Bayesian Network Anomaly Pattern Detection for Disease Outbreaks

Weng-Keen Wong WKW@CS.CMU.EDU
Andrew Moore AWM@CS.CMU.EDU
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15213 USA

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

Abstract

Early disease outbreak detection systems typically monitor health care data for irregularities by comparing the distribution of recent data against a baseline distribution. Determining the baseline is difficult due to the presence of different trends in health care data, such as trends caused by the day of week and by seasonal variations in temperature and weather. Creating the baseline distribution without taking these trends into account can lead to unacceptably high false positive counts and slow detection times. This paper replaces the baseline method of (Wong et al., 2002) with a Bayesian network which produces the baseline distribution by taking the joint distribution of the data and conditioning on attributes that are responsible for the trends. We show that our algorithm, called WSARE 3.0, is able to detect outbreaks in simulated data with almost the earliest possible detection time while keeping a low false positive count. We also include the results of running WSARE 3.0 on real Emergency Department data.

1. Introduction

Early disease outbreak detection systems monitor health care data for any irregularities due to the onset of an epidemic. These systems compare recent data against baseline data and raise an alarm if the deviations from the baseline are significant. On the surface, this problem seems like a traditional anomaly detection task. Typical anomaly detection, however, finds isolated anomalies in feature space which are not at all indicative of a disease outbreak. As an example, suppose we apply a traditional anomaly detection technique to Emergency Department (ED) records. We might then find an unusual record such as a patient that was over a hundred years old living in a sparsely populated region. Instead of finding such unusual isolated cases, early outbreak detection algorithms are interested in finding specific groups whose profile is anomalous relative to their typical profile. This type of *anomalous pattern* detection is similar to work done on mining contrast sets (Bay & Pazzani, 1999). 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.

Determining the baseline distribution is a problem that all early detection systems face. This distribution is usually obtained from a period of time in the past when no epidemics are known to happen. However, determining this distribution is extremely difficult due to the different trends present in health care data. Seasonal variations in weather and temperature can dramatically alter the distribution of health care data. For example, flu season typically occurs during mid-winter, resulting in an increase in ED cases involving cough and fever symptoms. 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 1, 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

Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003), Washington DC, 2003.

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 at a very late stage.

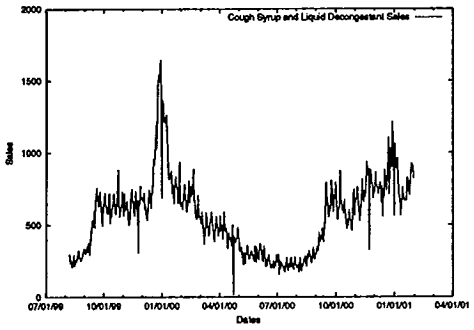

Figure 1: Cough syrup and liquid decongestant sales from (Goldenberg et al., 2002)

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, makes the algorithm susceptible to 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 (Wong et al., 2002), we made the baseline distribution to be data obtained 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. We concede that this baseline was chosen somewhat arbitrarily. Ideally, the detection system should determine the baseline automatically.

In this paper, we propose building a Bayesian network to represent the joint 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 features are precisely those attributes that account for trends in the data, such as the season, the current flu level and the day of week.

2. WSARE 3.0

The WSARE algorithm, which stands for "What's Strange About Recent Events", operates on discrete, multidimensional temporal data sets. This algorithm compares recent data against a baseline distribution with the aim of finding rules that summarize significant patterns of anomalies. Each rule takes the form $X_i = V_i^j$, where X_i is the ith feature and V_i^j is the jth value of that feature. Multiple components are joined together by a logical AND. For example, a two component rule would be Gender = Male AND Home Location = NW. Due to computational issues, the number of components for each rule is two or less. It is helpful to think of the rules as SQL SELECT queries. They characterize a subset of the data having records with attributes matching the components of the rule.

At this point we will provide an overview of our extended WSARE algorithm, which we will refer to as WSARE 3.0. We will refer to the WSARE algorithm in (Wong et al., 2002) as version 2.0. As in the previous version, WSARE 3.0 operates on a daily basis, in which for each day, the algorithm treats records from the past 24 hours as recent events. Using historical data beyond the past 24 hours, WSARE 3.0 then creates a baseline distribution which is assumed to capture the usual behaviour of the system being monitored under the environmental conditions of the current day. Once the baseline distribution has been created, the algorithm considers all possible one and two component rules over events occurring on the current day. The rules are scored with a scoring function that assigns high scores to rules corresponding to subsets of data that have unusual proportions when compared against the baseline distribution. The rule with the highest score for the day has its p-value calculated using a randomization test. If this p-value is lower than a specified threshold, an alert is raised.

The component that differentiates WSARE 3.0 from WSARE 2.0 is the step that creates the baseline distribution. The previous version simply used data from 35, 42, 49 and 56 days prior to the current day. Version 3.0 builds a Bayesian network from all data prior to the past 24 hours and then represents the baseline distribution as a data set sampled from the Bayesian network. We will describe this step in detail below, while the other parts of the algorithm will be described briefly since they are thoroughly discussed in (Wong

et al., 2002).

2.1. 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 features that cause trends in the data, and *response attributes*, which are the remaining features. 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 in this paper, WSARE 3.0 was able to overcome some hidden environmental attributes in our simulator.

The network structure is learned from data using an efficient structure search algorithm called Optimal Reinsertion (Moore & Wong, 2003) based on ADTrees (Moore & Lee, 1998). Environmental attributes in the structure 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. The structure search also exploits this constraint by avoiding search paths that assign parents to the environmental attributes.

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. Incorporating such knowledge into the Bayesian network can aid in detecting anomalies other than the ones we already know about. 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.

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, then 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 $P(X_1, ..., X_n \mid \text{Season} = \text{Winter, Day of Week} = \text{Weekday})$ from the Bayesian network. For simplicity, we represent the conditional distribution as a data set formed by sampling 10000 records from the Bayesian network conditioned on the environmental attributes. The size of this sampled data set has to be large enough to ensure that samples with rare combinations of attributes will be present, hence the choice of 10000 records. Note that this sampling is easily done in an efficient manner since environmental attributes have no parents. 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 chose to use a sampled data set instead of using inference mainly because sampling requires $DB_{baseline}$ to be generated only once and then we can use it to obtain the p-values for all the rules. With inference, we would need to sample a different $DB_{baseline}$ for every rule in order to perform the randomization test described in the next section. While a sampled data set provides the simplest way of obtaining the conditional distribution, we have not, however, completely ignored the possibility of using inference to speed up this process. We would like to investigate this direction further in our future work.

2.2. Scoring each rule

Finding the best rule for the current day requires comparing how different recent records are from the baseline. This step requires a two-by-two contingency table to be established for each rule. Suppose the rule is Respiratory Syndrome = True. We set up a contingency table as shown in Table 1 with the cells containing counts for records matching and not matching the rule for both data sets DB_{recent} and $DB_{baseline}$. Let C_{recent} be the count for DB_{recent} and $C_{baseline}$ be that for $DB_{baseline}$.

The score of a rule is determined through 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 the difference between the counts for the recent period and those for the baseline. This test produces a p-value that determines the significance of the anomalies found by the rule. This p-value will be referred to as the *score* in order to distinguish it from p-values used later on. We use the Chi Squared test whenever its assumptions are not violated. Since we are searching for anomalies, the counts in the contingency table are frequently small numbers and we resort to using Fisher's Exact Test (Good, 2000). Running Fisher's Exact Test on Table 1 yields a score of 0.00000464, which indicates that C_{recent} for the rule Respiratory Syndrome = *True* is

anomalous when compared to that of $C_{baseline}$.

Table 1. A Sample 2x2 Contingency Table

	C_{recent}	$C_{baseline}$
$RespiratorySyndrome = True$	58	653
$RespiratorySyndrome \neq True$	409	9347

When scoring a rule, we are making a comparison between current data and data obtained from the Bayesian network, which will differ slightly from the true baseline distribution because the network structure was learned from a finite amount of data. In the future, we would like to incorporate some notion of the Bayesian network's uncertainty into these scores, perhaps by reporting a confidence interval.

2.3. Obtaining the p-value for each rule

The score produced by the previous step cannot be accepted at face value as a p-value because of a multiple hypothesis testing problem. Suppose we follow the standard practice of rejecting the null hypothesis when the p-value is $< \alpha$, where $\alpha = 0.05$. When only one hypothesis test is performed, the probability of making a false discovery 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 making a false discovery could be as bad as $1 - (1 - 0.05)^{1000} \approx 1$, which is much greater than 0.05 (Miller et al., 2001).

We need to add an adjustment for the multiple hypothesis tests. This problem can be addressed using a Bonferroni correction (Bonferroni, 1936) but this approach can be unnecessarily conservative. Instead, we use a randomization test (Good, 2000) in which the date is assumed to be independent of the other features. In this test, the non-date features of both DB_{recent} and $DB_{baseline}$ remain the same but the dates are shuffled between the two data sets, resulting in two newly randomized data sets RDB_{recent} and $RDB_{baseline}$ respectively. RDB_{recent} and $RDB_{baseline}$ will now both contain records with dates from the original recent period and from the baseline period. The procedure is described below.

Let UCP = Uncompensated p-value (i.e., the score as defined above).

For j = 1 to 1000
  Let $DB = DB_{recent} \cup DB_{baseline}$
  Produce RDB^j_{recent} and $RDB^j_{baseline}$ from DB.
  Let $RDB^j = RDB^j_{recent} \cup RDB^j_{baseline}$
  Let BR^j = Best rule on RDB^j
  Let UCP^j = Uncompensated p-value of BR^j on RDB^j

Let the compensated p-value of BR be CPV, that is

$$CPV = \frac{\#\text{ of Rand Tests in which } UCP^j > UCP}{\#\text{ of Rand Tests}}$$

CPV estimates the chance of seeing an uncompensated p-value as good as UCP if in fact there was no relationship between the date and the other features.

3. Evaluation

3.1. The Simulator

We evaluated WSARE 3.0 on a small scale city simulator. Our city 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. We ran 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 made 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.

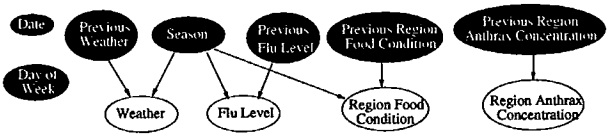

Figure 2. City Status Bayesian Network

We implemented this city simulation using a Bayesian network, as shown in Figure 2. We will use the convention that any nodes shaded black in the Bayes network 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, 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 Bayesian network have an arity of 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 Bayesian network. These records are stored in the City Status (CS) data set. The simulated anthrax release was 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 a randomly chosen length of time.

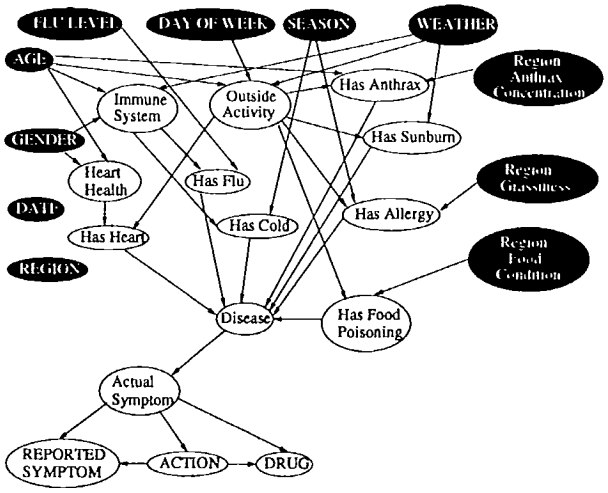

Figure 3. Patient Status Bayesian Network

Table 2. 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 Bayesian network used in our simulation produces individual health care cases. Figure 3 depicts the Patient Status (PS) network. On each day, for each person in each region, we sample the individual's values from this network. The black nodes first have their values assigned from the CS data set record for the current day. The white nodes are then sampled from the PS network. Each individual's health profile for the day is thus generated. The disease node indicates the status of each person in the simulation. 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 an individual has more than one disease, the disease node picks the disease with the highest precedence. A sick individual then exhibits one of the following symptoms: none, respiratory problems, nausea, or a rash. 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 include 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 3 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 generated daily by the PS network is typically in the range of 30 to 50 records. Table 2 contains two examples of records in the PS data set.

3.2. Algorithms

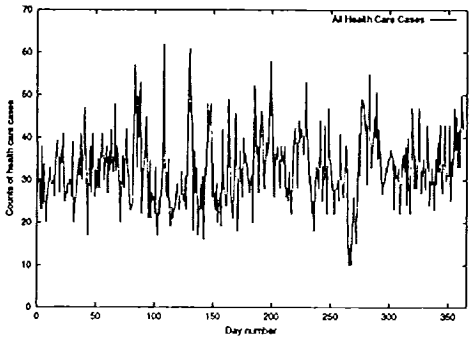

Figure 4: Daily Counts of Health Care Data

We ran five detection algorithms on 100 different PS data sets. Each data set was generated for a two year period, beginning on January 1, 2002 and ending December 31, 2003. The detection algorithms trained on data from the first year until the current day while the second year was used for evaluation. The anthrax release was randomly chosen in the period between January 1, 2003 to December 31, 2003.

We tried to simulate anthrax attacks that are not trivially detectable. Figure 4 plots the total count of health care cases on each day during the evaluation period. 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 for Fig-

ure 4 occurred on day 276. Occasionally the anthrax releases affect such a limited number of people that it was virtually undetectable. Consequently, we only used data sets with more than nine reported anthrax cases on any day during the attack period.

The Standard Algorithm The first algorithm used is a common anomaly detection algorithm which we will call the Standard Algorithm. 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 Standard Algorithm raises an alarm. We used a training period of January 1, 2002 to December 31, 2002.

WSARE 2.0 WSARE 2.0 was also evaluated, using a baseline distribution of records from 7, 14, 21 and 28 days before the current day. The attributes used by WSARE 2.5 and 3.0 as environmental attributes were ignored by WSARE 2.0. If these attributes were not ignored, WSARE 2.0 would report many trivial anomalies. For instance, suppose the environmental attribute Day of Week = Sunday for the current day. If this attribute were not ignored, WSARE 2.0 would notice that 100% of the records for the current day had Day of Week = Sunday while only 14.2% of records in the baseline data set matched 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 used 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 match, WSARE 2.5 cannot 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 beginning of the current day. We hypothesized that WSARE 3.0 will 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 allowed WSARE 3.0 to learn a network structure from scratch once every 30 days. On intermediate days, WSARE 3.0 simply updates the parameters of the previously learned network without altering its structure. We also ran a version of WSARE 3.0 which used a Bonferroni correction instead of a randomization test.

3.3. Real Emergency Department Data

We also tested out the performance of WSARE 3.0 on real ED data from a major US city. The database used contained almost seven years worth of data, with personal identifying information excluded in order to protect patient confidentiality. The features in this database include date of admission, coded hospital ID, age decile, gender, syndrome information and both home location and work location on a latitude-longitude grid. WSARE was run on data from the year 2001 and was allowed to use over five full years worth of training data from the start of 1996 to the current day. The environmental attributes used were 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 used 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 False Discovery Rate (Benjamini & Hochberg, 1995) algorithm with $\alpha = 0.1$ to compensate for another layer of multiple hypothesis testing when dealing with historical data as described in (Wong et al., 2002).

4. Results

Our evaluation criteria examines the algorithms' performance in terms of detection time versus false positives over p-values ranging from between 0.01 to 0.15 in 0.01 increments. The lower p-values yield lower false

positives and higher detection times while the converse is true with higher p-values. Figure 5 fills in the lines to display the asymptotic behaviour of the algorithms. 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.

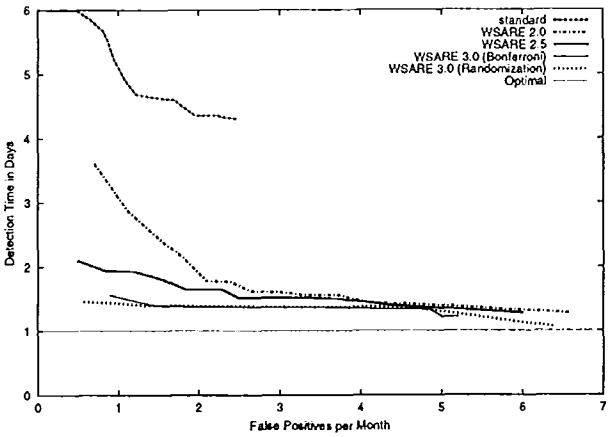

Figure 5: Asymptotic Behavior of Algorithms for Simulated Data

From the results of our simulation, WSARE 2.5 and both versions of WSARE 3.0 outperform the other two algorithms in terms of the detection time and false positive tradeoff. On average, WSARE 2.5 and WSARE 3.0 were able to detect the anthrax release within a period of one to two days. Both the Standard Algorithm and WSARE 2.0 were thrown off by the periodic trends present in the PS data. We had 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 outperforms WSARE 2.5, we conclude that in some cases, our proposition was true. In others, the p-values estimated by WSARE 2.5 were 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 were 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.

The Bonferroni-correction version of WSARE 3.0 had nearly identical performance to the randomization test version. We suspect that as the number of attributes in the data increases, thereby increasing the number of hypothesis tests, the Bonferroni correction becomes more conservative and unable to detect some attacks. We would like to investigate further the differences between these two approaches.

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

1. Sat 2001-02-13: SCORE = -0.00000004 PVALUE = 0.00000000
   14.80% (74/500) of today's cases have Viral Syndrome = True
   and Encephalitic Prodome = False
   7.42% (742/10000) of baseline have Viral Syndrome = True
   and Encephalitic Syndrome = False
2. Sat 2001-03-13: SCORE = -0.00000464 PVALUE = 0.00000000
   12.42% (58/467) of today's cases have Respiratory Syndrome = True
   6.53% (653/10000) of baseline have Respiratory Syndrome = True
3. Wed 2001-06-30: SCORE = -0.00000013 PVALUE = 0.00000000
   1.44% (9/625) of today's cases have 100 <= Age < 110
   0.08% (8/10000) of baseline have 100 <= Age < 110
4. Sun 2001-08-08: SCORE = -0.00000007 PVALUE = 0.00000000
   83.80% (481/574) of today's cases have Unknown Syndrome = False
   74.29% (7430/10001) of baseline have Unknown Syndrome = False
5. Thu 2001-12-02: SCORE = -0.00000087 PVALUE = 0.00000000
   14.71% (70/476) of today's cases have Viral Syndrome = True
   and Encephalitic Syndrome = False
   7.89% (789/9999) of baseline have Viral Syndrome = True
   and Encephalitic Syndrome = False
6. Thu 2001-12-09: SCORE = -0.00000000 PVALUE = 0.00000000
   8.58% (38/443) of today's cases have Hospital ID = 1
   and Viral Syndrome = True
   2.40% (240/10000) of baseline have Hospital ID = 1
   and Viral Syndrome = True

Rule 3 is 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. For Rules 1, 2, 5 and 6, we went back to the original ED database to look at the text description of the chief complaints for the cases related to these three rules. Rule 2 cases contain a large number of complaints of shortness of breath, possibly due to an illness causing respiratory problems. The symptoms related to Rules 1, 5 and 6 involve dizziness, fever and sore throat. Given that Rules 1, 5 and 6 have dates in winter, along with the symptoms mentioned, we speculate that this anomalous pattern is likely caused by an influenza strain.

5. Conclusion

Even with multiple periodic trends and other non-anthrax illnesses present in the simulated data, WSARE 3.0 has been shown to be successful at detecting anomalous patterns that are indicative of an anthrax release. WSARE 3.0 outperformed all the other algorithms evaluated by detecting the anthrax releases with close to a one day delay, which is the earliest possible detection time in our simulation. For a false positive rate of one per month, WSARE 3.0 detects the simulated anthrax release about 2 days earlier than WSARE 2.0 and 12 hours earlier than WSARE 2.5. In addition, the false positive rate for WSARE 3.0 could be reduced even further if more environmental attributes capturing the current state of the system had been added to the Bayesian network. We have also demonstrated that WSARE 3.0 is able to find anomalous patterns in real ED data.

6. Related Work

Market basket analysis (Agrawal & Srikant, 1994; Brin et al., 1997) uses association rules to find patterns in sales data. Contrast set mining (Bay & Pazzani, 1999) has the same flavor as the approach taken by WSARE except it finds rules with more than two components 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-Squared test. Many of these rules are interesting to WSARE. Multiple hypothesis testing problems are addressed using 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, with the main difference being the additional steps of the randomization test and FDR in WSARE. Kuldorff's Spatial Scan Statistic (Kulldorff, 1997) finds clusters in a multi-dimensional point process that are not explained by a baseline distribution. Both of these detection algorithms do not take periodic trends into account. In (Goldenberg et al., 2002), the authors investigate the use of grocery data for the early detection of bio-terrorism attacks. Periodic trends in the grocery data are handled by using a wavelet transform while an autoregressive model is used for prediction of next day sales.

7. Acknowledgements

This research is supported by DARPA grant F30602-01-2-0550. Thanks to Rich Tsui, Bob Olszewski, Jeremy Espino, Jeff Schneider and the anonymous reviewers for helpful comments and suggestions.

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules. *Proceedings of the 20th VLDB Conference, Santiago, Chile* (pp. 487–499). Morgan Kaufmann.

Bay, S. D., & Pazzani, M. J. (1999). Detecting change in categorical data: Mining contrast sets. *Knowledge Discovery and Data Mining* (pp. 302–306).

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

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

Brin, S., Motwani, R., Ullman, J. D., & Tsur, S. (1997). Dynamic itemset counting and implication rules for market basket data. *SIGMOD 1997, Proceedings ACM SIGMOD International Conference on Management of Data, May 13-15, 1997, Tucson, Arizona, USA* (pp. 255–264). ACM Press.

Brossette, S. E., Sprague, A. P., Hardin, J. M., Waites, K. B., Jones, W. T., & Moser, S. A. (1998). Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association, 5*, 373–381.

Goldenberg, A., Shmueli, G., Caruana, R. A., & Fienberg, S. E. (2002). Early statistical detection of anthrax outbreaks by tracking over-the-couner medication sales. *Proceedings of the National Academy of Sciences* (pp. 5237–5249).

Good, P. (2000). *Permutation tests - a practical guide to resampling methods for testing hypotheses.* New York: Springer-Verlag. 2nd edition.

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

Miller, C. J., Genovese, C., Nichol, R. C., Wasserman, L., Connolly, A., Reichart, D., Hopkins, A., Schneider, J., & Moore, A. (2001). *Controlling the false discovery rate in astrophysical data analysis* (Technical Report). Carnegie Mellon University.

Moore, A., & Lee, M. S. (1998). Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research, 8*, 67–91.

Moore, A., & Wong, W.-K. (2003). Optimal reinsertion: A new search operator for accelerated and more accurate Bayesian network structure learning. *Proceedings of ICML 2003.*

Wong, W.-K., Moore, A. W., Cooper, G., & Wagner, M. (2002). Rule-based anomaly pattern detection for detecting disease outbreaks. *Proceedings of AAAI-02* (pp. 217–223). MIT Press.