the tradeoff between false alarm frequency and detection delay. For a more detailed description of CUSUM and SPRT, we refer the reader to [12] and [17] respectively.

A. HMM CUSUM-Like Procedure for Quickest Detection

Designing CUSUM procedures involves finding a relevant function $g(X_n)$ of the observations X_n that satisfies Page's recursion [12] and the antipodality condition, namely that $E(g(X_n)|H) \leq 0$ and $E(g(X_n)|K) \geq 0$), where K(H) is the hypothesis that change has (has not) occured. For the HMM-quickest detection problem, [5] proposes using the following form of $g(X_n)$. For the k'th sample after the last SPRT, and taking f_H and f_K to be the HMM probability measure before and after change (calculated using the scaled forward algorithm as described in [5]),

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

The CUSUM recursion is then

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

The stopping time is $\min\{n; S_n \geq h\}$, where h is a threshold chosen according to the desired tradeoff between false alarm frequency and detection delay. Average detection delay increases linearly with h, whereas the average delay between false alarms increases exponentially in h [5]. This "log-linear" behavior is the main reason why CUSUM algorithms work.

B. Simulation Results

In this section, we discuss the results of running HMM-CUSUM on the Rome data. We use a 6-state HMM description of the data (parameters can be found in [1]). Figure 4 shows the stopping times obtained for different values of h. Notice that small h ($h < 1.5$) leads to false alarms. As h increases, the time between false alarm increases exponentially, and for $1.5 \leq h \leq 1100$, HMM-CUSUM

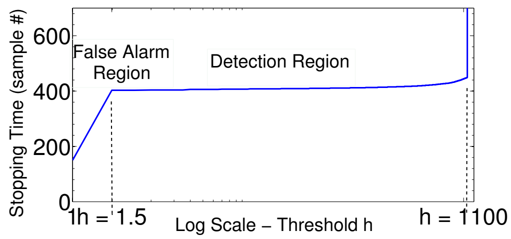


Fig. 4: Detection Time Versus h for 1 week Termini Data Fit with a 6-state HMM: $h \in (0, 1.6)$

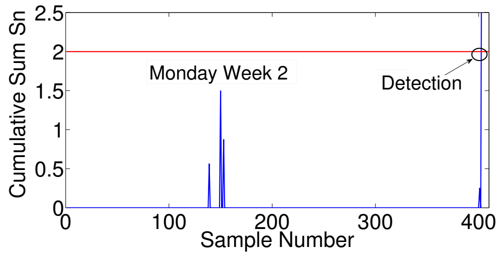


Fig. 5: HMM-CUSUM as a Repeated SPRT

consistently detects the disruption with linearly increasing delay. Beyond $h = 1100$, the disruption goes undetected (alarm at infinity).

This behavior can be explained by observing the trajectory of the cumulative sum S_n for the Rome data (figure 5). Initially, the data shows an overwhelming likelihood of being business-as-usual which leads to restarting the SPRT with $S_n < 0$ reset to 0. Around Monday's peak traffic time, the likelihood of disruption increases leading to a significant positive drift. The increase, however, is not enough to cross the threshold h and raise an alarm. Eventually, Tuesday's surge in traffic raises the likelihood of disruption to the point of crossing the threshold, leading to detection. Low values of h intercept the Monday peak leading to false alarms, and very high values of h intercept neither peak, leading to failed detection.

Figure 6 shows an approximate average alarm time for data sampled from the Termini disruption model under different values of h. The average alarm time for each h is obtained through Monte Carlo simulations. To guarantee detection, on average, the value of h required is greater than 7.5.

C. Robustness Results

First, consider the effect of designing HMM-CUSUM based on the Rome disruption model when the actual disruption is sampled from one of the models in table III which differ from the Rome model in that they have progressively smaller emission probability parameters. In figure 7a, we see

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

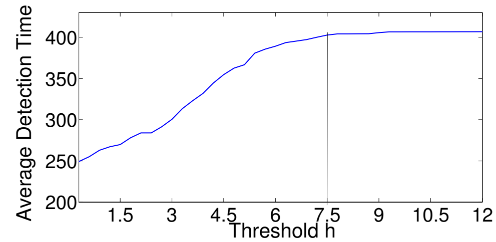


Fig. 6: Average Alarm Time Versus h for Locally Optimal 6-State Model Describing Rome Data

that this discrepancy between assumed and actual disruption models leads to a higher detection delay (less optimal) for $h = 7.5$ (compare with figure 6). Recall that $h = 7.5$ is the lowest threshold to guarantee "negligible" false alarm rate on average. Being the lowest such threshold, it also guarantees the smallest detection delay for the desired false alarm rate. The observed increase in detection delay can be explained as follows: when the actual emission probability parameters of the disruption model are less than the assumed ones, it takes more samples for S_n to show the magnitude of positive drift that would lead to crossing the threshold designed for higher emission probabilities.

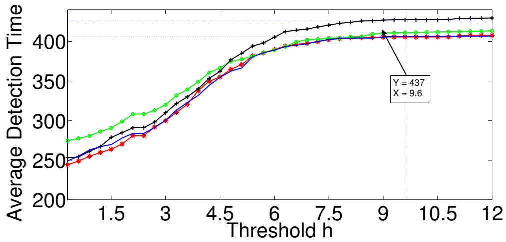
Second, we propose a simulation-based method to design robust HMM-CUSUM procedures when the disruption model is unknown but belongs to a known class $\mathcal{C}$ with $|C|$ models. We focus on minimizing average detection delay for a negligible false alarm rate. The design model is constrained to be in $\mathcal{C}$. Note that an HMM-CUSUM procedure is characterized by a threshold h and an assumed disruption model. We suggest the following method:

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

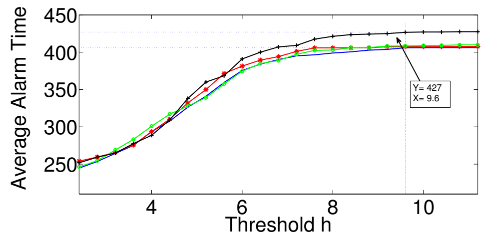
For the class $\mathcal{C}$ described in table III and the original model obtained from Termini, we obtain the plots in figures 7a-d. In figure 7a, we observe that when the HMM-CUSUM is designed for the Termini disruption, $h = 9.6$ guarantees a negligible false alarm rate for all models in $\mathcal{C}$. For this threshold, the worst average delay happens for data drawn from perturbed model 3 and has a value of 31 samples. Based on the simulation results (table IV), the design model of choice is perturbed Model 1 or 2 with a worst case detection delay of 21.

VI. CONCLUSIONS AND FUTURE WORK

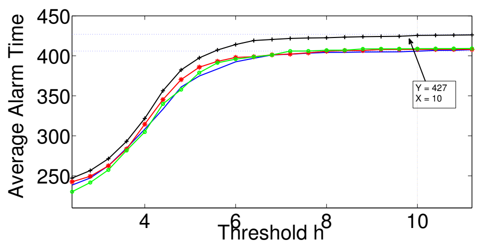
In this paper, we assessed the feasibility of two HMM quickest detection procedures for detecting rare events in a real data set in the context of urban planning.



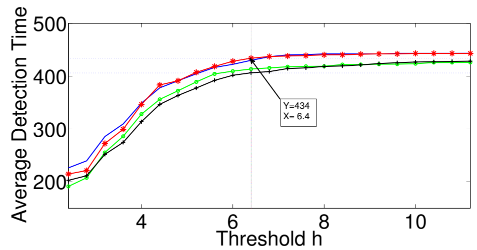
(a) Assumed model = Rome model



(b) Assumed model = Perturbed model 1



(c) Assumed model = Perturbed model 2



(d) Assumed model = Perturbed model 3

Fig. 7: Performance of HMM-CUSUM under different assumed and actual models in $\mathcal{C}$. (- = Termini model) (* = perturbed model 1) (o = perturbed model 2) (+ = perturbed model 3)

For the DP-based 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 suggested approximations do not scale well with increasing state-space dimension rendering the method infeasible for real applications. On the other hand, the performance guarantees on HMM-CUSUM [5] often involve approximations, but it is fast and scales well with increasing dimensions. Finally, we examined the robustness of the HMM-CUSUM when the disruption model is not exactly known, but belongs to a known class of HMMs.

Future work in this area will focus on finding theoretical results for the minimax-robust HMM quickest detection and finding rigorous notions of distance for HMMs compatible with the quickest detection problem.

Acknowledgement

This work was supported in part by NSF EFRI-ARES grant number 0735956. The authors thank the MIT Senseable City Lab for providing us the data for this paper.

References

[1] A. Atwi. A case study in robust quickest detection for hidden markov models. Master's thesis, Massachusetts Institute of Technology, 2010. http://web.mit.edu/aliaa87/www/MSthesis.pdf.

Assumed Model	h For False Alarm	Worst Average Alarm At h	Worst Delay Model
Termini	9.6	437	Pert. Model 3
Perturbed Model 1	9.6	427	Pert. Model 3
Perturbed Model 2	10	427	Pert. Model 3
Perturbed Model 3	6.4	434	Pert. Model 1

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

[2] 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.

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

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3):1–58, 2009.

[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] Alexander Ihler, Jon Hutchins, and Padhraic Smyth. Learning to detect events with markov-modulated poisson processes. *ACM Trans. Knowl. Discov. Data*, 1(3):13, 2007.

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

[9] 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.

[10] 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.

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

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

[13] Alec Pawling, Nitesh Chawla, and Greg Madey. Anomaly detection in a mobile communication network. *Computational amp; Mathematical Organization Theory*, 13:407–422, 2007. 10.1007/s10588-007-9018-7.

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

[15] 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.

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

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