Journal of Machine Learning Research 6 (2005) 1961–1998 Submitted 8/04; Revised 3/05; Published 12/05

What's Strange About Recent Events (WSARE): An Algorithm for the Early Detection of Disease Outbreaks

Weng-Keen Wong WONG@EECS.OREGONSTATE.EDU
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97330, USA

Andrew Moore AWM@CS.CMU.EDU
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Gregory Cooper GFC@CBMI.PITT.EDU
Michael Wagner MMW@CBMI.PITT.EDU
Center For Biomedical Informatics
University of Pittsburgh
Pittsburgh, PA 15213, USA

Editor: Dale Schuurmans

Abstract

Traditional biosurveillance algorithms detect disease outbreaks by looking for peaks in a univariate time series of health-care data. Current health-care surveillance data, however, are no longer simply univariate data streams. Instead, a wealth of spatial, temporal, demographic and symptomatic information is available. We present an early disease outbreak detection algorithm called What's Strange About Recent Events (WSARE), which uses a multivariate approach to improve its timeliness of detection. WSARE employs a rule-based technique that compares recent health-care data against data from a baseline distribution and finds subgroups of the recent data whose proportions have changed the most from the baseline data. In addition, health-care data also pose difficulties for surveillance algorithms because of inherent temporal trends such as seasonal effects and day of week variations. WSARE approaches this problem using a Bayesian network to produce a baseline distribution that accounts for these temporal trends. The algorithm itself incorporates a wide range of ideas, including association rules, Bayesian networks, hypothesis testing and permutation tests to produce a detection algorithm that is careful to evaluate the significance of the alarms that it raises.

Keywords: anomaly detection, syndromic surveillance, biosurveillance, Bayesian networks, applications

1. Introduction

Detection systems inspect routinely collected data for anomalies and raise an alert upon discovery of any significant deviations from the norm. For example, Fawcett and Provost (1997) detect cellular phone fraud by monitoring changes to a cell phone user's typical calling behavior. In intrusion detection systems, anomalies in system events might indicate a possible breach of security (Warrender et al., 1999). In a similar manner, we would like to tackle the problem of early disease outbreak detection, in which the disease outbreak can be due to either natural causes or a bioterrorist attack.

One of the challenges for early disease outbreak detection is finding readily available data that contains a useful signal (Tsui et al., 2001). Data sources that require definitive diagnosis of the disease, such as lab reports, can often be obtained several days to weeks after the samples are submitted. By that point, the outbreak may have already escalated into a large scale epidemic. Instead of waiting for definite diagnostic data, we can monitor pre-diagnosis data, such as the symptoms exhibited by patients at an Emergency Department (ED). In doing so, we risk increasing the false positive rate, such as mistakenly attributing an increase in patients exhibiting respiratory problems to an anthrax attack rather than to influenza. Nevertheless, we have a potential gain in timeliness of detection. This type of surveillance of pre-diagnosis data is commonly referred to as *syndromic surveillance* (Mostashari and Hartman, 2003; Sosin, 2003).

In our syndromic surveillance infrastructure, we have real-time access to a database of emergency department (ED) cases from several hospitals in a city. Each record in this multivariate database contains information about the individual who is admitted to the ED. This information includes fields such as age, gender, symptoms exhibited, home zip code, work zip code, and time of arrival at the ED. In accordance with the HIPAA Privacy Rule (45 CFR Parts 160 through 164, 2003), personal identifying information, such as patient names, addresses, and identification numbers are removed from the data set used in this research. When a severe epidemic sweeps through a region, there will obviously be extreme perturbations in the number of ED visits. While these dramatic upswings are easily noticed during the late stages of an epidemic, the challenge is to detect the outbreak during its early stages and mitigate its effects. We would also like to detect outbreaks that are more subtle than a large scale epidemic as early as possible.

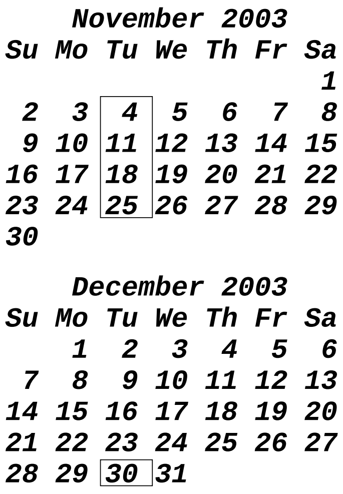
Although we have posed our problem in an anomaly detection framework, traditional anomaly detection algorithms are inappropriate for this domain. In the traditional approach, a probabilistic model of the baseline data is built using techniques such as neural nets (Bishop, 1994) or a mixture of naive Bayes submodels (Hamerly and Elkan, 2001). Anomalies are identified as individual data points with a rare attribute or rare combination of attributes. If we apply traditional anomaly detection to our ED data, we would find, for example, a patient that is over a hundred years old living in a sparsely populated region of the city. These isolated outliers in attribute space are not at all indicative of a disease outbreak. Instead of finding such unusual isolated cases, we are interested in finding *anomalous patterns*, which are specific groups whose profile is anomalous relative to their typical profile. Thus, in our example of using ED records, if there is a dramatic upswing in the number of children from a particular neighborhood appearing in the ED with diarrhea, then an early detection system should raise an alarm.

Another common approach to early outbreak detection is to convert the multivariate ED database into a univariate time series by aggregating daily counts of a certain attribute or combination of attributes. For instance, a simple detector would monitor the daily number of people appearing in the ED. Many different algorithms can then be used to monitor this univariate surveillance data, including methods from Statistical Quality Control (Montgomery, 2001), time series models (Box and Jenkins, 1976), and regression techniques (Serfling, 1963). This technique works well if we know beforehand which disease to monitor, since we can improve the timeliness of detection by monitoring specific attributes of the disease. For example, if we are vigilant against an anthrax attack, we can concentrate our efforts on ED cases involving respiratory problems. In our situation, we need to perform non-specific disease monitoring because we do not know what disease to expect, particularly in the case of a bioterrorist attack. Instead of monitoring health-care data for pre-defined patterns, we detect any significant anomalous patterns in the multivariate ED data. Furthermore, by taking a multivariate approach that inspects all available attributes in the data, particularly the temporal, spatial, demographic, and symptomatic attributes, we will show that such an approach can improve on the detection time of a univariate detection algorithm if the outbreak initially manifests itself as a localized cluster in attribute space.

Our approach to early disease outbreak detection uses a rule-based anomaly pattern detector called What's Strange About Recent Events (WSARE) (Wong et al., 2002, 2003). WSARE operates on discrete, multidimensional data sets with a temporal component. This algorithm compares recent data against a baseline distribution with the aim of finding rules that summarize significant patterns of anomalies. Each rule is made up of components of the form $X_i = V_i^j$, where X_i is the ith attribute and V_i^j is the jth value of that attribute. Multiple components are joined together by a logical AND. For example, a two component rule would be $Gender = Male$ AND $Home\ Location = NW$. These rules should not be interpreted as rules from a logic-based system in which the rules have an antecedent and a consequent. Rather, these rules can be thought of as SQL SELECT queries because they identify a subset of the data having records with attributes that match the components of the rule. WSARE finds these subsets whose proportions have changed the most between recent data and the baseline.

We will present versions 2.0 and 3.0 of the WSARE algorithm. We will also briefly describe WSARE 2.5 in order to illustrate the strengths of WSARE 3.0. These three algorithms only differ in how they create the baseline distribution; all other steps in the WSARE framework remain identical. WSARE 2.0 and 2.5 use raw historical data from selected days as the baseline while WSARE 3.0 models the baseline distribution using a Bayesian network.

2. What's Strange About Recent Events



```
    November 2003
Su Mo Tu We Th Fr Sa
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

    December 2003
Su Mo Tu We Th Fr Sa
    1  2  3  4  5  6
 7  8  9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31
```

Figure 1: The baseline for WSARE 2.0 if the current day is December 30, 2003

The basic question asked by all detection systems is whether anything strange has occurred in recent events. This question requires defining what it means to be recent and what it means to be strange. Our algorithm considers all patient records falling on the current day under evaluation to be recent events. Note that this definition of recent is not restrictive – our approach is fully general and recent can be defined to include all events within some other time period such as over the last six hours. In order to define an anomaly, we need to establish the concept of something being normal. In WSARE version 2.0, baseline behavior is assumed to be captured by raw historical data from the same day of the week in order to avoid environmental effects such as weekend versus weekday differences in the number of ED cases. This baseline period must be chosen from a time period similar to the current day. This can be achieved by being close enough to the current day to capture any seasonal or recent trends. On the other hand, the baseline period must also be sufficiently distant from the current day. This distance is required in case an outbreak happens on the current day but it remains undetected. If the baseline period is too close to the current day, the baseline period will quickly incorporate the outbreak cases as time progresses. In the description of WSARE 2.0 below, we assume that baseline behavior is captured by records that are in the set *baseline_days*. Typically, *baseline_days* contains the days that are 35, 42, 49, and 56 days prior to the day under consideration. We would like to emphasize that this baseline period is only used as an example; it can be easily modified to another time period without major changes to our algorithm. In Section 3 we will illustrate how version 3.0 of WSARE automatically generates the baseline using a Bayesian network.

We will refer to the events that fit a certain rule for the current day as C_{recent}. Similarly, the number of cases matching the same rule from the baseline period will be called $C_{baseline}$. As an example, suppose the current day is Tuesday December 30, 2003. The baseline used for WSARE 2.0 will then be November 4, 11, 18 and 25 of 2003 as seen in Figure 1. These dates are all from Tuesdays in order to avoid day of week variations.

2.1 Overview of WSARE

Parameter Name	Description	Default value
max_rule_components	Maximum number of components to a rule	2
num_randomizations	Number of iterations to the randomization test	1000
α_{FDR}	The significance level of the False Discovery Rate	0.05
baseline_days (WSARE 2.0 only)	Days to be used for the baseline	35, 42, 49, and 56 days prior to current date
environmental_attributes (WSARE 2.5 and 3.0)	Attributes that account for temporal trends	Not applicable
num_baseline_samples (WSARE 3.0 only)	The number of sampled records from the baseline Bayesian network	10000

Table 1: The main parameters in WSARE

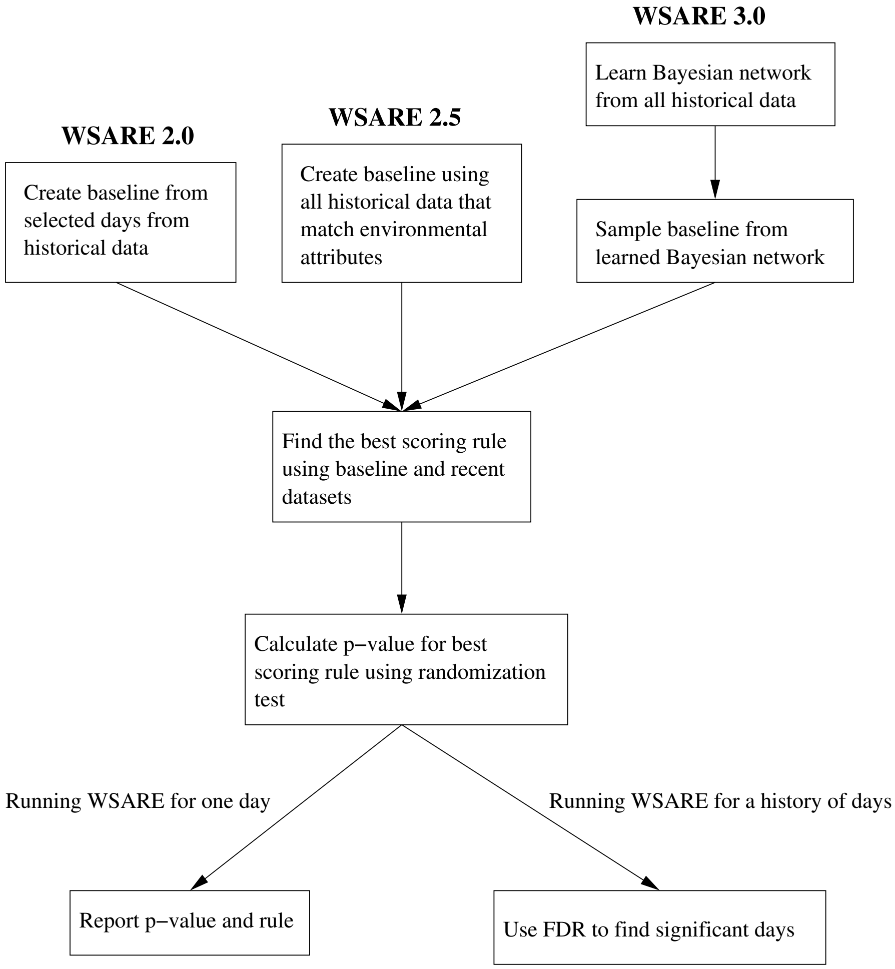

WSARE 3.0
Learn Bayesian network from all historical data
Sample baseline from learned Bayesian network
WSARE 2.5
Create baseline using all historical data that match environmental attributes
WSARE 2.0
Create baseline from selected days from historical data
Find the best scoring rule using baseline and recent datasets
Calculate p−value for best scoring rule using randomization test
Running WSARE for one day
Running WSARE for a history of days
Report p−value and rule
Use FDR to find significant days

Figure 2: A schematic overview of the steps involved in the WSARE algorithms

We will begin this section with an overview of the general WSARE algorithm followed by a more detailed example. Figure 2 gives a pictorial overview of the three WSARE algorithms discussed in this paper. Note that the three algorithms differ only in how they create the baseline while all of the other steps remain identical. Table 1 describes the main parameters used by the WSARE algorithms.

WSARE first finds the best scoring rule over events occurring on the current day using a greedy search. The limit to the number of components in a rule is set to the parameter *max_rule_components*, which is typically set to be 2 for computational reasons although in Section 2.5 we describe a greedy procedure for *n* component rules. The score of a rule is determined by comparing the events on the current day against events in the past. More specifically, we are comparing if the ratio between certain events on the current day and the total number of events on the current day differ dramatically between the recent period and the past. Following the score calculation, the best rule for that day has its p-value estimated by a randomization test. The p-value for a rule is the likelihood of finding a rule with as good a score under the hypothesis that the date and the other attributes are independent. The randomization-based p-value takes into account the effect of the multiple testing that occurs during the rule search. The number of iterations of the randomization test is determined by the parameter *num_randomizations*. If we are running the algorithm on a day-by-day basis we would end at this step. However, if we are looking at a history of days and we want to control for some level of false discoveries over this group of days, we would need the additional step of using the False Discovery Rate (FDR) method (Benjamini and Hochberg, 1995) to determine which of the p-values are significant. The days with significant p-values are returned as the anomalies.

2.2 One Component Rules

In order to illustrate this algorithm, suppose we have a large database of 1,000,000 ED records over a two-year span. This database contains roughly 1370 records a day. Suppose we treat all records within the last 24 hours as "recent" events. In addition, we can build a baseline data set out of all cases from exactly 35, 42, 49, and 56 days prior to the current day. We then combine the recent and baseline data to form a record subset called DB_i, which will have approximately 5000 records. The algorithm proceeds as follows. For each day i in the surveillance period, retrieve the records belonging to DB_i. We first consider all possible one-component rules. For every possible attribute-value combination, obtain the counts C_{recent} and $C_{baseline}$ from the data set DB_i. As an example, suppose the attribute under consideration is $Age\ Decile$ for the ED case. There are 9 possible values for $Age\ Decile$, ranging from 0 to 8. We start with the rule $Age\ Decile = 3$ and count the number of cases for the current day i that have $Age\ Decile = 3$ and those that have $Age\ Decile \neq 3$. The cases from five to eight weeks ago are subsequently examined to obtain the counts for the cases matching the rule and those not matching the rule. The four values form a two-by-two contingency table such as the one shown in Table 2.

2.3 Scoring Each One Component Rule

The next step is to evaluate the "score" of the rule using a hypothesis test in which the null hypothesis is the independence of the row and column attributes of the two-by-two contingency table. In effect, the hypothesis test measures how different the distribution for C_{recent} is compared to that of $C_{baseline}$. This test will generate a p-value that determines the significance of the anomalies found by the rule. We will refer to this p-value as the *score* in order to distinguish this p-value from the p-value that is obtained later on from the randomization test. We use the Chi Square test for independence of variables whenever the counts in the contingency table do not violate the validity of the Chi Square test. However, since we are searching for anomalies, the counts in the contingency table frequently involve small numbers. In this case, we use Fisher's exact test (Good, 2000) to find the score for each rule since the Chi Square test is an approximation to Fisher's exact test when counts are large. Running Fisher's exact test on Table 2 yields a score of 0.025939, which indicates that the count C_{recent} for cases matching the rule $Home\ Location = NW$ are very different from the count $C_{baseline}$. In biosurveillance, we are usually only interested in an increase in the number of certain records. As a result, we commonly use a one-sided Fisher's exact test.

	C_{recent}	$C_{baseline}$
$Home\ Location = NW$	6	496
$Home\ Location \neq NW$	40	9504

Table 2: A Sample 2x2 Contingency Table

2.4 Two Component Rules

At this point, the best one component rule for a particular day has been found. We will refer to the best one component rule for day i as BR_i^1. The algorithm then attempts to find the best two component rule for the day by adding on one extra component to BR_i^1 through a greedy search. This extra component is determined by supplementing BR_i^1 with all possible attribute-value pairs, except for the one already present in BR_i^1, and selecting the resulting two component rule with the best score. Scoring is performed in the exact same manner as before, except the counts C_{recent} and $C_{baseline}$ are calculated by counting the records that match the two component rule. The best two-component rule for day i is subsequently found and we will refer to it as BR_i^2

Suppose BR_i^1 has as its first component the attribute-value pair $C_1 = V_1$. Furthermore, let BR_i^2's components be $C_1 = V_1$ and $C_2 = V_2$. Adding the component $C_2 = V_2$ to BR_i^1 may not result in a better scoring rule. During our search for the best scoring two component rule, we only consider two component rules in which adding either component has a significant effect. Determining if either component has a significant effect can be done through two hypothesis tests. In the first hypothesis test, we use Fisher's exact test to determine the score of adding $C_2 = V_2$ to the one component rule $C_1 = V_1$. Similarly, in the second hypothesis test, we use Fisher's exact test to score the addition of the component $C_1 = V_1$ to $C_2 = V_2$. The 2-by-2 contingency tables used by the two hypothesis tests are shown in Table 3.

Records from Today with $C_1 = V_1$ and $C_2 = V_2$	Records from Other with $C_1 = V_1$ and $C_2 = V_2$
Records from Today with $C_1 \neq V_1$ and $C_2 = V_2$	Records from Other with $C_1 \neq V_1$ and $C_2 = V_2$

Records from Today with $C_1 = V_1$ and $C_2 = V_2$	Records from Other with $C_1 = V_1$ and $C_2 = V_2$
Records from Today with $C_1 = V_1$ and $C_2 \neq V_2$	Records from Other with $C_1 = V_1$ and $C_2 \neq V_2$

Table 3: 2x2 Contingency Tables for a Two Component Rule

Once we have the scores for both tables, we need to determine if they are significant or not. A score is considered significant if the result of a hypothesis test is significant at the $\alpha = 0.05$ level. If the scores for the two tables are both significant, then the presence of both components has an effect. As a result, the best rule overall for day i is BR_i^2. On the other hand, if any one of the scores is not significant, then the best rule overall for day i is BR_i^1.

2.5 *n* Component Rules

Let BR_i^{k-1} be the best $k-1$ component rule found for day i. In the general case of finding the best n component rule, the procedure is analogous to that of the previous section. Given BR_i^{k-1}, we produce BR_i^k by greedily adding on the best component, which is found by evaluating all possible attribute-value pairs as the next component, excluding those already present in components of BR_i^{k-1}. Starting with BR_i^1, we repeat this procedure until we reach BR_i^n.

In order to determine if the addition of a component is significant, we should in theory test all possible combinations of the n components. In general, we need $2\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i}$ such tests. Having this many tests is clearly computationally intensive as n increases. As an approximation, we resort to testing if adding the nth component is significant with respect to the $n-1$ other components. The two significance tests are as shown in Table 4, where $C_n = V_n$ refers to the last component added and $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ refers to the conjunction of the previous $n-1$ components. As before, if both of the Fisher's exact tests return a score less than $\alpha = 0.05$, then we consider the addition of the rule component significant. Due to this step, the probability of having a rule with many components is low because for each component added, it needs to be significant at the 95% level for both of the Fisher's exact tests.

Records from Today with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n = V_n$	Records from Other with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n = V_n$
Records from Today with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n \neq V_n$	Records from Other with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n \neq V_n$

Records from Today with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n = V_n$	Records from Other with $C_1 = V_1, \ldots, C_{n-1} = V_{n-1}$ and $C_n = V_n$
Records from Today with $\neg(C_1 = V_1, \ldots, C_{n-1} = V_{n-1})$ and $C_n = V_n$	Records from Other with $\neg(C_1 = V_1, \ldots, C_{n-1} = V_{n-1})$ and $C_n = V_n$

Table 4: 2x2 Contingency Tables for an N Component Rule

2.6 Finding the p-value for a Rule

The algorithm above for determining scores is prone to overfitting due to multiple hypothesis testing. Even if data were generated randomly, most single rules would have insignificant p-values but the best rule would be significant if we had searched over 1000 possible rules. In order to illustrate this point, suppose we follow the standard practice of rejecting the null hypothesis when the p-value is $< \alpha$, where $\alpha = 0.05$. In the case of a single hypothesis test, the probability of a false positive under the null hypothesis would be α, which equals 0.05. On the other hand, if we perform 1000 hypothesis tests, one for each possible rule under consideration, then the probability of a false positive could be as bad as $1-(1-0.05)^{1000} \approx 1$, which is much greater than 0.05 (Miller et al., 2001). Thus, if our algorithm returns a significant p-value, we cannot accept it at face value without adding an adjustment for the multiple hypothesis tests we performed. This problem can be addressed using a Bonferroni correction (Bonferroni, 1936) but this approach would be unnecessarily conservative. Instead, we use a randomization test. Under the null hypothesis of this randomization test, the date and the other ED case attributes are assumed to be independent. Consequently, the case attributes in the data set DB_i remain the same for each record but the date field is shuffled between records from the current day and records from five to eight weeks ago. The full method for the randomization test is shown below.

Let UCP_i = Uncompensated p-value i.e. the score as defined above.

```
For j = 1 to 1000
    Let $DB_i^{(j)}$ = newly randomized data set
    Let $BR_i^{(j)}$ = Best rule on $DB_i^{(j)}$
    Let $UCP_i^{(j)}$ = Uncompensated p-value of $BR_i^{(j)}$ on $DB_i^{(j)}$
```

Let the compensated p-value of BR_i be CPV_i i.e.

$$CPV_i = \frac{\text{\# of Randomized Tests in which } UCP_i^{(j)} < UCP_i}{\text{\# of Randomized Tests}}$$

CPV_i is an estimate of the chance that we would have seen an uncompensated p-value as good as UCP_i if in fact there was no relationship between date and case attributes. Note that we do not use the uncompensated p-value UCP_i after the randomization test. Instead, the compensated p-value CPV_i is used to decide if an alarm should be raised.

The bottleneck in the entire WSARE procedure is the randomization test. If implemented naively, it can be extremely computationally intense. In order to illustrate its complexity, suppose there are M attributes and each attribute can take on K possible values. In addition, let there be N_T records for today and N_B records for the baseline period. Note that typically, N_T is 4 to 20 times smaller than N_B. At iteration j of the randomization test, we need to search for the best scoring rule over $DB_i^{(j)}$. Assuming we limit the number of components in a rule to be two, searching for the best rule using a greedy search requires scoring $KM + K(M-1)$ rules. Scoring a rule requires us to obtain the entries for the two by two contingency table by counting over $N_T + N_B$ records. Thus, each iteration of the randomization test has a complexity of $(KM + K(M-1)) * (N_T + N_B)$. With Q iterations, the overall complexity of the randomization test is $O(QKM(N_T + N_B))$.

One of the key optimizations to speeding up the randomization test is the technique of "racing" (Maron and Moore, 1997). If BR_i is highly significant, we run the full 1000 iterations but we stop early if we can show with very high confidence that CPV_i is going to be greater than 0.1. As an example, suppose we have gone through j iterations and let CPV_i^j be the value of CPV_i on the current iteration j (CPV_i^j is calculated as the number of times so far that the best scoring rule on the randomized data set has a lower p-value than the best scoring rule over the original unrandomized data set). Using a normality assumption on the distribution of CPV_i, we can estimate the standard deviation σ_{CPV_i} and form a 95% confidence interval on the true value of CPV_i. This is achieved using the interval $CPV_i^j \pm \frac{1.96\sigma_{CPV_i}}{\sqrt{n}}$. If the lower half of this interval, namely $CPV_i^j - \frac{1.96\sigma_{CPV_i}}{\sqrt{n}}$, is greater than, say 0.1, we are 95% sure that this score will be insignificant at the 0.1 level. On a typical data set where an outbreak is unlikely, the majority of days will result in insignificant p-values. As a result, we expect the racing optimization to allow us to stop early on many days.

2.7 Using FDR to Determine Which p-values are Significant

This algorithm can be used on a day-to-day basis or it can operate over a history of several days to report all significantly anomalous patterns. When using our algorithm on a day-to-day basis, the compensated p-value CPV_i obtained for the current day through the randomization tests can be interpreted at face value. However, when analyzing historical data, we need to characterize the false discovery rate over the group of days in the history, which requires comparing the CPV_i values for each day. Comparison of multiple CPV_i values in the historical window results in a second overfitting opportunity analogous to that caused by performing multiple hypothesis tests to determine the best rule for a particular day. As an illustration, suppose we took 500 days of randomly generated data. Then, approximately 5 days would have a CPV_i value less than 0.01 and these days would naively be interpreted as being significant. Two approaches can be used to correct this problem. The Bonferroni method (Bonferroni, 1936) aims to reduce the probability of making one or more false positives to be no greater than α. However, this tight control over the number of false positives causes many real discoveries to be missed. The other alternative is Benjamini and Hochberg's False Discovery Rate method, (Benjamini and Hochberg, 1995), which we will refer to as BH-FDR. BH-FDR guarantees that the false discovery rate, which is the expected fraction of the number of false positives over the number of tests in which the null hypothesis is rejected, will be no greater than α_{FDR}. The FDR method is more desirable as it has a higher power than the Bonferroni correction while keeping a reasonable control over the false discovery rate. We incorporate the BH-FDR method into our rule-learning algorithm by first providing an α_{FDR} value and then using BH-FDR to find the cutoff threshold for determining which p-values are significant.

3. WSARE 3.0

Many detection algorithms (Goldenberg et al., 2002; Zhang et al., 2003; Fawcett and Provost, 1997) assume that the observed data consist of cases from background activity, which we will refer to as the baseline, plus any cases from irregular behavior. Under this assumption, detection algorithms operate by subtracting away the baseline from recent data and raising an alarm if the deviations from the baseline are significant. The challenge facing all such systems is to estimate the baseline distribution using data from historical data. In general, determining this distribution is extremely difficult due to the different trends present in surveillance data. Seasonal variations in weather and temperature can dramatically alter the distribution of surveillance data. For example, flu season typically occurs during mid-winter, resulting in an increase in ED cases involving respiratory problems. Disease outbreak detectors intended to detect epidemics such as SARS, West Nile Virus and anthrax are not interested in detecting the onset of flu season and would be thrown off by it. Day of week variations make up another periodic trend. Figure 3, which is taken from Goldenberg et al. (2002), clearly shows the periodic elements in cough syrup and liquid decongestant sales.

Choosing the wrong baseline distribution can have dire consequences for an early detection system. Consider once again a database of ED records. Suppose we are presently in the middle of flu season and our goal is to detect anthrax, not an influenza outbreak. Anthrax initially causes symptoms similar to those of influenza. If we choose the baseline distribution to be outside of the current flu season, then a comparison with recent data will trigger many false anthrax alerts due to the flu cases. Conversely, suppose we are not in the middle of flu season and that we obtain the baseline distribution from the previous year's influenza outbreak. The system would now consider high counts of flu-like symptoms to be normal. If an anthrax attack occurs, it would be detected late, if at all.

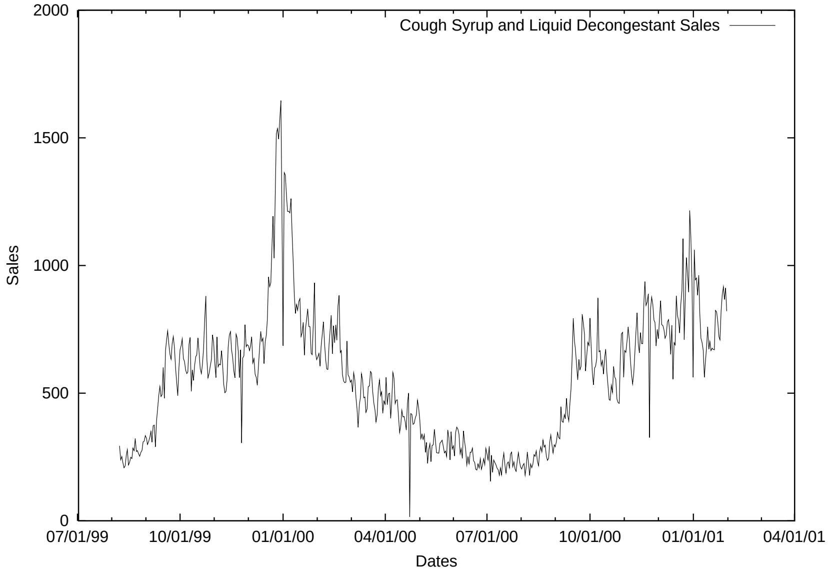


Figure 3: Cough syrup and liquid decongestant sales from (Goldenberg et al., 2003)

There are clearly tradeoffs when defining this baseline distribution. At one extreme, we would like to capture any current trends in the data. One solution would be to use only the most recent data, such as data from the previous day. This approach, however, places too much weight on outliers that may only occur in a short but recent time period. On the other hand, we would like the baseline to be accurate and robust against outliers. We could use data from all previous years to establish the baseline. This choice would smooth out trends in the data and likely raise alarms for events that are due to periodic trends.

In WSARE 2.0, we made the baseline distribution to be raw data obtained from selected historical days. For example, we chose data from 35, 42, 49, and 56 days prior to the current day under examination. These dates were chosen to incorporate enough data so that seasonal trends could be captured and they were also chosen to avoid weekend versus weekday effects by making all comparisons from the same day of week. This baseline was chosen manually in order to tune the performance of WSARE 2.0 on the data set. Ideally, the detection system should determine the baseline automatically.

In this section, we describe how we use a Bayesian network to represent the joint probability distribution of the baseline. From this joint distribution, we represent the baseline distributions from the conditional distributions formed by conditioning on what we term *environmental attributes*. These attributes are precisely those attributes that account for trends in the data, such as the season, the current flu level and the day of week.

3.1 Creating the Baseline Distribution

Learning the baseline distribution involves taking all records prior to the past 24 hours and building a Bayesian network from this subset. During the structure learning, we differentiate between environmental attributes, which are attributes that cause trends in the data, and *response attributes*, which are the remaining attributes. The environmental attributes are specified by the user based on the user's knowledge of the problem domain. If there are any latent environmental attributes that are not accounted for in this model, the detection algorithm may have some difficulties. However, as will be described later on in Section 4, WSARE 3.0 was able to overcome some hidden environmental attributes in our simulator.

While learning the structure of the Bayesian network, environmental attributes are prevented from having parents because we are not interested in predicting their distributions, but rather, we want to use them to predict the distributions of the response attributes. In general, any structure learning algorithm can be used in this step as long as it follows this restriction. In fact, the structure search can even exploit this constraint by avoiding search paths that assign parents to the environmental attributes.

We experimented with using hillclimbing to learn the Bayesian network structure and found it to be both slow and prone to being trapped in local optima. As a result, we developed an efficient structure search algorithm called Optimal Reinsertion based on ADTrees (Moore and Lee, 1998). Unlike hillclimbing, which performs a single modification to a directed acyclic graph (DAG) on each step, Optimal Reinsertion is a larger scale search operator that is much less prone to local optima. Optimal Reinsertion first picks a target node T from the DAG, disconnects T from the graph, and efficiently finds the optimal way to reinsert T back into the graph according to the scoring function. The details of this algorithm can be found in (Moore and Wong, 2003).

We have often referred to environmental attributes as attributes that cause periodic trends. Environmental attributes, however, can also include any source of information that accounts for recent changes in the data. For example, suppose we detect that a botulism outbreak has occurred and we would still like to be on alert for any anthrax releases. We can add “Botulism Outbreak” as an environmental attribute to the network and supplement the current data with information about the botulism outbreak. Incorporating such knowledge into the Bayesian network allows WSARE to treat events due to the botulism outbreak as part of the baseline.

Once the Bayesian network is learned, we have a joint probability distribution for the data. We would like to produce a conditional probability distribution, which is formed by conditioning on the values of the environmental attributes. Suppose that today is February 21, 2003. If the environmental attributes were *Season* and $Day\ of\ Week$, we would set $Season = Winter$ and $Day\ of\ Week = Weekday$. Let the response attributes in this example be $X_1, ..., X_n$. We can then obtain the probability distribution $P(X_1, ..., X_n \mid Season = Winter, \ Day\ of\ Week = Weekday)$ from the Bayesian network. For simplicity, we represent the conditional distribution as a data set formed by sampling a large number of records from the Bayesian network conditioned on the environmental attributes. The number of samples is specified by the parameter *num_baseline_samples*, which has to be large enough to ensure that samples with rare combinations of attributes will be present. In general, this number will depend on the learned Bayesian network's structure and the parameters of the network. We chose to sample 10000 records because we determined empirically that this number is a reasonable compromise between running time and accuracy on our data. We will refer to this sampled data set as $DB_{baseline}$. The data set corresponding to the records from the past 24 hours of the current day will be named DB_{recent}.

We used a sampled data set instead of using inference mainly for simplicity. Inference might be faster than sampling to obtain the conditional probability $P(X_1, \ldots, X_n \mid \text{Environmental Attributes})$, especially when the learned Bayesian networks are simple. However, if inference is used, it is somewhat unclear how to perform the randomization test. With sampling, on the other hand, we only need to generate $DB_{baseline}$ once and then we can use it for the randomization test to obtain the p-values for all the rules. In addition, sampling is easily done in an efficient manner since environmental attributes have no parents. While a sampled data set provides the simplest way of obtaining the conditional distribution, we have not completely ignored the possibility of using inference to speed up this process. We would like to investigate this direction further in our future work.

3.2 Dealing with New Hospitals Coming Online

WSARE 3.0 assumes that the baseline distribution remains relatively stable, with the environmental attributes accounting for the only sources of variation. However, in a real life situation where data are pooled from various EDs around a city, new hospitals frequently come online and become a new source of data to be monitored. These new data sources cause a shift from the baseline distribution that is not accounted for in WSARE 3.0. For example, suppose a children's hospital begins sending data to the surveillance system. In this case, WSARE 3.0 would initially detect an anomalous pattern due to an increase in the number of cases involving children from the part of the city where the children's hospital is located. Over time, WSARE 3.0 would eventually incorporate the newly added hospital's data into its baseline.

In general, this problem of a shifted distribution is difficult to address. We approach this issue by ignoring the new data sources until we have enough data from them to incorporate them into the baseline. Our solution relies on the data containing an attribute such as $Hospital\ ID$ that can identify the hospital that the case originated from. HIPAA regulations can sometimes prevent ED data from containing such identifying attributes. In this case, we recommend using WSARE 2.0 with a recent enough baseline period in order to avoid instabilities due to new data sources. Whenever the data includes a $Hospital\ ID$ attribute, we first build a list of hospitals that provide data for the current day. For each hospital in this list, we keep track of the first date a case came from that particular hospital. If the current day is less than a year after the first case date, we consider that hospital to have insufficient historical data for the baseline and we ignore all records from that hospital. For each hospital with sufficient historical records, we then build a Bayesian network using only historical data originating from that particular hospital.

In order to produce the baseline data set, we sample a total of 10000 records from all the hospital Bayesian networks. Let hospital h have n_h records on the current day and suppose there are H hospitals with sufficient historical data for the current date. Then let $N_h = \sum_{h=1}^{H} n_h$. Each hospital Bayesian network contributes $10000 * \frac{n_h}{N_h}$ number of samples to the baseline data set. As an example, suppose we have 5 hospitals with 100 records each. Furthermore, assume that we can ignore the fourth hospital's records since its first case is less than a year prior to the current date. We are then left with 4 hospitals with 100 records each. After we build the Bayesian network for each hospital, we sample 2500 records from the Bayesian network belonging to each of the four hospitals.

4. Evaluation

Validation of early outbreak detection algorithms is generally a difficult task due to the type of data required. Health-care data during a known disease outbreak, either natural or induced by a bioagent release, are extremely limited. Even if such data were plentiful, evaluation of biosurveillance algorithms would require the outbreak periods in the data to be clearly labelled. This task requires an expert to inspect the data manually, making this process extremely slow. Consequently, such labelled data would still be scarce and making statistically significant conclusions with the results of detection algorithms would be difficult. Furthermore, even if a group of epidemiologists were to be assembled to label the data, there would still be disagreements as to when an outbreak begins and ends.

As a result of these limitations, we validate the WSARE algorithms on data from a simulator which we will refer to as the city Bayesian network (CityBN) simulator. The CityBN simulator is based on a large Bayesian network that introduces temporal fluctuations based on a variety of factors. The structure and the parameters for this Bayesian network are created by hand. This simulator is not intended to be a realistic epidemiological model. Instead, the model is designed to produce extremely noisy data sets that are a challenge for any detection algorithm. In addition to simulated data, we also include WSARE output from ED data from an actual city. Due to the fact that epidemiologists have not analyzed this real world data set for known outbreaks, we are only able to provide annotated results from the runs of WSARE.

4.1 The CityBN Simulator

The city in the CityBN simulator consists of nine regions, each of which contains a different sized population, ranging from 100 people in the smallest area to 600 people in the largest section, as shown in Table 5. We run the simulation for a two year period starting from January 1, 2002 to December 31, 2003. The environment of the city is not static, with weather, flu levels and food conditions in the city changing from day to day. Flu levels are typically low in the spring and summer but start to climb during the fall. We make flu season strike in winter, resulting in the highest flu levels during the year. Weather, which only takes on the values of hot or cold, is as expected for the four seasons, with the additional feature that it has a good chance of remaining the same as it was yesterday. Each region has a food condition of good or bad. A bad food condition facilitates the outbreak of food poisoning in the area.

NW (100)	N (400)	NE (500)
W (100)	C (200)	E (300)
SW (200)	S (200)	SE (600)

Table 5: The geographic regions in the CityBN simulator with their populations in parentheses

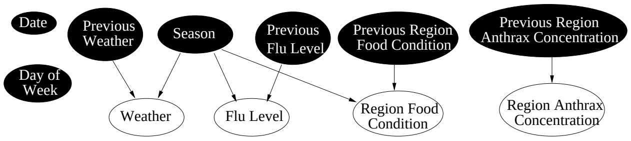

Date
Previous Weather
Season
Previous Flu Level
Previous Region Food Condition
Previous Region Anthrax Concentration
Day of Week
Weather
Flu Level
Region Food Condition
Region Anthrax Concentration

Figure 4: City Status Bayesian Network

We implement this city simulation using a single large Bayesian network. For simplicity, we will describe this large Bayesian network in two parts, as shown in Figures 4 and 5. The subnetwork shown in Figure 4 is used to create the state of the city for a given day. Given the state of the city, the network in Figure 5 is used to generate records for individual patients.

We use the convention that any nodes shaded black in the subnetwork are set by the system and do not have their values generated probabilistically. Due to space limitations, instead of showing eighteen separate nodes for the current and previous food conditions of each region in Figure 4, we summarize them using the generic nodes *Region Food Condition* and *Previous Region Food Condition* respectively. This same space saving technique is used for the current and previous region anthrax concentrations. Most of the nodes in this subnetwork take on two to three values. For each day, after the black nodes have their values set, the values for the white nodes are sampled from the subnetwork. These records are stored in the City Status (CS) data set. The simulated anthrax release is selected for a random date during a specified time period. One of the nine regions is chosen randomly for the location of the simulated release. On the date of the release, the *Region Anthrax Concentration* node is set to have the value of *High*. The anthrax concentration remains high for the affected region for each subsequent day with an 80% probability. This probability is chosen in order to ensure that enough individuals in the simulation are being infected by anthrax over an extended period of time after the attack.

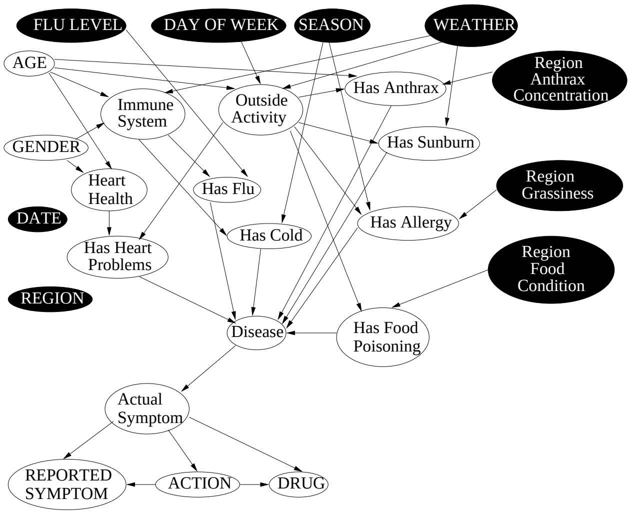


Figure 5: Patient Status Bayesian Network

Table 6: Examples of two records in the PS data set

Location	NW	N
Age	Child	Senior
Gender	Female	Male
Flu Level	High	None
Day of Week	Weekday	Weekday
Weather	Cold	Hot
Season	Winter	Summer
Action	Absent	ED visit
Reported Symptom	Nausea	Rash
Drug	None	None
Date	Jan-01-2002	Jun-21-2002

The second subnetwork used in our simulation produces individual health care cases. Figure 5 depicts the Patient Status (PS) network. On each day, for each person in each region, we sample the individual's values from this subnetwork. The black nodes first have their values assigned from the CS data set record for the current day. For the very first day, the black nodes are assigned a set of initial values. The white nodes are then sampled from the PS network. Each individual's health profile for the day is thus generated. The nodes *Flu Level*, *Day of Week*, *Season*, *Weather*, *Region Grassiness*, and *Region Food Condition* are intended to represent environmental variables that affect the upswings and downswings of a disease. The *Region Grassiness* nodes indicate the amount of pollen in the air and thus affect the allergies of a patient. We choose these environmental variables because they are the most common factors influencing the health of a population. Two of the environmental variables, namely *Region Grassiness* and *Region Food Condition*, are hidden from the detection algorithm while the remaining environmental attributes are observed. We choose to hide these two attributes because the remaining four attributes that are observed are typically considered when trying to account for temporal trends in biosurveillance data.

As for the other nodes, the *Disease* node indicates the status of each person in the simulation. We assume that a person is either healthy or they can have, in order of precedence, allergies, the cold, sunburn, the flu, food poisoning, heart problems or anthrax. If the values of the parents of the *Disease* node indicate that the individual has more than one disease, the *Disease* node picks the disease with the highest precedence. This simplification prevents individuals from having multiple diseases. A sick individual then exhibits one of the following symptoms: none, respiratory problems, nausea, or a rash. Note that in our simulation, as in real life, different diseases can exhibit the same symptoms, such as a person with the flu can exhibit respiratory problems as could a person with anthrax. The actual symptom associated with a person may not necessarily be the same as the symptom that is reported to health officials. Actions available to a sick person included doing nothing, buying medication, going to the ED, or being absent from work or school. As with the CS network, the arities for each node in the PS network are small, ranging from two to four values. If the patient performs any action other than doing nothing, the patient's health care case is added to the PS data set. Only the attributes in Figure 5 labelled with uppercase letters are recorded, resulting in a great deal of information being hidden from the detection algorithm, including some latent environmental attributes. The number of cases the PS network generates daily is typically in the range of 30 to 50 records. Table 6 contains two examples of records in the PS data set.

We run six detection algorithms on 100 different PS data sets. Each data set is generated for a two year period, beginning on January 1, 2002 and ending December 31, 2003. The detection algorithms train on data from the first year until the day being monitored while the second year is used for evaluation. The anthrax release is randomly chosen in the period between January 1, 2003 and December 31, 2003.

We try to simulate anthrax attacks that are not trivially detectable. Figure 6 plots the total count of health-care cases on each day during the evaluation period while Figure 7 plots the total count of health-care cases involving respiratory symptoms for the same simulated data set. A naive detection algorithm would assume that the highest peak in this graph would be the date of the anthrax release. However, the anthrax release occurs on day index 74,409, which is clearly not the highest peak in either graph. Occasionally the anthrax releases affects such a limited number of people that it is undetected by all the algorithms. Consequently, we only use data sets with more than eight reported anthrax cases on any day during the attack period.

The following paragraphs describe the six detection algorithms that we run on the data sets. Three of these methods, namely the control chart, moving average, and ANOVA regression algorithms, operate on univariate data. We apply these three algorithms to two different univariate data

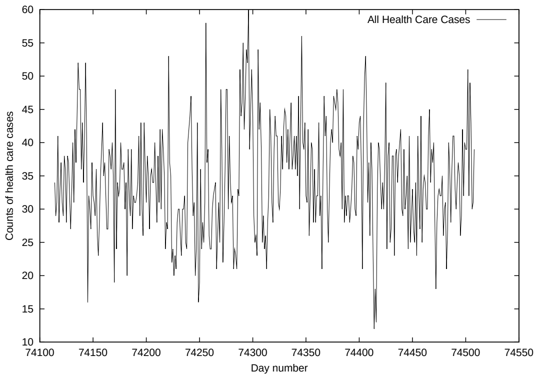


Figure 6: Daily counts of health-care data

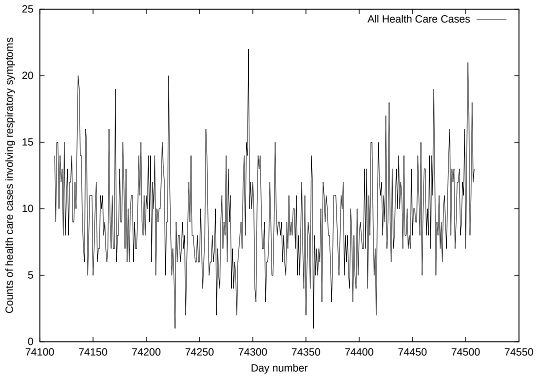


Figure 7: Daily counts of health-care data involving respiratory symptoms

sets – one data set is composed of total daily counts and the other of daily counts of cases involving respiratory symptoms. The remaining three algorithms are variations on WSARE.

The Control Chart Algorithm The first algorithm used is a common anomaly detection algorithm called a control chart. This detector determines the mean and variance of the total number of records on each day in the PS data set during the training period. A threshold is calculated based on the formula below, in which Φ^{-1} is the inverse to the cumulative distribution function of a standard normal while the p-value is supplied by the user.

$$\text{threshold} = \mu + \sigma * \Phi^{-1}(1 - \frac{\text{p-value}}{2})$$

If the aggregate daily counts of health care data exceeds this threshold during the evaluation period, the control chart raises an alarm. We use a training period of January 1, 2002 to December 31, 2002.

Moving Average Algorithm The second algorithm that we use is a moving average algorithm that predicts the count for the current day as the average of counts from the previous 7 days. The window of 7 days is intended to capture any recent trends that might appear in the data. An alarm level is generated by fitting a Gaussian to data prior to the current day and obtaining a p-value for the current day's count. The mean and standard deviation for the Gaussian is calculated using data from 7 days before the current day.

ANOVA Regression A simple detector that accounts for environmental factors is ANOVA regression, which is simply linear regression supplemented with covariates for the environmental variables. We include 6 covariates for the days of the week, 3 for the seasons and one for the daily aggregate count from the previous day. ANOVA regression is a fairly powerful detector when temporal trends are present in the data, as was shown in (Buckeridge et al., 2005).

WSARE 2.0 WSARE 2.0 is also evaluated, using a baseline distribution of records from 35, 42, 49 and 56 days before the current day. The attributes used by WSARE 3.0 as environmental attributes are ignored by WSARE 2.0. If these attributes are not ignored, WSARE 2.0 would report many trivial anomalies. For instance, suppose that the current day is the first day of fall, making the environmental attribute $Season = Fall$. Furthermore, suppose that the baseline is taken from the summer season. If the environmental attributes are not ignored, WSARE 2.0 would notice that 100% of the records for the current day have $Season = Fall$ while 0% of the records in the baseline data set match this rule.

WSARE 2.5 Instead of building a Bayesian network over the past data, WSARE 2.5 simply builds a baseline from all records prior to the current period with their environmental attributes equal to the current day's. In our simulator, we use the environmental attributes $Flu\ Level$, $Season$, $Day\ of\ Week$ and $Weather$. To clarify this algorithm, suppose for the current day we have the following values of these environmental attributes: $Flu\ Level = High$, $Season = Winter$, $Day\ of\ Week = Weekday$ and $Weather = Cold$. Then $DB_{baseline}$ would contain only records before the current period with environmental attributes having exactly these values. It is possible that no such records exist in the past with exactly this combination of environmental attributes. If there are fewer than five records in the past that matched, WSARE 2.5 can not make an informed decision when comparing the current day to the baseline and simply reports nothing for the current day.

WSARE 3.0 WSARE 3.0 uses the same environmental attributes as WSARE 2.5 but builds a Bayesian network for all data from January 1, 2002 to the day being monitored. We hypothesize that WSARE 3.0 would detect the simulated anthrax outbreak sooner than WSARE 2.5 because 3.0 can handle the cases where there are no records corresponding to the current day's combination of environmental attributes. The Bayesian network is able to generalize from days that do not match today precisely, producing an estimate of the desired conditional distribution. For efficiency reasons, we allow WSARE 3.0 to learn the network structure from scratch once every 30 days on all data since January 1, 2002. On intermediate days, WSARE 3.0 simply updates the parameters of the previously learned network without altering its structure. In practice, we expect WSARE 3.0 to be used in this way since learning the network structure on every day may be very expensive computationally.

4.1.1 RESULTS

In order to evaluate the performance of the algorithms, we plot an Activity Monitoring Operating Characteristic (AMOC) curve (Fawcett and Provost, 1999), which is similar to an ROC curve. On the AMOC curves to follow, the x-axis indicates the number of false positives per month while the y-axis measures the detection time in days. For a given alarm threshold, we plot the performance of the algorithm at a particular false positive level and detection time on the graph. As an example, suppose we are dealing with an alarm threshold of 0.05. We then take all the alarms generated by an algorithm, say WSARE 3.0, that have a p-value less than or equal to 0.05. Suppose there are two such alarms, with one alarm appearing 5 days before the simulated anthrax release, which would be considered a false positive, and the other appearing 3 days after the release, making the detection time 3 days. If we run the detection algorithms for 1 month, then we would plot a point at $(1,3)$.

We then vary the alarm threshold in the range of 0 to 0.2 and plot points at each threshold value. For a very sensitive alarm threshold such as 0.2, we expect a higher number of false positives but a lower detection time. Hence the points corresponding to a sensitive threshold would be on the lower right hand side of the graph. Conversely, an insensitive alarm threshold like 0.01 would result in a lower number of false positives and a higher detection time. The corresponding points would appear on the upper left corner of the graph.

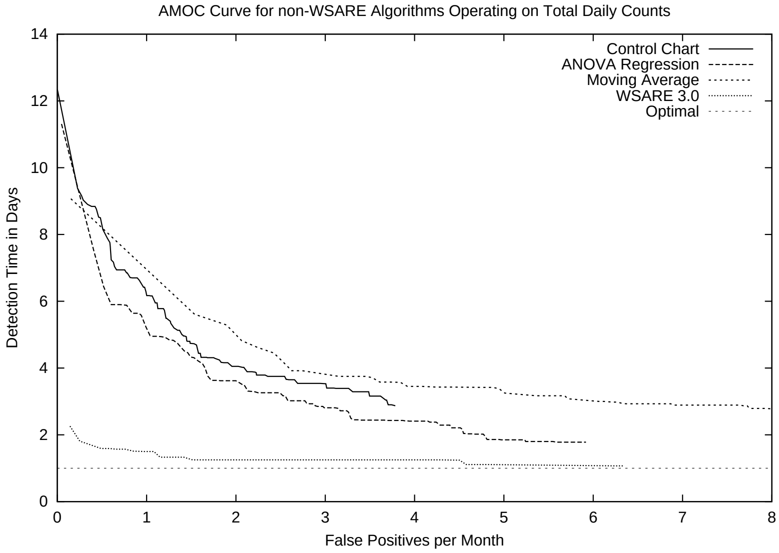


Figure 8: AMOC curves comparing WSARE 3.0 to univariate algorithms operating on total daily counts from the CityBN simulator

Figures 8 to 10 plot the AMOC curve, averaged over the 100 data sets, with an alarm threshold increment of 0.001. On these curves, the optimal detection time is one day, as shown by the dotted line at the bottom of the graph. We add a one day delay to all detection times to simulate reality where current data is only available after a 24 hour delay. Any alert occurring before the start of the simulated anthrax attack is treated as a false positive. Detection time is calculated as the first alert raised after the release date. If no alerts are raised after the release, the detection time is set to 14 days.

Figures 8 and 9 show that WSARE 3.0 clearly outperform the univariate algorithms when the univariate algorithms operate on the total daily counts and also when the univariate algorithms operate on the daily counts of cases involving respiratory symptoms. In Figure 10, WSARE 2.5 and WSARE 3.0 outperform the other algorithms in terms of the detection time and false positive tradeoff. For a false positive rate of one per month, WSARE 2.5 and WSARE 3.0 are able to detect the anthrax release within a period of one to two days. The Control Chart, moving average, ANOVA regression and WSARE 2.0 algorithms are thrown off by the periodic trends present in the PS data.

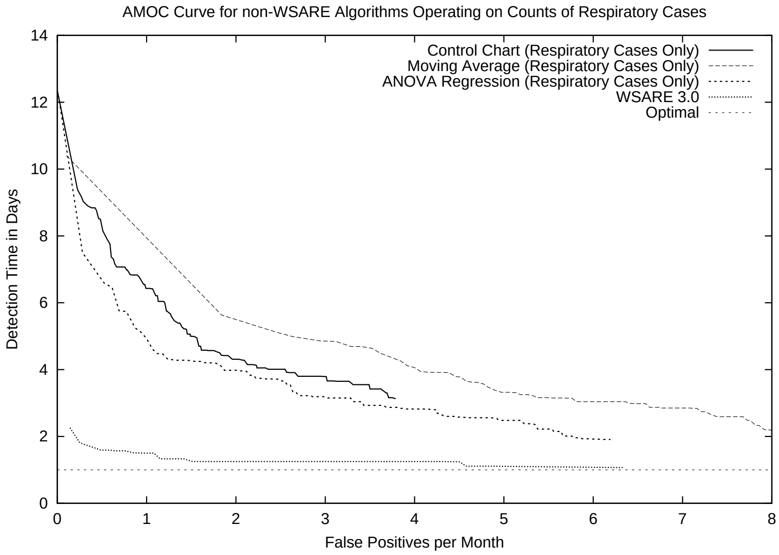


Figure 9: AMOC curves comparing WSARE 3.0 to univariate algorithms operating on cases involving respiratory symptoms from the CityBN simulator

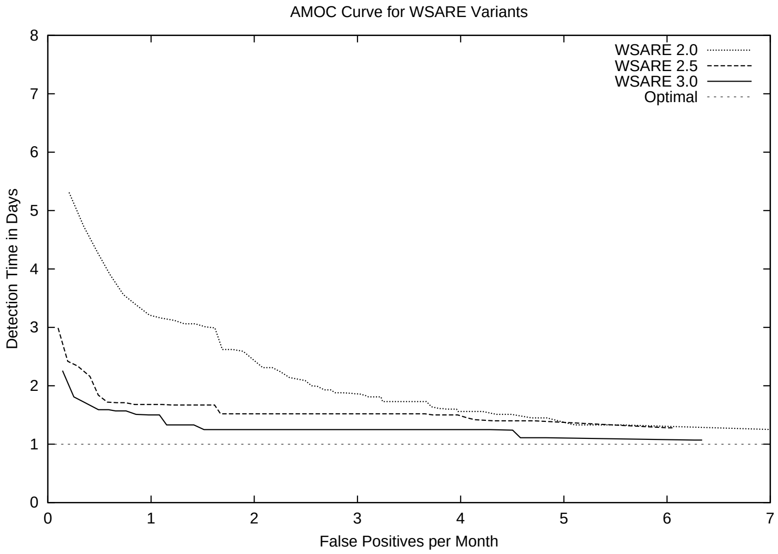


Figure 10: AMOC curves for WSARE variants operating on CityBN data

We previously proposed that WSARE 3.0 would have a better detection time than WSARE 2.5 due to the Bayesian network's ability to produce a conditional distribution for a combination of environmental attributes that may not exist in the past data. After checking the simulation results for which WSARE 3.0 outperformed WSARE 2.5, we conclude that in some cases, our proposition is true. In others, the p-values estimated by WSARE 2.5 are not as low as those of version 3.0. The baseline distribution of WSARE 2.5 is likely not as accurate as the baseline of WSARE 3.0 due to smoothing performed by the Bayesian network. The false positives found by WSARE 2.5 and WSARE 3.0 are likely due to other non-anthrax illnesses that are not accounted for in the Bayesian network. Had we explicitly added a Region Food Condition environmental attribute to the Bayesian network, this additional information would likely have reduced the false positive count.

Figures 11 to 14 illustrate the various outbreak sizes in the simulated data by plotting the number of anthrax cases per day during the outbreak period. Since the outbreak sizes and durations are randomly generated for each of the 100 data sets, we do not have room to show plots for each data set. Instead, we include representative plots of the outbreaks that appeared in our simulated data. Figure 11 represents a large scale outbreak which was easily detected on the first day by most algorithms. Large scale outbreaks were rare in our simulated data. Figure 12 is a representative plot of a medium scale outbreak that is most common in the data. The particular outbreak shown in Figure 12 is also detected by WSARE 3.0 on the first day for an alarm threshold of 0.005. Small scale outbreaks, as shown in Figure 13, are the most difficult to detect. WSARE 3.0 detects the outbreak in Figure 13 on the third day with a very insensitive alarm threshold of 0.005. Figure 14 contains an outbreak that WSARE 3.0 is unable to detect using an alarm threshold of 0.03.

We also conduct four other experiments to determine the effect of varying certain parameters of WSARE 3.0. In the first experiment, we use a Bonferroni correction to correct for multiple hypothesis testing instead of a randomization test. The AMOC curve for the results, as shown in Figure 15 indicate that the Bonferroni correction results are almost identical to those of the randomization test. This similarity was expected because on each day, there are approximately only 50 hypothesis tests being performed to find the best scoring rule and the hypothesis tests are weakly dependent on each other. However, as the number of hypothesis tests increases and as the dependence between the hypothesis tests increases, the results of the randomization test should be better than those of the Bonferroni correction.

In order to illustrate the advantages of the randomization test, we produce dependent hypothesis tests in WSARE by creating attributes that are dependent on each other. We generate a data set using a Markov chain $X_0, \ldots, X_n$ in which the states of each random variable in the chain become the attributes in the data set. Each random variable X_t in the Markov chain can be in state A, B, C, or D, except for X_0 which always starts at A. At each time step t, the random variable X_t retains the state of X_{t-1} in the Markov chain with a 90% chance. With a 10% chance, X_t takes on the next state in the ordered sequence A, B, C and D. As an example, if $X_{t-1} = A$, X_t can remain as A or it can become B. If $X_{t-1} = D$, X_t can retain the same state as X_{t-1} or transition back to the state A, which is the first state of the ordered sequence. We use this model to generate 150 days worth of data in which each day contains 1000 records and each record contains 100 attributes. We then sample 14 days of data with the same characteristics except the Markov chain is altered slightly so that each random variable X_t remains in the same state as X_{t-1} with an 89% probability. Thirty data sets, each containing a total of 164 days are produced. Two variations of WSARE 2.0, one with a randomization test and the other with a Bonferroni correction, are applied to these thirty data sets in order to detect the change.

Figure 16 plots the average AMOC curve of this experiment. As the graph illustrates, at a false positive rate of less than 0.4 per month, the randomization test has a much better detection time. Upon further analysis, we find that the reduced performance of the Bonferroni correction are due to a much higher number of false positives. As an example, we find that WSARE often notices that a rule such as $X_{27} = C$ AND $X_{96} = B$ produces a very good score. The Bonferroni correction deals

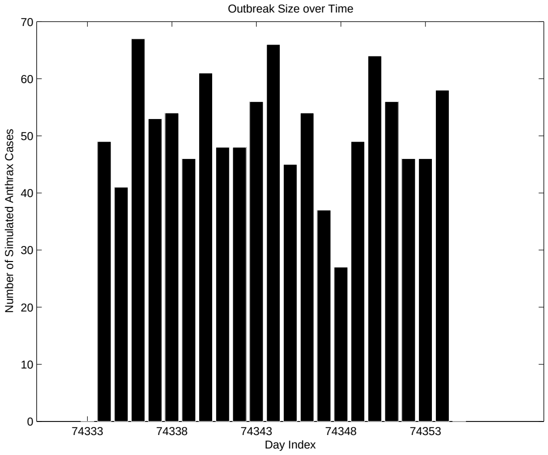


Figure 11: An example of a large scale outbreak in the CityBN data

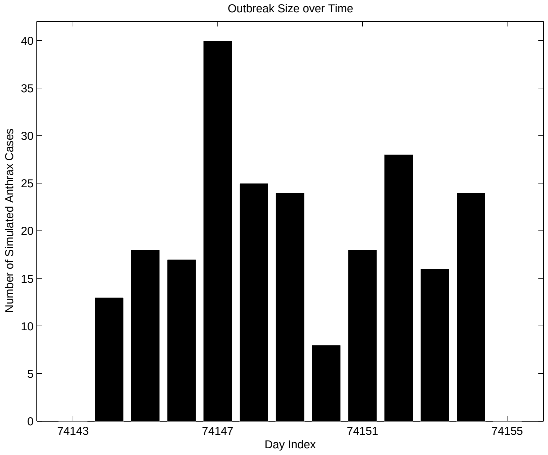


Figure 12: An example of a medium scale outbreak in the CityBN data

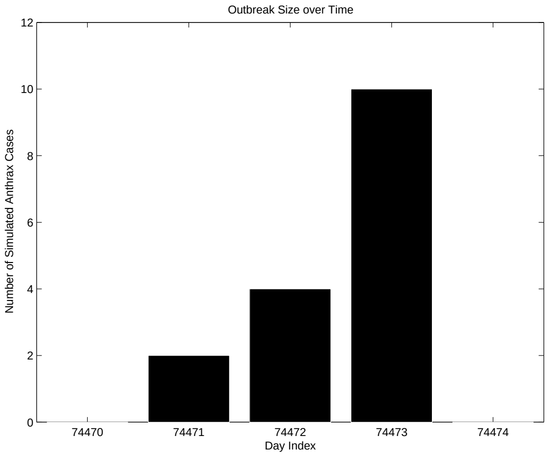


Figure 13: An example of a small scale outbreak in the CityBN data

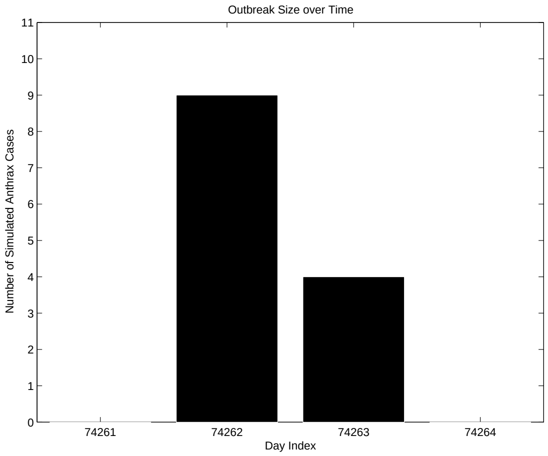


Figure 14: An example of an outbreak that was not detected in the CityBN data by WSARE 3.0 with an alarm threshold of 0.03

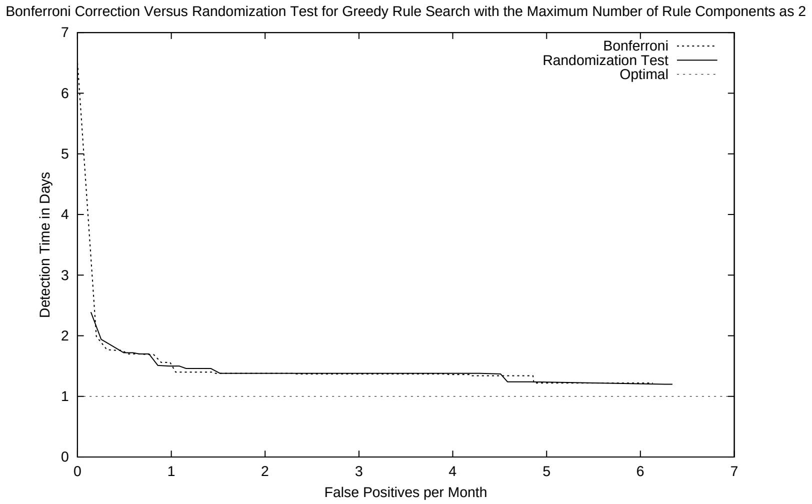


Figure 15: The Bonferroni correction version of WSARE versus the randomization test version on the CityBN data

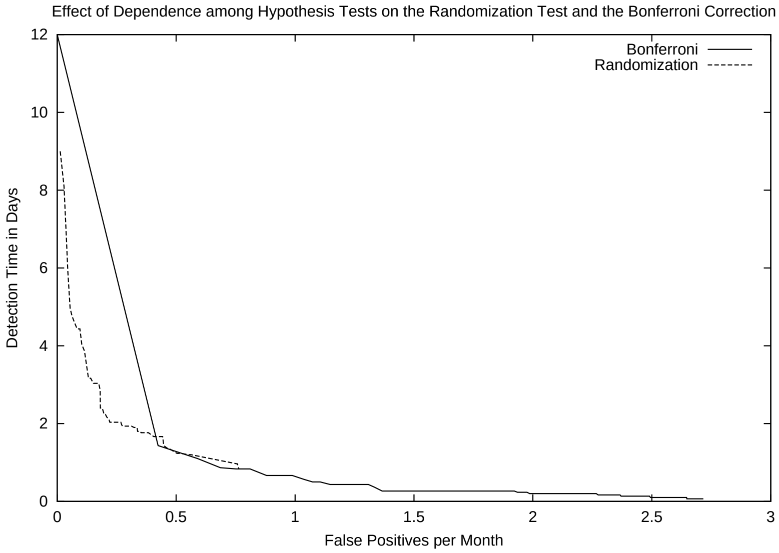


Figure 16: A comparison between the Bonferroni correction version of WSARE and the randomization test version on data generated from a Markov chain

with the multiple hypothesis problem by simply multiplying the score with the number of hypothesis tests. Although there are a high number of hypothesis tests in this experiment, multiplying by the number of hypothesis tests still results in a low compensated p-value. The randomization test, on the other hand, notices that although the score is very good, the probability of finding an equal or better score for another rule, such as $X_{46} = A$ AND $X_{94} = B$ is quite high because of the dependence between attributes. Thus, the resulting compensated p-value from the randomization test is quite high, signifying that the pattern defined by the rule is not so unusual after all.

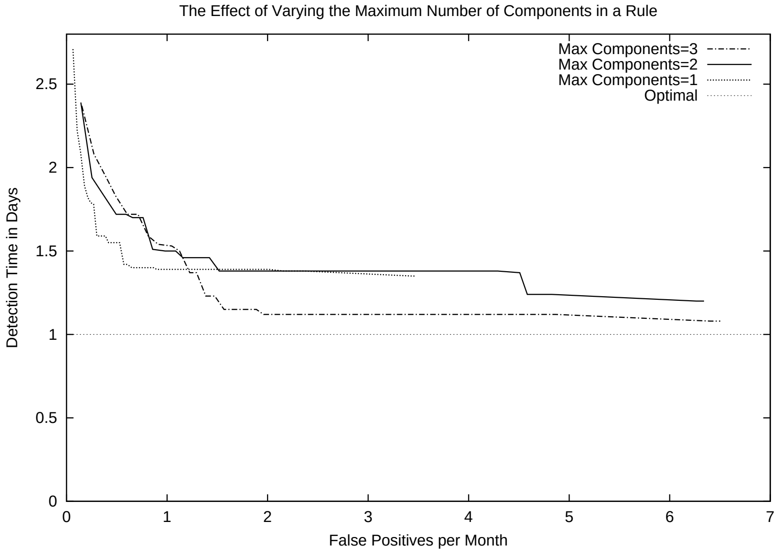


Figure 17: The effect of varying the maximum number of components for a rule on the AMOC curve for CityBN data

The second experiment involves varying the maximum components allowed per rule from one to three. As seen on the AMOC curve in Figure 17, the variations do not seem significantly different to the left of the one false positive per month mark. However, after this point, a version of WSARE with a three component limit outperforms the other two variations. By setting the maximum number of components per rule to be three, WSARE is capable of being more expressive in its description of anomalous patterns. On the other hand, WSARE also guards against overfitting by requiring each component added to be 95% significant for the two hypothesis tests performed in Section 2.5. This criterion makes the addition of a large number of rule components unlikely and we expect the optimal number of components to be about two or three.

The third experiment involves changing the rule search to be exhaustive rather than greedy. Note that if we compare the score of the best rule found by the exhaustive method against that found by the greedy method, the exhaustive method would unquestionably find a rule with an equal or greater score than the greedy method. In Figure 18, however, we compare the performance of the two algorithms using AMOC curves. Each coordinate on the AMOC curve is a result of a compensated p-value produced by the randomization test and not the rule score. Thus, even though an exhaustive rule search will always equal or outperform a greedy rule search in terms of the best rule score, it is not guaranteed to be superior to the greedy rule search on an AMOC curve due to the fact that the randomization test adjusts the rule score for multiple hypothesis testing. In Figure 18, we plot the AMOC curves comparing the average performance for both the exhaustive and greedy algorithms over 100 experiments; we do not show the confidence intervals in order to avoid clutter. The confidence intervals for both the greedy and the exhaustive curves do overlap substantially. Therefore, there appears to be no significant difference between the two algorithms for the data

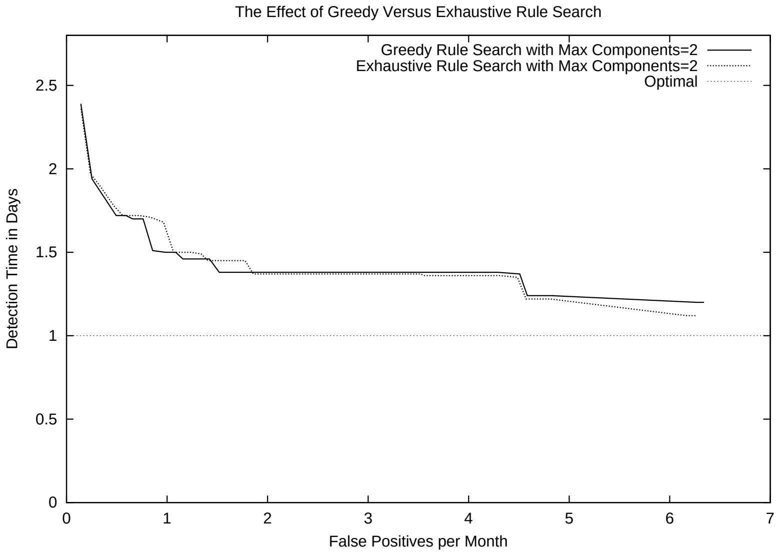


Figure 18: AMOC curves for greedy versus exhaustive rule search for CityBN data

from this simulator. We measure the exhaustive search to be 30 times slower than the greedy search. Since the AMOC curves are nearly identical for our simulated data, we prefer the greedy search.

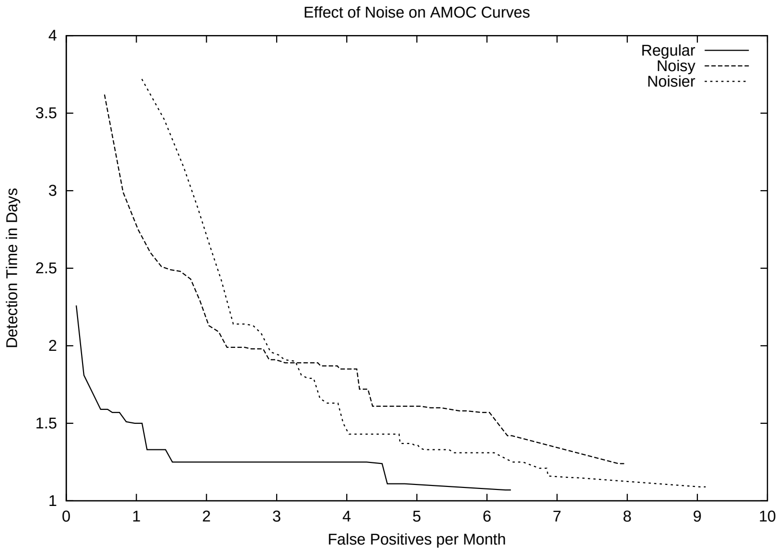


Figure 19: The effect of increased noise levels in the data on WSARE 3.0

Finally, we experiment with adding noise to the data by increasing the number of ED cases due to allergies, food poisoning, sunburns and colds. We increase the noise levels by increasing the probabilities of *Region Food Condition* $=$ *bad*, *Has Allergy* $=$ *true*, *Has Cold* $=$ *true*, and

$Has\ Sunburn = true$ in their respective conditional probability tables. Note that these nodes are all not visible in the output data. Increasing these probabilities involves changes to many entries of the conditional probability tables and we do not have space to list all of the changes. In general, we increase the probabilities of the corresponding entries in the conditional probability tables by approximately 0.004-0.005. We cannot say specifically how many noisy cases are generated since this amount fluctuates over time.

We produce 100 data sets with increased noise levels which we will refer to as "Noisy" and we also produce another 100 data sets with even more noise which we will refer to as "Noisier". The "Regular" data sets are the 100 data sets used in all previous experiments. We then apply WSARE 3.0 to these three groups. The average AMOC curve for each group of 100 data sets is plotted in Figure 19. As in previous experiments, we use the environmental attributes of $Flu\ Level$, $Season$, $Day\ of\ Week$ and $Weather$. As shown in Figure 19, both the detection time and the false positive rate degrade with increased noise levels.

4.2 Annotated Output of WSARE 3.0 on Actual ED Data for 2001

We also test the performance of WSARE 3.0 on actual ED data from a major US city. This database contains almost seven years worth of data, with personal identifying information excluded in order to protect patient confidentiality. The attributes in this database include date of admission, coded hospital ID, age decile, gender, syndrome information, discretized home latitude, discretized home longitude, discretized work latitude, discretized work longitude and both home location and work location on a coarse latitude-longitude grid. In this data, new hospitals come online and begin submitting data during the time period that the data is collected. We use the solution described in Section 3.2 to address this problem. WSARE operates on data from the year 2001 and is allowed to use over five full years worth of training data from the start of 1996 to the current day. The environmental attributes used are month, day of week and the number of cases from the previous day with respiratory problems. The last environmental attribute is intended to be an approximation to the flu levels in the city. We use a one-sided Fisher's exact test to score the rules such that only rules corresponding to an upswing in recent data are considered. In addition, we apply the Benjamini-Hochberg FDR procedure with $\alpha_{FDR} = 0.1$.

The following list contains the significant anomalous patterns found in the real ED data for the year 2001.

```
1. 2001-02-20: SCORE = -2.15432e-07 PVALUE = 0
   15.9774% (85/532) of today's cases have Viral Syndrome = True and Respiratory Syndrome = False
   8.84% (884/10000) of baseline cases have Viral Syndrome = True and Respiratory Syndrome = False
2. 2001-06-02: SCORE = -3.19604e-08 PVALUE = 0
   1.27971% (7/547) of today's cases have Age Decile = 10 and Home Latitude = Missing
   0.02% (2/10000) of baseline cases have Age Decile = 10 and Home Latitude = Missing
3. 2001-06-30: SCORE = -2.39821e-07 PVALUE = 0
   1.44% (9/625) of today's cases have Age Decile = 10
   0.09% (9/10000) of baseline cases have Age Decile = 10
4. 2001-08-08: SCORE = -1.21558e-08 PVALUE = 0
   83.7979% (481/574) of today's cases have Unknown Syndrome = False
   73.6926% (7370/10001) of baseline cases have Unknown Syndrome = False
```

```
5. 2001-10-10: SCORE = -1.42315e-06 PVALUE = 0
   0.994036% (5/503) of today's cases have Age Decile = 10 and Home Latitude = Missing
   0.009998% (1/10002) of baseline cases have Age Decile = 10 and Home Latitude = Missing
6. 2001-12-02: SCORE = -4.31806e-07 PVALUE = 0
   14.7059% (70/476) of today's cases have Viral Syndrome = True and Encephalitic Syndrome = False
   7.73077% (773/9999) of baseline cases have Viral Syndrome = True and Encephalitic Syndrome = False
7. 2001-12-09: SCORE = -3.31973e-10 PVALUE = 0
   8.57788% (38/443) of today's cases have Hospital ID = 1 and Viral Syndrome = True
   2.49% (249/10000) of baseline cases have Hospital ID = 1 and Viral Syndrome = True
```

Rules 2, 3 and 5 are likely due to clerical errors in the data since the rule finds an increase in the number of people between the ages of 100 and 110. Furthermore, the home zip code for these patients appears to be missing in rules 2 and 5. Rule 4 is uninteresting since it indicates that the number of cases without an unknown symptom, which is typically around 73.7%, has experienced a slight increase. For rules 1, 6 and 7 we went back to the original ED data to inspect the text descriptions of the chief complaints for the cases related to these three rules. The symptoms related to Rules 1, 6 and 7 involve dizziness, fever and sore throat. Given that Rules 1, 6 and 7 have dates in winter, along with the symptoms mentioned, we speculate that this anomalous pattern is likely caused by an influenza strain.

We also include results from WSARE 2.0 running on the same data set. Unlike WSARE 3.0, WSARE 2.0 does not have a similar solution to the approach taken in Section 3.2 to deal with new hospitals coming online. However, by using a short enough baseline period, such as the standard baseline of 35, 42, 49, and 56 days prior to the current date, we can capture fairly recent trends and deal with a changing distribution as new hospitals submit data. The results are shown below. Note that we group together identical rules from consecutive days in order to save space.

```
1. 2001-01-31: SCORE = -8.0763e-07 PVALUE = 0
   21.2766% (110/517) of today's cases have Unknown Syndrome = True
   12.5884% (267/2121) of baseline cases have Unknown Syndrome = True
2. 2001-05-01: SCORE = -1.0124e-06 PVALUE = 0.001998
   18.4739% (92/498) of today's cases have Gender = Male and Home Latitude > 40.5
   10.2694% (202/1967) of baseline cases have Gender = Male and Home Latitude > 40.5

Rules 3-6 from 2001-10-28 to 2001-10-31 all have PVALUE = 0 and involve rules with Hospital ID = Missing

7. 2001-11-01: SCORE = -7.78767e-21 PVALUE = 0
   5.87084% (30/511) of today's cases have Hospital ID = Missing and Hemorrhagic Syndrome = True
   0% (0/1827) of baseline cases have Hospital ID = Missing and Hemorrhagic Syndrome = True

Rules 8-14 from 2001-11-02 to 2001-11-08 all have PVALUE = 0 and have the rule Hospital ID = Missing

Rules 15-37 from 2001-11-09 to 2001-12-02 all have PVALUE = 0 and have the rule Hospital ID = 14

Rules 38-59 from 2001-12-03 to 2001-12-24 all have PVALUE = 0 and have the rule Hospital ID = 50
```

```
60. 2001-12-25: SCORE = -2.99132e-09 PVALUE = 0
   53.1835% (284/534) of today's cases have Rash Syndrome = False and Unmapped Syndrome = False
   39.2165% (911/2323) of baseline cases have Rash Syndrome = False and Unmapped Syndrome = False

Rules 61-63 from 2001-12-26 to 2001-12-30 all have PVALUE = 0 and have the rule Hospital ID = 50

64. 2001-12-31: SCORE = -7.30783e-07 PVALUE = 0
   52.071% (352/676) of today's cases have Hemorrhagic Syndrome = True and Unmapped Syndrome = False
   41.6113% (1064/2557) of baseline cases have Hemorrhagic Syndrome = True and Unmapped Syndrome = False
```

From the output above, WSARE 2.0 produces a large number of rules that involves hospital IDs 14 and 50 because those two hospitals start providing data in 2002. These rules typically persist for about a month, at which point the new hospitals begin to appear in the baseline of WSARE 2.0. We speculate that the missing hospital IDs in rules 3-14 are due to hospital 14 coming online and a new hospital code not being available. The other rules produced by WSARE 2.0 are very different from those generated by WSARE 3.0. This difference is likely due to the fact that WSARE 3.0 considers the effects of the environmental attributes. The most interesting rules produced by WSARE 2.0 are rules 2 and 64. Rule 2 highlights the fact that more male patients with a home zip code in the northern half of the city appear in the EDs on 2001-05-01. Rule 64 indicates that an increase in the number of hemorrhagic syndromes have occurred. Both of these rules are unlikely to have been caused by environmental trends; they are simply anomalous patterns when compared against the baseline of WSARE 2.0. From our available resources, we are unable to determine if rules 2 and 64 are truly indicative of an outbreak.

4.3 Results from the Israel Center for Disease Control

The Israel Center for Disease Control evaluated WSARE 3.0 retrospectively using an unusual outbreak of influenza type B that occurred in an elementary school in central Israel (Kaufman et al., 2004). WSARE 3.0 was applied to patient visits to community clinics between the dates of May 24, 2004 to June 11, 2004. The attributes in this data set include the visit date, area code, ICD-9 code, age category, and day of week. The day of week was used as the only environmental attribute. WSARE 3.0 reported two rules with p-values at 0.002 and five other rules with p-values below 0.0001. Two of the five anomalous patterns with p-values below 0.0001 corresponded to the influenza outbreak in the data. The rules that characterized the two anomalous patterns consisted of the same three attributes of ICD-9 code, area code and age category, indicating that an anomalous pattern was found involving children aged 6-14 having viral symptoms within a specific geographic area. WSARE 3.0 detected the outbreak on the second day from its onset. The authors of (Kaufman et al., 2004) found the results from WSARE 3.0 promising and concluded that the algorithm was indeed able to detect an actual outbreak in syndromic surveillance data.

4.4 Summary of Results

Overall, WSARE 2.0 and 3.0 have been demonstrated to be more effective than univariate methods at finding anomalous patterns in multivariate, categorical data. The advantage that the WSARE algorithms have over univariate methods is their ability to identify the combination of attributes that characterize the most anomalous groups in the data rather than relying on a user to specify beforehand which combination of characteristics to monitor. WSARE 3.0 has a further advantage in its ability to account for temporal trends when producing the baseline distribution while WSARE 2.0 can be thrown off by these temporal trends when it uses raw historical data for the baseline.

We would like to emphasize the fact that WSARE 3.0 is not necessarily the best version of WSARE in all cases. WSARE 3.0 needs a large amount of data in order to learn the structure and parameters of its Bayesian network reliably, particularly if there are many attributes in the data. If WSARE 3.0 is intended to model long term trends such as seasonal fluctuations, several years worth of historical data are needed. Large amounts of historical data are not available in many cases, such as when a syndromic surveillance system needs to be set up from scratch in a few months for a major event like the Olympic games. In these scenarios, WSARE 2.0 may have an advantage over WSARE 3.0. This disadvantage of WSARE 3.0 highlights the fact that the learned Bayesian network only stores the posterior mean in the conditional probability tables of each node. Future work on WSARE 3.0 will involve accounting for the variance of the network parameters in the p-value calculation, perhaps using the approaches proposed by van Allen (2000), van Allen et al. (2001), and Singh (2004).

Moreover, WSARE 3.0 assumes that the environmental attributes are the only source of variation in the baseline distribution. If other hidden variables cause a significant amount of noise in the baseline, then WSARE 3.0 will not be very effective. In this situation, a better approach might be to use WSARE 2.0 with a baseline of raw historical data from a very recent time period. Finally, we do not recommend using WSARE 2.5 because the algorithm is unable to make predictions for days in which the combination of environmental attributes do not exist in historical data. The Bayesian network used by WSARE 3.0 is able to handle such situations and WSARE 3.0 effectively supersedes WSARE 2.5.

5. Finding Anomalous Patterns in Real-Valued Data

The WSARE algorithm can only be used on categorical data sets. If the data is entirely real-valued, the attributes can certainly be discretized in a pre-processing step before WSARE operates on the data. Discretization, however, treats all data points in the same discretization bin identically; the distances between data points in the same bin are lost. If these distances are important, then a real-valued version of WSARE is needed. Fortunately, the spatial scan statistic (Kulldorff, 1997) can be considered as the real-valued analog of WSARE.

The spatial scan statistic works on a geographic area A in which there is an underlying population n and within this population there is a count c of interest. The distribution of the counts c is assumed to follow either a Bernoulli model or a Poisson model. A window of variable size and shape then passes through the geographic area A. The crucial characteristic of this window is that the union of the areas covered by the window is the entire area A. Existing spatial scan statistic applications typically use window shapes of circles (Kulldorff, 1999) although ellipses (Kulldorff et al., 2002) and rectangles (Neill and Moore, 2004) have also been used. In order to set up the scan statistic, we need to define p as the probability of being a "count" within the scanning window. Furthermore, let q be the probability of being a "count" outside of the scanning window. Under the null hypothesis, $p = q$ while the alternative hypothesis is $p > q$. The spatial scan statistic then consists of the maximum likelihood ratio between L_W, the likelihood of the counts in the scanning window area W, and L_0, the likelihood under the null hypothesis. Equation 1 illustrates the spatial scan statistic in its general form, using the term W for the zone covered by a scanning window and

$\mathcal{W}$ for the entire collection of zones:

$$S_{\mathcal{W}} = max_{W \varepsilon \mathcal{W}} \frac{L(W)}{L_0}. \tag{1}$$

Since an analytical form for the distribution of the spatial scan statistic is not available, a Monte Carlo simulation is needed to obtain the significance of the hypothesis test. Typically 999 or 9999 replications of the data set are used for the simulation. In terms of computational complexity, the bottleneck for the algorithm is the Monte Carlo simulation.

The spatial scan statistic has been extended to three dimensions in the space-time scan statistic (Kulldorff, 1999, 2001). Instead of using a circular window over space, the scanning window is now a cylinder, with its circular base for the spatial dimension and its height over a time interval. Cylinders of varying heights and base radii are moved through space and time to find potential disease clusters.

Naive implementations of the spatial scan statistic and the space-time scan statistic are too computationally expensive for large data sets. Assuming that the circular windows are centered on an NxN grid and the dimensionality is D, the complexity is $O(RN^{2D})$ where R is the number of Monte Carlo simulations. Neill et al. (2005) have developed a fast spatial scan using overlap-kd trees that can reduce the complexity to $O(R(NlogN)^D)$ in the best case. The algorithms discussed so far find abnormally high density regions in data sets that are entirely real-valued. Efficiently finding anomalous patterns in a data set with a mixture of categorical and real-valued attributes remains an open problem.

6. Related Work

The task of detecting anomalous events in data is most commonly associated with monitoring systems. As a result, related work can be found in the domains of computer security, fraud detection, Topic Detection and Tracking (TDT) and fMRI analysis. In computer security, anomaly detection has been most prominent in intrusion detection systems, which identify intrusions by distinguishing between normal system behavior and behavior when security has been compromised (Lane and Brodley, 1999; Warrender et al., 1999; Eskin, 2000; Lee et al., 2000; Maxion and Tan, 2002; Kruegel and Vigna, 2003). In other related security work, Cabuk et al. (2004) describe methods to detect IP covert timing channels, which surreptitiously use the arrival pattern of packets to send information. As in computer security, automated fraud detection systems differentiate between normal and unusual activity on a variety of data such as cellular phone calls (Fawcett and Provost, 1997) and automobile insurance fraud (Phua et al., 2004). TDT is the task of identifying the earliest report of a previously unseen news story from a sequence of news stories. Clustering approaches are typically used in TDT (Yang et al., 1998; Zhang et al., 2005). Finally, anomalous event detection has also been used in fMRI analysis to identify regions of increased brain activity corresponding to given cognitive tasks (Neill et al., 2005).

In general, WSARE can be applied to data from these different domains as long as the data and the anomalous events satisfy several criteria. WSARE is intended to operate on categorical, case-level records in which the presence of a record can be considered an event. For instance, in ED data, an event is defined as the appearance of a person at the ED since it provides a signal of the community health and we are interested in the characteristics of that person. Secondly, WSARE only finds differences between the recent data and the baseline data. If we consider the baseline data to be a “class”, then WSARE looks for deviations from a single class. Some domains, such as TDT, require comparisons between several classes. For instance, the current news story needs to be compared against several categories of news stories. Thirdly, as was discussed in Section 2.6, WSARE's running time depends on the number of attributes and the number of values each attribute can take. If the number of attributes and the number of values for each attribute are too high, WSARE may not finish in a reasonable amount of time. Some domains require the running time of the detection algorithm to be a few seconds or less in order for the entire detection system to be effective. In these situations, using WSARE is not appropriate. On the other hand, for domains such as biosurveillance, the running time of WSARE is acceptable since it takes approximately a minute to a few minutes to complete on real ED data sets. Finally, WSARE treats each record in the data independently of the other records. If a sequence of records is highly indicative of, for instance, a security breach in a network, WSARE will not be able to detect this pattern.

Other related work can also be found in the area of stream mining. In stream mining, the focus is on the online processing of large amounts of data as it arrives. Many algorithms have been developed to detect anomalies in the current stream of data. Ma and Perkins (2003) develop a novelty detection algorithm based on online support vector regression. Anomalies can also be characterized by an abnormal burst of data. The technique described by Zhu and Shasha (2003) simultaneously monitors windows of different sizes and reports those that have an abnormal aggregation of data. A density estimation approach is used by Aggarwal (2003) to help visualize both spatial and temporal trends in evolving data streams. Finally, Hulten et al. (2001) present an efficient algorithm for mining decision trees from continuously changing data streams. While this work is primarily concerned with maintaining an up-to-date concept, detecting concept drift is similar to detecting changes in a data stream. WSARE 3.0 cannot be directly applied to stream mining because the amount of historical data needed to create the baseline distribution is typically not accessible in a stream mining context. However, WSARE 2.0 could possibly be modified for a stream mining application.

In the following paragraphs, we will briefly review methods that have been used for the detection of disease outbreaks. Readers interested in a detailed survey of biosurveillance methods can be found in (Wong, 2004) and (Moore et al., 2003). The majority of detection algorithms in biosurveillance operate on univariate time series data. Many of these univariate algorithms have been taken from the field of Statistical Quality Control and directly applied to biosurveillance. The three most common techniques from Statistical Quality Control include the Shewhart control chart (Montgomery, 2001), CUSUM (Page, 1954; Hutwagner et al., 2003), and EWMA (Roberts, 1959; Williamson and Hudson, 1999). Although these three algorithms are simple to implement, they have difficulty dealing with temporal trends. Univariate algorithms based on regression and time series models, on the other hand, are able to model explicitly the seasonal and day of week effects in the data. The Serfling method (Serfling, 1963) uses sinusoidal components in its regression equation to model the seasonal fluctuations for influenza. A Poisson regression model that included a day of week term as a covariate was demonstrated to be a fairly capable detector in (Buckeridge et al., 2005). As for time series models, the ARIMA and SARIMA models (Choi and Thacker, 1981; Watier et al., 1991; Reis and Mandl, 2003) are commonly used in biosurveillance to deal with temporal trends. Recently, wavelets (Goldenberg et al., 2002; Zhang et al., 2003) have been used as a preprocessing step to handle temporal fluctuations including unusually low values due to holidays.

The most common algorithm used in biosurveillance of spatial data is the Spatial Scan Statistic (Kulldorff, 1997), which has already been discussed. The Spatial Scan Statistic has been generalized to include a time dimension (Kulldorff, 2001) such that the algorithm searches for cylinders in spatio-temporal data. Recent work has improved the speed of the Spatial Scan method using an overlap-kd tree structure (Neill and Moore, 2004; Neill et al., 2005).

The algorithms mentioned thus far have only looked at either univariate or spatial data. Only a few multivariate biosurveillance algorithms that consider spatial, temporal, demographic, and symptomatic attributes for individual patient cases currently exist. BCD (Buckeridge et al., 2005) is a multivariate changepoint detection algorithm that monitors in a frequentist manner whether a Bayesian network learned from past data (during a “safe” training period) appears to have a distribution that differs from the distribution of more recent data. If so, then an anomaly may have occurred. The Bayesian Aerosol Release Detector (BARD) (Hogan et al., 2004) is an algorithm specifically designed to detect an outbreak of inhalational anthrax due to atmospheric dispersion of anthrax spores. BARD combines information from ED visits, recent meteorological data, and spatial and population information about the region being monitored in order to determine if an anthrax attack has occurred. Finally, PANDA (Cooper et al., 2004) is a population-based anomaly detection algorithm that uses a massive causal Bayesian network to model each individual in the region under surveillance. By modeling at the individual level, PANDA is able to coherently represent different types of background knowledge in its model. For example, spatio-temporal assumptions about a disease outbreak can be incorporated as prior knowledge. In addition, the characteristics of each individual, such as their age, gender, home zip, symptom information and admission date to the ED can be used to derive a posterior probability of an outbreak.

There are two algorithms that are similar to the approach taken by WSARE. Contrast set mining (Bay and Pazzani, 1999) finds rules that distinguish between two or more groups using a pruning algorithm to reduce the exponential search space. This optimization prunes away rules whose counts are too small to yield a valid Chi Square test. Many of these rules are interesting to WSARE. Multiple hypothesis testing problems are addressed in contrast set mining through a Bonferroni correction. In health care, Brossette et al. use association rules for hospital infection control and public health surveillance (Brossette et al., 1998). Their work is similar to WSARE 2.0 (Wong et al., 2002), with the main difference being the additional steps of the randomization test and FDR in WSARE.

7. Conclusions

WSARE approaches the problem of early outbreak detection on multivariate surveillance data using two key components. The first component is association rule search, which is used to find anomalous patterns between a recent data set and a baseline data set. The contribution of this rule search is best seen by considering the alternate approach of monitoring a univariate signal. If an attribute or combination of attributes is known to be an effective signal for the presence of a certain disease, then a univariate detector or a suite of univariate detectors that monitors this signal will be an effective early warning detector for that specific disease. However, if such a signal is not known beforehand, then the association rule search will determine which attributes are of interest. We intend WSARE to be a general purpose safety net to be used in combination with a suite of specific disease detectors. Thus, the key to this safety net is to perform non-specific disease detection and notice any unexpected patterns.

With this perspective in mind, the fundamental assumption to our association rule approach is that an outbreak in its early stages will manifest itself in categorical surveillance data as an anomalous cluster in attribute space. For instance, a localized gastrointestinal outbreak originating at a popular restaurant in zipcode X would likely cause an upswing in diarrhea cases involving people with home zipcode X. These cases would appear as a cluster in the categorical attributes of $Home\ Zip\ Code = X$ and $Symptom = Diarrhea$. The rule search allows us to find the combination of attributes that characterize the set of cases from recent data that are most anomalous when compared to the baseline data. The nature of the rule search, however, introduces the problem of multiple hypothesis testing to the algorithm. Even with purely random data, the best scoring rule may seem like a truly significant anomalous pattern. We are careful to evaluate the statistical significance of the best scoring rule using a randomization test in which the null hypothesis is the independence of date and case attributes.

The second major component of WSARE is the use of a Bayesian network to model a baseline that changes due to temporal fluctuations such as seasonal trends and weekend versus weekday effects. In WSARE 3.0, attributes are divided into environmental and response attributes. Environmental attributes, such as season and day of week, are attributes which are responsible for the temporal trends while response attributes are the non-environmental attributes. When the Bayesian network structure is learned, the environmental attributes are not permitted to have parents because we are not interested in predicting their distributions. Instead, we want to determine how the environmental attributes affect the distributions of the response attributes. WSARE 3.0 operates on an assumption that the environmental attributes account for the majority of the variation in the data. Under this assumption, the ratios compared in the rule search should remain reasonably stable over historical time periods with similar environmental attribute values. As an example, if the current day is a winter Friday and we use season and day of week as environmental attributes, then the fraction of male senior citizens, for instance, showing up at an ED to the total number of patients should remain roughly stable over all winter Fridays in the historical period over which the Bayesian network is learned. Once the Bayesian network structure is learned, it represents the joint probability distribution of the baseline. We can then condition on the environmental attributes to produce the baseline given the environment for the current day.

Multivariate surveillance data with known outbreak periods is extremely difficult to obtain. As a result, we resorted to evaluating WSARE on simulated data. Although the simulators do not reflect real life, detecting an outbreak in our simulated data sets is a challenging problem for any detection algorithm. We evaluated WSARE on the CityBN simulator, which was implemented to generate surveillance data which contained temporal fluctuations due to day of week effects and seasonal variations of background illnesses such as flu, food poisoning and allergies. Despite the fact that the environmental attributes used by WSARE 3.0 did not account for all of the variation in the data, WSARE 3.0 detected the anthrax outbreaks with nearly the optimal detection time and a very low false positive rate. We show that WSARE 3.0 outperformed three common univariate detection algorithms in terms of false positives per month and detection time. WSARE 3.0 also produced a better AMOC curve than WSARE 2.0 because the latter was thrown off by the temporal trends in the data. Finally, the Bayesian network provided some smoothing to the baseline distribution which enhanced WSARE 3.0's detection capability as compared to that of WSARE 2.5.

WSARE has been demonstrated to outperform traditional univariate methods on simulated data in terms of false positives per month and detection time. Its performance on real world data requires further evaluation. Currently, WSARE is part of the collection of biosurveillance algorithms in the RODS system (Real-time Outbreak Detection System, 2004). WSARE 2.0 was deployed to monitor ED cases in western Pennsylvania and Utah. It was also used during the 2002 Salt Lake City winter

Olympics. WSARE 3.0 is currently being used as a tool for analysis of public health data by several American state health departments and by the Israel Center for Disease Control.

Acknowledgments

This research was supported by DARPA grant F30602-01-2-0550, the State of Pennsylvania, and by the National Science Foundation grant IIS-0325581. Many thanks to Rich Tsui, Bob Olszewski, Jeremy Espino, Jeff Schneider and Robin Sabhnani for their helpful comments and technical expertise.

References

45 CFR Parts 160 through 164, April 2003.
Available at http://www.hhs.gov/ocr/combinedregtext.pdf.

Charu C. Aggarwal. A framework for diagnosing changes in evolving data streams. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 575–586, New York, NY, 2003. ACM Press.

Stephen D. Bay and Michael J. Pazzani. Detecting change in categorical data: Mining contrast sets. In *Knowledge Discovery and Data Mining*, pages 302–306, 1999.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society, Series B.*, 57:289–300, 1995.

Chris M. Bishop. Novelty detection and neural network validation. *IEEE Proceedings - Vision, Image and Signal Processing*, 141(4):217–222, August 1994.

Carlo E. Bonferroni. Teoria statistica delle classi e calcolo delle probabilità. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze*, 8:3–62, 1936.

George E. P. Box and Gwilym M. Jenkins. *Time series analysis: Forecasting and control*. Holden-Day, San Francisco, 1976.

Stephen E. Brossette, Alan P. Sprague, J. Michael Hardin, Ken B. Waites, Warren T. Jones, and Stephen A. Moser. Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association*, 5:373–381, 1998.

David L. Buckeridge, Howard Burkom, Murray Campbell, William R. Hogan, and Andrew W. Moore. Algorithms for rapid outbreak detection: a research synthesis. *Biomedical Informatics*, 38(2):99–113, 2005.

Serdar Cabuk, Carla E. Brodley, and Clay Shields. IP covert timing channels: design and detection. In *Proceedings of the 11th ACM conference on Computer and Communications Security*, pages 178–187, New York, NY, 2004. ACM Press.

Keewhan Choi and Stephen B. Thacker. An evaluation of influenza mortality surveillance, 1962-1979 I. time series forecasts of expected pneumonia and influenza deaths. *American Journal of Epidemiology*, 113(3):215–226, 1981.

Gregory F. Cooper, Denver H. Dash, John D. Levander, Weng-Keen Wong, William R. Hogan, and Michael M. Wagner. Bayesian biosurveillance of disease outbreaks. In Max Chickering and Joseph Halpern, editors, *Proceedings of the Twentieth Conference on Uncertainty in Artificial Intelligence*, pages 94–103, Banff, Alberta, Canada, 2004. AUAI Press.

Eleazar Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proceedings of the 2000 International Conference on Machine Learning (ICML-2000)*, Palo Alto, CA, July 2000.

Tom Fawcett and Foster Provost. Adaptive fraud detection. *Data Mining and Knowledge Discovery*, 1(3):291–316, 1997.

Tom Fawcett and Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In Chaudhuri and Madigan, editors, *Proceedings on the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, San Diego, CA, 1999. URL `citeseer.nj.nec.com/fawcett99activity.html`.

Anna Goldenberg, Galit Shmueli, and Rich Caruana. Using grocery sales data for the detection of bio-terrorist attacks. Submitted to Statistics in Medicine, 2003.

Anna Goldenberg, Galit Shmueli, Richard A. Caruana, and Stephen E. Fienberg. Early statistical detection of anthrax outbreaks by tracking over-the-counter medication sales. *Proceedings of the National Academy of Sciences*, 99(8):5237–5240, April 2002. http://www.pnas.org/cgi/doi/10.1073/pnas.042117499.

Phillip Good. *Permutation Tests - A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer-Verlag, New York, 2nd edition, 2000.

Greg Hamerly and Charles Elkan. Bayesian approaches to failure prediction for disk drives. In *Proceedings of the eighteenth international conference on machine learning*, pages 202–209. Morgan Kaufmann, San Francisco, CA, 2001.

William R. Hogan, Gregory F. Cooper, Garrick L. Wallstrom, and Michael M. Wagner. The Bayesian aerosol release detector. In *Proceedings of the Third National Syndromic Surveillance Conference [CD-ROM]*, Boston, MA, 2004. Fleetwood Multimedia, Inc.

Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. In *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 97–106, San Francisco, CA, 2001. ACM Press. URL `citeseer.ist.psu.edu/hulten01mining.html`.

Lori Hutwagner, William Thompson, G. Matthew Seeman, and Tracee Treadwell. The bioterrorism preparedness and response early aberration reporting system (EARS). *Journal of Urban Health*, 80(2):i89–i96, 2003.

Zalman Kaufman, Erica Cohen, Tamar Peled-Leviatan, Hanna Lavi, Gali Aharonowitz, Rita Dichtiar, Michal Bromberg, Ofra Havkin, Yael Shalev, Rachel Marom, Varda Shalev, Joshua Shemer, and Manfred S Green. Evaluation of syndromic surveillance for early detection of bioterrorism using a localized, summer outbreak of Influenza B. In *Proceedings of the Third National Syndromic Surveillance Conference [CD-ROM],* Boston, MA, 2004. Fleetwood Multimedia Inc. Poster Presentation.

Christopher Kruegel and Giovanni Vigna. Anomaly detection of web-based attacks. In *Proceedings of the* 10th *ACM Conference on Computer and Communication Security (CCS '03)*, pages 251–261, Washington, DC, October 2003. ACM Press.

Martin Kulldorff. A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26 (6):1481–1496, 1997.

Martin Kulldorff. Spatial scan statistics: models, calculations, and applications. In J. Glaz and N. Balakrishnan, editors, *Scan Statistics and Applications*, pages 303–322. Birkhauser, Boston, MA, 1999.

Martin Kulldorff. Prospective time periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society, Series A*, 164:61–72, 2001.

Martin Kulldorff, Lan Huang, and Linda Pickle. An elliptic spatial scan statistic and its application to breast cancer mortality data in the northeastern united states. In *Proceedings of the National Syndromic Surveillance Conference*, 2002.

Terran Lane and Carla E. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security*, 2:295–331, 1999.

Wenke Lee, Salvatore J. Stolfo, and Kui W. Mok. Adaptive intrusion detection: A data mining approach. *Artificial Intelligence Review*, 14(6):533–567, 2000. URL `citeseer.ist.psu.edu/lee00adaptive.html`.

Junshui Ma and Simon Perkins. Online novelty detection on temporal sequences. In *Proceedings on the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 613–618, New York, NY, 2003. ACM Press.

Oded Maron and Andrew W. Moore. The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, 11(1-5):193–225, 1997.

Roy A. Maxion and Kymie M.C. Tan. Anomaly detection in embedded systems. *IEEE Trans. Comput.*, 51(2):108–120, 2002. ISSN 0018-9340.

Christopher J. Miller, Christopher Genovese, Robert C. Nichol, Larry Wasserman, Andrew Connolly, Daniel Reichart, Andrew Hopkins, Jeff Schneider, and Andrew Moore. Controlling the false-discovery rate in astrophysical data analysis. *The Astronomical Journal*, 122:3492–3505, Dec 2001.

Douglas C. Montgomery. *Introduction to Statistical Quality Control*. John Wiley and Sons, Inc., 4th edition, 2001.

Andrew Moore and Mary Soon Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8:67–91, March 1998.

Andrew Moore and Weng-Keen Wong. Optimal reinsertion: A new search operator for accelerated and more accurate Bayesian network structure learning. In T. Fawcett and N. Mishra, editors, *Proceedings of the 20th International Conference on Machine Learning*, pages 552–559, Menlo Park, CA, August 2003. AAAI Press.

Andrew W. Moore, Gregory F. Cooper, Rich Tsui, and Michael M. Wagner. Summary of biosurveillance-related technologies. Technical report, Realtime Outbreak and Disease Surveillance Laboratory, University of Pittsburgh, 2003. Available at http://www.autonlab.org/autonweb/showPaper.jsp?ID=moore-biosurv.

Farzad Mostashari and Jessica Hartman. Syndromic surveillance: a local perspective. *Journal of Urban Health*, 80(2):i1–i7, 2003.

Daniel B. Neill and Andrew W. Moore. A fast multi-resolution method for detection of significant spatial disease clusters. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, Cambridge, MA, 2004.

Daniel B. Neill, Andrew W. Moore, Francisco Pereira, and Tom Mitchell. Detecting significant multidimensional spatial clusters. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

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

Clifton Phua, Damminda Alahakoon, and Vincent Lee. Minority report in fraud detection: classification of skewed data. *ACM SIGKDD Explorations Newsletter Special issue on learning from imbalanced datasets*, 6(1):50–59, 2004.

Real-time Outbreak Detection System, 2004. Online at http://www.health.pitt.edu/rods/default.htm.

Ben Y. Reis and Kenneth D. Mandl. Time series modeling for syndromic surveillance. *BMC Medical Informatics and Decision Making*, 3(2), 2003. http://www.biomedcentral.com/1472-6947/3/2.

S. W. Roberts. Control chart tests based on geometric moving averages. *Technometrics*, 1:239–250, 1959.

Robert E. Serfling. Methods for current statistical analysis of excess pneumonia-influenza deaths. *Public Health Reports*, 78:494–506, 1963.

Ajit P. Singh. What to do when you don't have much data: Issues in small sample parameter learning in Bayesian Networks. Master's thesis, Dept. of Computing Science, University of Alberta, 2004.

Daniel M. Sosin. Draft framework for evaluating syndromic surveillance systems. *Journal of Urban Health*, 80(2):i8–i13, 2003.

Fu-Chiang Tsui, Michael M. Wagner, Virginia Dato, and Chung-Chou Ho Chang. Value of ICD-9-coded chief complaints for detection of epidemics. In S Bakken, editor, *Journal of the American*

Medical Informatics Association, Supplement i ssue on the Proceedings of the Annual Fall Symposium of the American Medical Inf ormatics Association, pages 711–715. Hanley and Belfus, Inc, 2001.

Tim van Allen. Handling uncertainty when you're handling uncertainty: Model selection and error bars for belief networks. Master's thesis, Dept. of Computing Science, University of Alberta, 2000.

Tim van Allen, Russell Greiner, and Peter Hooper. Bayesian error-bars for belief net inference. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, Aug 2001.

Christina Warrender, Stephanie Forrest, and Barak Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 133–145. IEEE Computer Society, 1999.

Laurence Watier, Sylvia Richardson, and Bruno Hubert. A time series construction of an alert threshold with application to s. bovismorbificans in france. *Statistics in Medicine*, 10:1493–1509, 1991.

G. David Williamson and Ginner Weatherby Hudson. A monitoring system for detecting aberrations in public health surveillance reports. *Statistics in Medicine*, 18:3283–3298, 1999.

Weng-Keen Wong. *Data mining for early disease outbreak detection*. PhD thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, Pennsylvania, 2004.

Weng-Keen Wong, Andrew Moore, Gregory Cooper, and Michael Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In Tom Fawcett and Nina Mishra, editors, *Proceedings of the Twentieth International Conference on Machine Learning*, pages 808–815, Menlo Park, California, August 2003. AAAI Press.

Weng-Keen Wong, Andrew W. Moore, Gregory Cooper, and Michael Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 217–223. MIT Press, 2002.

Yiming Yang, Tom Pierce, and Jiame Carbonell. A study on retrospective and on-line event detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 28–36, New York, NY, 1998. ACM Press.

Jian Zhang, Zoubin Ghahramani, and Yiming Yang. A probabilistic model for online document clustering with application to novelty detection. In Lawrence K. Saul, Yair Weiss, and Léon Bottou, editors, *Advances in Neural Information Processing Systems 17*. MIT Press, Cambridge, MA, 2005.

Jun Zhang, Fu-Chiang Tsui, Michael M. Wagner, and William R. Hogan. Detection of outbreaks from time series data using wavelet transform. In *Proc AMIA Fall Symp*, pages 748–752. Omni Press CS, 2003.

Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proceedings of the ninth ACM SIGKDD international conference on knowledge discovery and data mining*, pages 336–345, New York, NY, 2003. ACM Press.