Belief State Approaches to Signaling Alarms in Surveillance Systems

Kaustav Das
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
kaustav@cs.cmu.edu

Andrew Moore
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
awm@cs.cmu.edu

Jeff Schneider
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
schneide@cs.cmu.edu

ABSTRACT

Surveillance systems have long been used to monitor industrial processes and are becoming increasingly popular in public health and anti-terrorism applications. Most early detection systems produce a time series of p-values or some other statistic as their output. Typically, the decision to signal an alarm is based on a threshold or other simple algorithm such as CUSUM that accumulates detection information temporally.

We formulate a POMDP model of underlying events and observations from a detector. We solve the model and show how it is used for single-output detectors. When dealing with spatio-temporal data, scan statistics are a popular method of building detectors. We describe the use of scan statistics in surveillance and how our POMDP model can be used to perform alarm signaling with them. We compare the results obtained by our method with simple thresholding and CUSUM on synthetic and semi-synthetic health data.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search

General Terms

Algorithms, Experimentation

Keywords

Probabilistic Model, Scan Statistic, Signaling Alarms, Surveillance Systems

1. INTRODUCTION

Automatic surveillance systems are becoming more popular and are increasingly using data mining methods to perform detection. The observation of industrial manufacturing processes is one traditional application of these systems. Another application is public health monitoring, which has the goal of detecting new disease outbreaks as early as possible. Searching for terrorist activity or attacks is also becoming important. Applications in that area include monitoring human health and behavioral data to detect a chemical or biological attack, or searching for signs of radiation to detect development or deployment of nuclear devices. The RODS lab at the University of Pittsburgh (see www.health.pitt.edu/rods/) is focused both on public health monitoring and detection of biological attacks. This paper is based on our work in the RODS lab and thus focuses on these applications, but the algorithms we present are not specific to them. They are appropriate for a variety of monitoring tasks.

Modern surveillance systems are characterized by the need to analyze many variables simultaneously – in some cases hundreds or thousands of variables. Because of this fact, the traditional method of setting upper and lower bounds for a single variable are no longer appropriate. Data mining methods are used that must address the complex interactions between variables, the dangers of multiple hypothesis testing, and the computational issues caused by large data sets. See [9] for an overview of detection methods.

Typically, a detection algorithm will take a time-series of many variables as input and produce a time-series of p-values, or some other indication of alarm level, as output. Many of the detection algorithms use sophisticated means such as randomization testing and additional correction for multiple hypothesis testing to make their outputs as accurate as possible. Often, these outputs are followed up with simple thresholding to determine whether to signal an alarm and call for further investigation. Since the outputs are generated as a time series it makes sense to combine the information provided by them temporally in order to make better decisions about signaling alarms. One popular method of doing this is CUSUM and we describe that algorithm in the next section.

In this paper, we propose a probabilistic model of the process being monitored and the detection algorithm watching it. Based on those models we can determine the correct belief state for the underlying process and the optimal decision when considering the costs of signaling and alarm and allowing an event to go undetected. We compare this method to CUSUM and thresholding on synthetic data and show its superior performance. Secondly, we describe scan statistics and how they are used to construct detection algorithms on spatio-temporal data. We then show how our probabilistic model is used with scan statistics and present empirical

KDD'04, August 22–25, 2004, Seattle, Washington, USA.

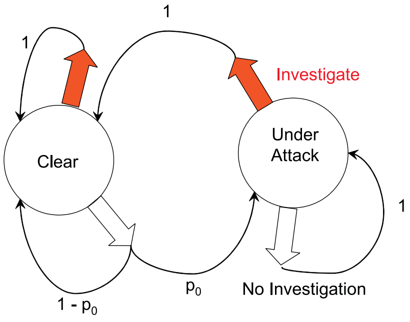

Figure 1: Two state model of Biological attack

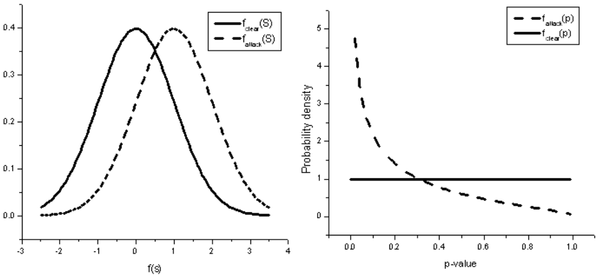

Figure 2: The clear (solid) and attak (dashed) state distributions of the underlying variable s and p-values

results on synthetic and semi-synthetic health data.

2. CUSUM

Before presenting our algorithm, we describe a popular method used in signaling alarms. CUSUM was originally developed to detect changes in the quality of output of continuous production process. It can quickly detect a shift in the mean of a process. As the name suggests, CUSUM maintains a cumulative sum of deviations from a reference value r. Let us consider a time series where at time i we have measurement X_i. The one-sided CUSUM calculation is as follows:

$$C_0 = 0$$

$$C_m = max(0, X_m - (\mu_0 + K) + C_{m-1})$$

μ_0 is the in-process mean. From the equations above, if the X_m values are close to the mean, then the C_m values will be some small value. However once a positive shift from the mean occurs, the C_m value will increase rapidly. K is known as the slack value or allowance. In the equation above, any values within K units of μ_0 will be effectively ignored. The allowance K is usually set to be the midpoint between the in control process mean μ_0 and the out-of-control process mean μ_1.

Alerts are raised whenever C_m exceeds a threshold decision interval H. The Average Run Length (ARL) is controlled by this parameter. The ARL is the average number of timesteps before an alert is raised.

The CUSUM algorithm described here has been extensively used in biosurvelliance systems. It has been used for influenza survellience [8], detection of salmonella outbreaks [3] and in the Early Aberration Reporting system [2]. CUSUM algorithms have also been extended to incorporate spatial information such as [6] and [7].

3. POMDP MODEL OF AN ATTACK

We assume a Markov Decision Process model for a terrorist attack scenario. Again, we note that the model is not specific to this type of monitoring application. Fig 1 shows the two state model considered for the attack. The first state represents a situation when there is no attack and the second state represents the situation when there is an attack. There are two possible actions at each state, either to investigate or to not investigate. These are represented by the block arrows. If we are in the clear state and choose not to investigate then there is a prior probability p_0 of an attack occurring on that day. At any state if we choose to investigate, we return to the clear state. This action can be thought of as a reset action where the state is reset to clear. And of course, in the attack state if we do not investigate, we remain in that state.

There is a cost associated with each action. There is a cost of investigation which is same for both the states. If we are in the attack state, and we do not investigate, then there is an associated cost (cost of false negative). In the clear state, if we choose not to investigate, we do not incur any cost.

The problem stems from the fact that at a particular instant of time, we do not know our current state. Instead, at each time instant we percieve an observation which is dependent on the underlying state. The observation in our case is some informative statistic that is the output of a detection algorithm. This makes the system a Partially Observable MDP (POMDP) (see [4]). In order to fully characterize the POMDP, we need to know the exact distribution of this statistic under each state.

For our purposes we use the p-value that is usually output by a detection algorithm. By definition, the distribution of the p-values under the null hypothesis (there is no attack) should be a uniform 0-1 distribution. Let $f_{clear}(p)$ be this distribution. We need to determine $f_{attack}(p)$, the distribution of p-values under the alternate hypothesis, ie., in the case when there is an attack. For our experiments we have derived this distribution by assuming some form of an underlying distribution. First consider that there is some underlying variable s that is normally distributed. Under the null hypothesis (clear state), it has mean 0, and variance 1. Under the alternate hypothesis (attack state), it has mean M and variance 1. These two distributions are shown in Fig 2. It is possible to derive the distribution of p-values under the alternate hypothesis from these distributions. $f_{clear}(p)$ and $f_{attack}(p)$ distributions obtained from the gaussian assuption is shown in Fig 2. The parameter M can be varied to obtain different distributions of $f_{attack}(p)$. As M increases, the distribution $f_{attack}(p)$ becomes more skewed towards p=0. Apart from using the normal distribution, we also consider the gamma distribution for the underlying variable.

3.1 Belief state representation

We now solve this POMDP using value iteration over the belief states. The POMDP is first converted into a belief state MDP. Each state of the MDP represents a particular belief that we are in the attack state. Since the belief is a real value ranging from 0 to 1, we have a continuous state MDP. To use value iteration, we discretize this state space into N discrete states labelled 0 to N-1. The state i represents a belief $\frac{i}{N-1}$ that we are in the attack state. Typically we use N=100 in our experiments.

3.2 Belief state update equation

We start with a particular belief that we are under attack. At each time step, we get an observation p, and we update our belief state. Let us assume that at time instant t we are in the belief state i_t, where $\frac{x_t=i}{N-1}$ is the belief that we are under attack. From our model, we know that there is a prior probability p_0 of there being an attack on that day. So our apriori belief that we are under attack is $x' = x_t + (1-x_t) * p_0$. The posterior belief $x_{t+1} = \frac{x' f_{attack}(p)}{x' f_{attack}(p) + (1-x') f_{clear}(p)}$.

3.3 Estimating the Transition Probabilities

The transition probability matrix is determined by a random simulation as follows:

- We start in the belief state i, and randomly choose a p value from the distribution $P_i(clear) * f_{clear}(p) + P_i(attack) * f_{attack}(p)$. Here $P_i(attack)$ is the belief that we are in the attack state in the belief state i, and $P_i(clear)$ is the belief that we are in the clear state.

- Then we update our belief state according to the update equation.

- The proportion of time we reach state j starting from state i gives an estimate of p_{ij}, the probability of transition from state i to j.

- We repeat this procedure for each belief state to determine all the p_{ij} values

The cost function for the belief state MDP can be defined as

$$C(b(x), A_j) = (1-x) * C(Clear, A_j) + x * C(Attack, A_j)$$

Here b(x) is the belief state where our belief that we are under attack is x. A_j is the action, either to investigate, or to not investigate.

3.4 Solving the MDP

We now use the standard value iteration algorithm to solve the MDP. Due to the structure of our two state POMDP, the optimal solution of the belief state MDP has a particular property. There exists a threshold belief $b_{threshold}$, such that in all the states corresponding to a belief less than $b_{threshold}$, the optimal action is to not investigate. And the optimal action is to investigate in all the states that correspond to a belief more than $b_{threshold}$.

4. EXTENSIONS TO THE MODEL

In our model we have assumed a fixed distribution of the observation parameter once an attack has occured. But in case of an actual attack we might expect that after the attack has occured, the distribution of the observed parameter keeps changing with time. For example the symptoms get more pronounced with time, and the deviation from the clear state distribution becomes more marked. Also the cost of not detecting the attack might not vary linearly with the number of days the attack is not detected. These considerations led us to introduce additional states in our basic two state model. The difference between two successive days when an attack has occured is going to be most pronounced in the case of the first day as compared to the later days. So we split the attack state into two states. One corresponds to the first day of the attack, and the other corresponds to when the attack is in progress for more than a day. This model is shown in Fig 3.

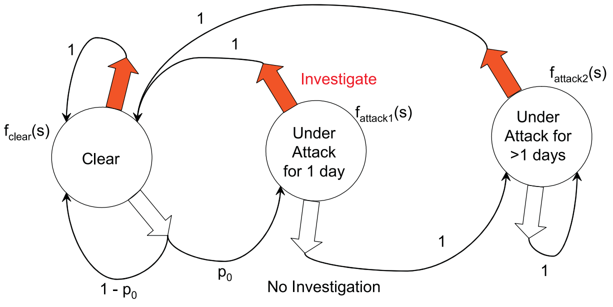

Figure 3: Three state model of Biological attack

We have assumed different distributions of the p-values for the three states. Also the cost of not investigating while under attack is different in the two cases. Here we have assumes a lower false negatve cost for the first day than the cost when it is underway for more than a day.

It is also possible to extend this model by introducing additional states for the 2nd day and so on. The complexity of solving the POMDP increases exponenially with the number of states. So we have done empirical tests only upto 3 states.

5. SIMULATION OF THE MODEL

We simulate the two state model starting from the clear state and going to the attack state with a probability p_0 on any day. Once in the attack state, we remain there until alarm is signalled. The state is then reset to clear. The observations (p-values) are generated from either of the two distributions $f_{attack}(p)$ or $f_{clear}(p)$ depending on the current state.

We use Thresholding, CUSUM and the MDP based solution independently to signal alarm. In order to evaluate the performance of the algorithms, we measure the number of false positives and the number of days till the attack was detected. To obtain this AMOC curve, we need to control the false positive rate of the diffrent procedures through a parameter. In Thresholding, we vary the Threshold p-value below which alarm is signalled. In CUSUM, we vary the decision interval H to obtain different false positive rates. And in the case of the MDP solution, we vary the ratio of the cost of false negative to the cost of investigation.

The simulation is carried on for a 100 year period with 1 day as the unit of time. We have done 100 random runs of this stretch to determine the confidence intervals. The number of false positives per year is plotted against the average number of days required to detect an attack.

6. RESULTS

These results from the simulated data indicate that CUSUM

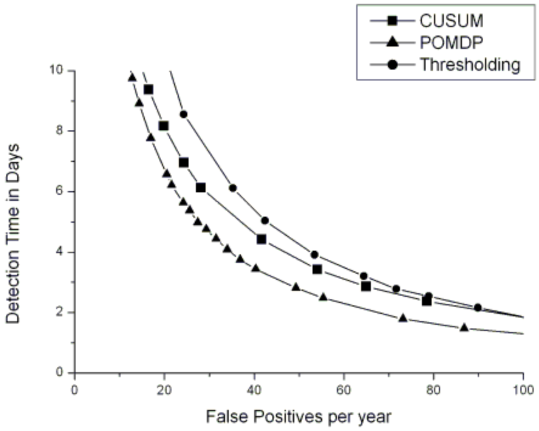

Figure 4: Plot of Detection Time vs False Positives assuming $p_0 = 0.005$

Table 1: Area under the AMOC curves for 2 state model with 95% confidence intervals

Distr.	p_{select}	POMDP	CUSUM	Thresholding
gaussian	0.01	213.68 ±1.023	282.48 ±2.30	323.28 ±2.389
gaussian	0.005	219.10 ±1.056	292.56 ±2.355	337.08 ±2.467
gamma	0.01	205.81 ±0.985	265.94 ±2.156	280.52 ±2.061

Table 2: Area under the AMOC curve for 3 state model with 95% confidence intervals

Distr.	p_{select}	POMDP	CUSUM	Thresholding
gamma	0.01	208.39 ±0.981	264.51 ±2.145	280.64 ±2.062

performs better than p-value thresholding. The MDP based approach outperforms both the other methods. For all the experiments using gaussian distribution for the underlying variable, we have fixed the mean in the clear state $mean_{clear}$ $= 0$. Also the standard deviations are set to 1. Fig 4 shows the result when the prior probability of attack $p_0 = 0.005$ and the mean for the underlying variable in the attack state $mean_{attack} = 0.2$.

In our second experiment, we used all the same values for the parameters as in the previous case, but increased the prior probability of attack (p_0) to 0.01. We then calculated the area under the corresponding AMOC curves. The values are shown in row 2 of Table 1.

In our third experiment we used a Gamma distribution for the underlying variable. The parameters for the clear state distribution are taken as $\alpha = 2$, $\beta = 1$, and those for the attack state are $\alpha = 2$, $\beta = 0.85$. Row 3 of Table 1 gives the area under the AMOC curves.

In the final experiment we used the three state model described in section 4. The underlying variable is assumed to have gamma distribution. For the clear state the parameters are $\alpha = 2$, $\beta = 1$. $f_{attack1}(p)$ has the parameters $\alpha = 2$, β $= 0.9$, and those for $f_{attack2}(p)$ are $\alpha = 2$, $\beta = 0.85$. The results are shown in Table 2.

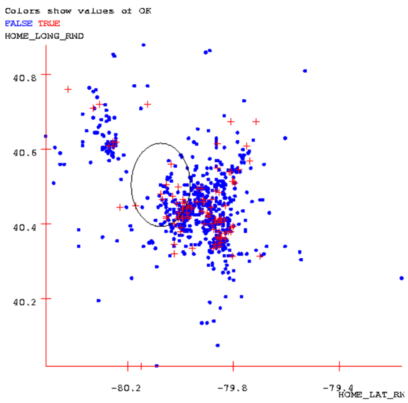

Figure 5: Sample scan statistic application.

7. APPLICATION TO SCAN STATISTICS

Consider the plot in Fig 5. Each point shows the location of a patient arriving in the emergency department[1]. The crosses mark points with a particular symptom of interest such as respiratory problems. We are interested in determining whether there is some region within this data (such as the circle shown in the plot) that has a higher incidence rate of the symptom of interest. This is a typical spatial scan statistic application. Studies of this sort are common in the field of public health and are used to determine whether environmental factors are causing higher disease rates in certain areas. In our case, we are interested in early detection of a bio-terrorist attack, which under several delivery mechanisms including airborne, may be clustered spatially. The algorithm for computing the scan statistic is as follows (adapted from [1, 5]):

1. Compute the likelihood of the data given the hypothesis that the incidence rate is uniform everywhere. If we assume that the marks are Bernoulli random variables having exactly the probability that appears in the data, then the likelihood of the data is as follows:

$$L_0 = \left(\frac{N^+}{N}\right)^{N^+} * \left(\frac{N^-}{N}\right)^{N^-} \qquad (1)$$

   where N is the total number of data points and N^+ and N^- are the number of positive and negative instances respectively.

2. For each candidate region, W, compute the likelihood that the incidence rate has one uniform value inside that region and another outside it. For Bernoulli random variables, the likelihood of the data is as follows:

$$\begin{aligned} L_W \;=\; & \left(\frac{N^+(W)}{N(W)}\right)^{N^+(W)} * \left(\frac{N^-(W)}{N(W)}\right)^{N^-(W)} * \\ & \left(\frac{N^+(\overline{W})}{N(\overline{W})}\right)^{N^+(\overline{W})} * \left(\frac{N^-(\overline{W})}{N(\overline{W})}\right)^{N^-(\overline{W})} \qquad (2) \end{aligned}$$

[1] This data comes from emergency departments in the Pittsburgh area. The data has been anonymized and the locations have significant noise added for further privacy protection.

where the W and $\overline{W}$ in parentheses indicate the respective counts of points inside and outside the region W. The space of regions to be considered will be discussed later.

3. For each region, compute the likelihood ratio: L_W/L_0.

4. Find the largest likelihood ratio. This is the scan statistic, which we call S_W. Also report the region, W, which yielded the maximum likelihood ratio.

Having computed the scan statistic for this example, we now turn to the question of what the null distribution of the statistic is under the assumption that there truly is a single uniform incidence rate over the whole data set. We simulate the null distribution by randomly shuffling the marks on the dataset and recomputing the scan statistic.

We can also calculate the p-value of each region by comparing its likelihood ratio with the samples obtained in step 3 of the randomization. In our experiments we choose a set of random $n_{regions}$ number of regions.

This is a spatial version of scan statistics. During survellience, we obtain new emergency department data each day. We can run this algorithm daily on that data. The simplest way to detect if an attack has occured is to signal an alarm whenever the p-value of any region is below a particular threshold value $p_{threshold}$. This corresponds to the p-value thresholding as described in the introduction.

Alternatively, we can use CUSUM to signal alerts. This can be done in two ways. Each day, we obtain a p-value for each region under consideration. We choose the minimum p-value p_{min} as the p-value for that day. We then run CUSUM with these p-values as the observations. We call this plain CUSUM. Another possibility is to run CUSUM in parallel for each region. We have $n_{regions}$ instances of CUSUM, where each corresponds to a particular region. The p-value of a region on any day is the observation X_i used by CUSUM. We signal an alert when any one of these CUSUM values goes above the predetermined threshold. This method will be refered to as regionwise CUSUM.

We also apply our belief state based approach on these p-values. Similar to the regionwise CUSUM, we maintain $n_{regions}$ different belief values, each corresponding to the belief that a particular region is under attack. Each day, these belief values are updated using the p-value of that region on that day.

8. EMPIRICAL TESTS

The base data set used for these experiments has the following attributes:

- HOME_LAT_RND, HOME_LONG_RND: Longitude and latitude of the patient's home (again note that coarse rounding and significant noise was added to these values to protect the patient's privacy).

- ADMIT_DAY_INDEX: Date on which the patient was admitted.

- PRODROME: The main category of the patient's complaint upon arrival at the emergency department.

We used two datasets for the experiments.

1. The first is a purely synthetic dataset. We randomly generate the HOME_LAT_RND, HOME_LONG_RND, ADMIT_DAY_INDEX values for a point. The PRODROME value is then randomly assigned such that with 0.1 probabily it is equal to $PRODROMEVAL_{select}$. $PRODROMEVAL_{select}$ is the particular PRODROME value that is of our current interest. This dataset spans over 150 days.

2. The second dataset is a real emergency department dataset from the regions around Pittsburgh. The data spans about 500 days.

These data contain no unnatural outbreaks. So in order to test our algorithms we must introduce aritificial outbreaks. In this section we use an outbreak simulation based on modified versions of our data set that matches the modeling assumptions we make in the previous section.

We use the two state model described in section 3. We start from the clear state and on any day there is a probability p_0=0.08 that there is an attack. Once in the attack state, we remain there until alarm is signalled. The state is then reset to clear. In the clear state, we use the part of the original data corresponding to that day. If we are in the attack state, the data is modified as follows:

1. Select the part of the original data set that corresponds to that day. Make a copy retaining only the location attributes (HOME_LAT_RND, HOME_LONG_RND and ADMIT_DAY_INDEX).

2. Choose a region, W, at random. The region is chosen only once for each attack. We do not change the attack region for successive days under attack.

3. For each data point, choose a data point from the original data set whose PRODROME attribute will be copied over to this point in the new data set.

   The selection strategy is to select with some probability, p_{select}, a data point that has a particular value, $PRODROMEVAL_{select}$ of the PRODROME attribute. With probability $1 - p_{select}$ select a data point uniformly at random. This selection strategy will be applied with different parameters for points inside and outside the chosen region W, in order to create a different distribution for each.

For CUSUM and belief state approach, we need to determine the $f_{clear}(p)$ and $f_{attack}(p)$ distributions. In the clear state, the p-values generated for each region does not correspond to the true p-value under the null hypothesis. This is because, we compare the likelihood of each region against the likelihood of the most significant region under the randomizations. So in this case $f_{clear}(p)$ is not an uniform distribution, but is heavily skewed towards p=1. These distributions are learnt from the data. We make an initial pass on the data when we output the p-values of the clear and attacked regions. We estimate the distributions from these values. In a real scenario, historical data can be used to learn these distributions.

For the purpose of our experiments, we need not solve the POMDP explicity. The solution to the POMDP model gives a belief threshold, $b_{threshold}$ as described in section 3.4. To plot the AMOC curves shown here, we can vary this threshold to obtain different points on the curve.

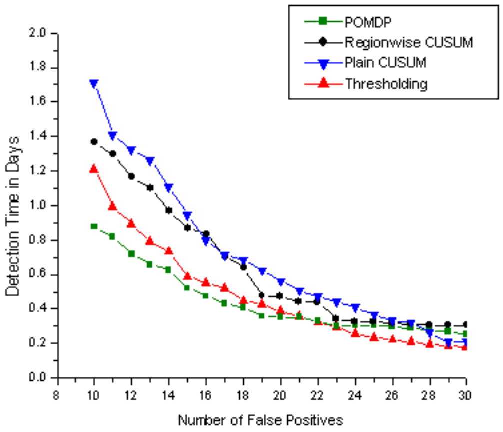

Figure 6: Plot of Detection time vs False Positives for Dataset 1, with p_{select}=0.1

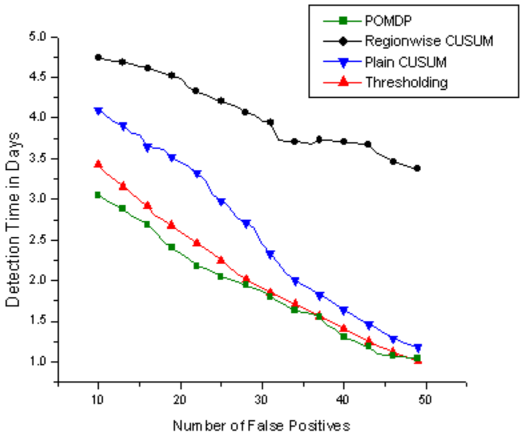

Figure 7: Plot of Detection time vs False Positives for Dataset 2, with p_{select}=0.2

The results involving dataset 1 are shown in Fig 6 and Table 3. Table 3 gives the area under corresponding AMOC curves for different values of p_{select}. Fig 7 and Table 4 shows the corresponding results for dataset 2.

The results for dataset 1 indicate that the detection time for the belief state based approach is significantly smaller than that for the other approaches. We see that the regionwise CUSUM does not perform any better than thresholding, and the plain CUSUM approach actually does much worse than thresholding. This might be because of the fact that the distribution of the observed p-values is heavily skewed. CUSUM works better when these distributions can be approximated by the normal distribution.

The results obtained on dataset 2 are similar. The POMDP approach again outperforms all the other approaches. CUSUM does not give much improvement over thresholding.

9. FUTURE WORK

Table 3: Area under the AMOC curves with 95% confidence intervals for Dataset 1

p_{select}	POMDP	Regionwise CUSUM	Plain CUSUM	Thresholding
0.1	25.69 ±23.26	36.20 ±31.70	42.03 ±30.04	32.83 ±25.18
0.2	3.05 ±2.96	4.46 ±3.35	5.21 ±4.10	3.76 ±3.52

Table 4: Area under the AMOC curves with 95% confidence intervals for Dataset 2

p_{select}	POMDP	Regionwise CUSUM	Plain CUSUM	Thresholding
0.1	119.31 ±21.98	143.37 ±21.32	120.07 ±24.65	131.22 ±30.73
0.2	76.10 ±14.27	161.82 ±24.31	102.54 ±21.18	81.79 ±19.32
0.4	43.31 ±7.68	68.85 ±10.13	87.43 ±17.97	58.35 ±12.67

We have assumed that successive p-values are independent. But, this assumption might not hold for many detection algorithms. In that case we need to take the dependency into account. Since the nature of dependency will depend heavily on the actual algorithm used for detection, this issue has to be addressed with regard to specific algorithms.

The procedure described here can be extended to include multiple detection algorithms. It might be used to consolidate the output of many detection algorithms to determine when to signal an alarm.

Also, while developing this method, we have assumed some definite distributions for an underlying variable. We need to evaluate the performance of the method if our assumption is not correct. As already mentioned, in a real life scenario, we might learn the underlying distribution from historical data.

10. REFERENCES

[1] J. Glaz and N. Balakrishnan. *Scan Statistics and Applications.* Birkhauser, 1999.

[2] L. Hutwagner, W. Thompspn, G. M. Seeman, and T. Treadwell. The bioterrorism preparedness and response early aberration reporting system(ears). *Journal of Urban Health*, 80:i89–i96, 2003.

[3] L. C. Hutwagner, E. Maloney, N. H. Bean, L. Slutsker, and S. Martin. Using laboratory-based surveillance data for prevention: An algorithm for detecting salmonella outbreaks. *Emerging Infectious Diseases*, 3:395–400, 1997.

[4] L. Kaelbling, M. Littman, and A. Cassandra. Planning and acting in paritally observable stochastic domains. *Artificial Intelligence*, 1997.

[5] M. Kulldorff. A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26:1481–1496, 1997.

[6] R. F. Raubertas. An analysis of disease surveillance data that uses the geographic locations of reporting units. *Statistics in Medicine*, 8:267–271, 1989.

[7] P. A. Rogerson. Surveillance systems for monitoring the development of spatial patterns. *Statistics in Medicine*, 16:2081–2093, 1997.

[8] H. E. Tillett and I. L. Spencer. Influenza survellience in england and wales using routine statistics. *Journal of Hygine*, 88:83–94, 1982.

[9] W.-K. Wong. *Data Mining for Early Disease Outbreak Detection.* PhD thesis, Carnegie Mellon University, 2004.