Data Mining for Early Disease Outbreak Detection

Weng-Keen Wong

January 2004

CMU-CS-04-125

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Thesis Committee:
Andrew Moore,
Scott Fahlman,
Jeff Schneider,
Gregory Cooper, University of Pittsburgh
Michael Wagner, University of Pittsburgh

This research is supported by DARPA grant F30602-01-2-0550 and by the State of Pennsylvania.

The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the sponsors.

Keywords: Anomaly Detection, Syndromic Surveillance, Biosurveillance, Bayesian Networks

For my mother who encouraged me to be the first one in her family to get a PhD.

“...much learning doth make thee mad.” - Acts 26:24

Contents

List of Figures

List of Tables

Abstract

This thesis presents an early disease outbreak detection algorithm called What's Strange About Recent Events (WSARE). Unlike traditional disease outbreak detection algorithms that look for peaks in a univariate time series of health-care data, WSARE tries to improve its timeliness of detection by taking a novel multivariate approach. Current health-care data used for surveillance are no longer simply a time series of aggregate daily counts. Instead, a wealth of spatial, temporal, demographic, and symptom information is available. WSARE incorporates all of this information using a rule-based approach that compares recent health-care data against data from a baseline distribution and finds subgroups of the data whose proportions have changed the most in the recent 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 powerful detection algorithm that is careful to evaluate the significance of the alarms that it raises.

Acknowledgments

I would like to thank the following people:

- My advisor Andrew Moore for his guidance, expertise and for being one of the nicest advisors a graduate student could have.
- My thesis committee members Greg Cooper, Scott Fahlman, Jeff Schneider and Mike Wagner for all their constructive comments and technical expertise
- My parents and brother for their support
- My fellow AUTON Lab members, particularly Pat Gunn, Paul Komarek, Jacob Joseph and Dan Pelleg for their technical support and also Daniel Neill, Alex Gray, Anya Goldenberg, Robin Sabhnani and Ajit Singh
- RODS Lab members Rich Tsui, Jeremy Espino, Oleg Ivanov, Bill Hogan, Bob Olszewski, Wendy Chapman and Feng Dong
- My officemates Paul Bennett, Stavros Harizopoulous, and Francisco Pereira
- A special thank you to the terrific staff in CSD, especially Sharon Burks, Jean Harpley, Karen Olack and Catherine Copetas
- Others that I have consulted with: Howard Burkom, Dave Buckeridge, Geoff Gordon and Larry Wasserman
- Maria Klawe who convinced me to do a PhD
- My friends from Mckeesport Gospel Hall who made my time here quite memorable

Chapter 1

Introduction

Disease outbreak detection algorithms have historically been used for retrospective analysis (Farrington & Beale, 1998; Sonesson & Bock, 2003), in which the identification of the outbreak occurs after the outbreak has already taken place. In retrospective analysis, the aim is to identify outbreak periods in a data set that has been accumulated over some historical time period. Algorithms performing this type of analysis operate on either mortality data associated with some disease (Serfling, 1963; Choi & Thacker, 1981) or lab report data containing counts of specimens of various microorganisms received (Stern & Lightfoot, 1999). Both of these data sources are specific to the disease in the sense that the disease has already been accurately diagnosed when the data are compiled. In addition, these data sources take the form of either a univariate time series, with the single dimension being aggregated counts over some time interval (Serfling, 1963; Choi & Thacker, 1981), or spatial data accumulated over some time interval (Marshall, 1991; Lawson, 2001). The main criteria for evaluation of these retrospective algorithms is sensitivity, which is related to the detection capability of the algorithms and specificity, which is related to the false positive rate.

In contrast to retrospective analysis, prospective surveillance requires monitoring of health-care cases in order to detect a disease outbreak as soon as possible. The data used in prospective surveillance do not come from a fixed historical time window but from a stream of data containing the most recent events. There are obvious benefits to the early detection of an epidemic as lives are saved and costs are reduced. However, in order to achieve timely detection, there needs to be a tradeoff between specificity and timeliness (Wagner et al., 2001a). Data sources that require diagnosis of the disease, such as lab reports, can only be obtained several weeks after the samples are submitted. By that point, the outbreak may have escalated into a large scale epidemic. Instead of waiting for post-diagnosis data, we can monitor pre-diagnosis data, which is often non-specific, such as the symptoms exhibited by patients. In doing so, we risk increasing the false positive rate, such as mistakenly attributing an increase in cases involving respiratory problems to an anthrax attack rather than influenza, but we also have a potential gain in timeliness. This type of pre-diagnosis surveillance is commonly referred to as syndromic surveillance (Mostashari & Hartman, 2003; Sosin, 2003). The simplest form of syndromic surveillance involves sentinel physicians (Feagin, 1971; Fitzner et al., 1999; Redondo et al., 2002; van Casteren & Leurquin, 1992) noticing an increase in the number of patients exhibiting certain symptoms and then alerting the appropriate public health institutions. Naturally, this process results in spotty coverage, a large possibility of missing an outbreak, and likely late detection times. Several automated detection algorithms were subsequently developed although these algorithms were fairly simple and operated on univariate time series data (See Chapter 3 for a full review). Furthermore, these algorithms were evaluated on specificity and sensitivity, without any consideration of the timeliness of detection.

Events in recent years have made timeliness the focus of new outbreak detection algorithms (Wagner et al., 2001b). After the October 2001 attacks, threats of further terrorist activity through biochemical weapons made the early detection of disease outbreaks an even more urgent issue. Moreover, the worldwide spread of the SARS outbreak highlighted the importance of timely detection and containment of an epidemic. However, improving the timeliness of an algorithm is difficult because outbreaks have weak, non-specific signals at their outset, resulting in a tradeoff between timeliness and specificity for detection algorithms as was mentioned above.

Strategies for improving timeliness without sacrificing too much specificity have been aided by recent developments in technology and law. First of all, as health-care records are becoming electronic, the quantity of data available for surveillance has increased and the quality has improved as well. Secondly, legal guidelines for the use of health-care records in surveillance and research have been established through the Health Insurance Portability and Accountability Act. These guidelines are designed to standardize electronic transfer of health information and also to protect patient confidentiality. We will discuss the privacy issues in more detail in Chapter 2.

Two main approaches have been proposed for improving timeliness while still maintaining reasonable specificity. The first strategy involves taking advantage of all the attributes in the available data. Traditional detection methods simply rely on detecting peaks in a univariate time series formed from aggregated daily counts. Health-care records are now highly multivariate, containing time information in the form of the case date and time, spatial information in the form of home, work, and hospital locations, demographic information about patients, and symptom information. Solely relying on aggregated daily counts would ignore almost all of these useful attributes. By considering time, space, demographic and symptom information, (Wagner et al., 2001a) suggest that we can improve the timeliness of detection. This approach certainly makes sense for localized outbreaks such as food-borne illnesses or an aerosolized anthrax release since the incorporation of spatial information would help the detection. For large scale epidemics, this approach could detect the weaker initial signals. Furthermore, suppose we know that a disease has a distinct fingerprint in time, space, demographics and patient symptoms. Then finding patterns in non-specific data which match this fingerprint would support our belief that an outbreak of this specific disease is occurring.

The second strategy is to make use of a variety of available data sources, both traditional and non-traditional. As an outbreak starts in a region, it will, over time, leave behind a trail in a variety of databases. For example, signals could be seen in Emergency Department cases, over-the-counter sales, absenteeism records for school and work, veterinarian data, lab culture reports, poison control centers, and even mortality rates. All of these data sources have some form of time lag between when the outbreak takes place and when the signal actually appears in the data. By monitoring all of these sources, (Wagner et al., 2001a) speculate that the additional signals could also improve timeliness.

In this thesis, we will deal with the first strategy of improving timeliness by monitoring the full range of attributes in health-care data. The second approach will not be discussed. Combining multiple data sources is still very much an open question for the area of syndromic surveillance. The focus of this thesis will be the What's Strange About Recent Events (WSARE) algorithm. WSARE 2.0 is a multivariate outbreak detection algorithm that finds differences between a baseline and a recent data set through significant changes in the proportions of subgroups in the data. This multivariate approach is completely different from the traditional univariate biosurveillance algorithms which monitor a time series. WSARE is a non-specific detector that is intended to play the role of a safety net when used with a suite of specific disease detectors. WSARE 3.0 extends the previous version by accounting for temporal fluctuations in the data, such as seasonal and weekend / weekday effects, which is a challenging aspect of health-care data that must be dealt with by detection algorithms.

The thesis hypothesis is as follows: New algorithms that combine association rule search and Bayesian network distribution learning will perform better than traditional univariate biosurveillance algorithms in terms of timeliness, sensitivity, and specificity in most cases of disease outbreaks, as evaluated using various outbreak simulators.

This thesis will continue with a description of the privacy issues in syndromic surveillance in Chapter 2. Chapter 3 contains a review of biosurveillance algorithms in order to illustrate how WSARE relates to these other algorithms. In Chapter 4, the association rule approach of WSARE 2.0 for detecting anomalous patterns will be described while the subsequent chapter will describe how WSARE 3.0 uses a Bayesian network to model a changing baseline distribution. Chapter 6 will include performance results based on data from a gridworld simulator, a Bayesian network simulator and real world data from the DARPA challenge. We will also include the output of running WSARE on actual ED data. Additionally, Chapter 6 includes timing results for various computational optimizations we implemented in WSARE. Improvements to the randomization test step of WSARE and ways to deal with multiple hypothesis testing will be discussed in Chapter 7. Chapter 8 includes an efficient algorithm for clustering with clutter, which we plan to modify for spatial biosurveillance. Finally, Chapter 9 will contain future work and a conclusion to this thesis.

Chapter 2

Privacy Issues

Before proceeding with the thesis, we will briefly discuss some of the privacy issues that have a direct effect on all algorithms designed for early detection of disease outbreaks. The Health Insurance Portability and Accountability Act (The Health Insurance Portability and Accountability Act, 1996), enacted in August 21, 1996, was aimed at standardizing the exchange of electronic health information between health providers and insurers. Congress then established the HIPAA Privacy Rule (45 CFR Parts 160 through 164, 2003) in August 14, 2002 in order to protect the privacy and security of health information. The HIPAA Privacy Rule presents strict guidelines for the use and disclosure of protected health information for research purposes. Violations of these regulations can result in civil or criminal penalties.

One of the aims of the HIPAA Privacy Rule when applied to research is to establish a balance between protecting the privacy of individually identifiable health information while allowing researchers access to information essential to their research. The privacy rule is an extensive document and we will only briefly mention the consequences of these regulations on our research. Readers interested in the full details are referred to (45 CFR Parts 160 through 164, 2003).

Due to the high volume of medical records required for surveillance, individual research participant authorization is impractical. The HIPAA Privacy Rule permits the use of protected health information without direct authorization by the patient through several options. Two of these options include HIPAA Safe Harbor de-identification or the provision of a Limited Data Set (Research and Practice Fundamentals, 2003). In HIPAA Safe Harbor de-identification, the information is completely de-identified through removal of the following 18 identifiers: names, all geographic subdivisions smaller than a state (except for the first three digits of a zipcode under specific conditions), all dates that are directly connected to an individual, telephone numbers, fax numbers, email addresses, social security numbers, medical record numbers, health plan beneficiary numbers, account numbers, certificate/license numbers, vehicle identifiers, device identifiers, URLs, IP address numbers, biometric identifiers, full face photos, and any other unique identifying numbers. The zipcode restriction is overly strict for most surveillance algorithms that have some spatial aspect. As a result, we use a Limited Data Set in which all of the Safe Harbor identifiers are removed except for city, state, zipcode, and dates. Use of a Limited Data Set requires a Data Use Agreement between the covered entity and the researcher. This agreement specifies the uses of this information, the individuals with access to this information, and a guarantee from the researchers that measures will be taken to protect the privacy of this information.

These legal regulations have a direct effect on surveillance algorithms, with the most significant result being that surveillance algorithms must operate on data with some fields that are not as precise as they could be. As an example, spatial information can contain either the exact locations of individuals or aggregate counts over an administrative region such as a census tract or zipcode. Under the HIPAA Privacy Rules, home and work locations in medical data are frequently given as zipcodes. Spatial surveillance algorithms need to be designed to operate on aggregate counts rather than on individual case events. Naturally, aggregation over zipcodes loses some degree of precision since the zipcode centroids are reported as the locations of the cases. Using the zipcode centroid coordinates can be problematic for irregularly shaped zipcode regions. In addition, zipcodes are administrative regions without any statistically meaningful boundaries. Thus, in order to be practical, surveillance algorithms must be able to cope with these issues.

Chapter 3

Related Work

Outbreak detection systems operate by detecting anomalies in surveillance data. An anomaly, which is also called an aberration in (Janes et al., 2000), is described as a statistically significant change in the occurrence of health events when compared with normal history. The process of detecting anomalies in health-care data can be summarized in two steps. First of all, a baseline that captures the usual pattern of a disease needs to be obtained. Secondly, using this baseline, a threshold must be calculated such that the presence of an anomaly is detected once this threshold is exceeded. In general, all outbreak detection algorithms follow this two-step procedure, with variations in the first step accounting for the majority of their differences.

Diseases can be characterized by the three basic epidemiological parameters of time, space, and person (Janes et al., 2000), where the "person" parameter corresponds to an individual's demographic information. Historically, the majority of outbreak detection algorithms have been developed to look for anomalies in time or for anomalies in space. Recently, timeliness of detection has become a new public health requirement due to emerging diseases and the threat of bioterrorist attacks (Wagner et al., 2001b). One strategy for improving timeliness is to exploit the spatial, temporal, and demographic aspects of health-care events (Wagner et al., 2001b). With many new sources of health-care data becoming available, along with existing data sources being augmented with such information, a variety of new surveillance algorithms using this strategy have been designed. This section will consist of a review of algorithms used in detection of anomalies over four categories: time, space, space-time, and space-time-person. Space-person algorithms are not very common in surveillance because most approaches simply stratify the data and use a purely spatial technique on each strata to search for anomalies. Other reviews have been written on surveillance algorithms (Farrington & Beale, 1998; Farrington & Andrews, 2004; Sonesson & Bock, 2003) that deal primarily with surveillance algorithms that detect anomalies in time, with much smaller sections dealing with space and space-time algorithms. The survey in this chapter, on the other hand, will have a much wider scope. Furthermore, this chapter will include several recent algorithms that are not included in the previously mentioned reviews. The focus of this survey will be on the algorithms and their characteristics. The scope of this review will be limited to existing algorithms used for the surveillance of diseases. We will mainly deal with algorithms for prospective surveillance although a few methods which have been used in retrospective analysis will also be discussed.

3.1 Time Surveillance

Algorithms detecting anomalies in time operate on aggregated data to find unusual spikes in time. We have chosen to discuss specific characteristics of the algorithms discussed in this section in order to highlight their differences. These characteristics are summarized in Table 3.1 and are described in detail below.

1. Implementation Complexity
   Implementation complexity describes how difficult a programmer's job would be to write the algorithm. Since numerous software packages are available that make statistical functions readily available to programmers, this column will describe the complexity of implementing the algorithm using available software packages. The name of some available software packages providing the necessary statistical functions for the particular detection algorithm is shown in brackets.

2. Computational Complexity
   Computational complexity is a measure of how computationally intensive the detection algorithm is. In general, the more computationally intense the algorithm, the longer its running time will be. Rather than giving a precise measurement of complexity, we will describe this attribute using the values of low, moderate or high. These values are intended to be relative to each other so that even though an algorithm may be efficient, it will still get a rating of high if it is much more complex than the simplest algorithm we review.

3. Baseline Smoothing

   Health-care data with few or no epidemics is needed to establish a baseline distribution. However, finding health-care data that contain no epidemic periods is a difficult task. Some detection algorithms are able to handle the outliers in the data due to epidemics by smoothing the baseline distribution in various ways. Included in this category is the ability of the algorithm to handle negative singularities, which are informally defined as values that are much lower than neighboring values in the time series (Zhang et al., 2003). Negative singularities are common in health-care data due to public holidays and extreme weather conditions.

4. Cyclical components (Cyc)

   A check mark in this column indicates that the algorithm can handle cyclical fluctuations in the data such as those due to seasonality or day of week effects. In general, cyclical components can be dealt with in two ways. Some methods explicitly include the cyclical components in the model. Others techniques simply restrict the baseline to include only data from a similar time period.

5. Long Term Trends (LTT)

   A signal containing a long term trend will gradually increase or decrease over an extended period of time. Algorithms that do not account for long term trends will have a large number of false alarms as the trend dominates data from the start of the surveillance period.

6. Recent Trends (RT)

   If the data displays a recent upward or downward trend, should the next day be expected to follow that increase or decrease? This column will have a check mark if the algorithm accounts for such recent trends. It is somewhat unclear whether accounting for recent trends benefits an algorithm since recent trends can also easily mislead the forecasts of the algorithm.

7. Parameter Tuning

   This column describes the difficulty in tuning an algorithm to perform well on data. For instance, an algorithm with many parameters whose values depend on the data and require hand-tuning will be rated as High. On the other hand, an algorithm with few parameters or one which autonomously assigns values to these parameters will be considered to be Low.

8. Forecasting
   Some algorithms are able to predict the expected value for the current day's signal while others simply return an alarm threshold for the current day. If an algorithm does perform forecasting, then a check mark is entered into this column.

9. Example Data
   This category lists examples of the types of data that the algorithm has been applied to.

3.1.1 Clustering Techniques

The algorithms in this section look for unusual clusters of health-care cases by monitoring the temporal distance between cases. These techniques include the Scan Statistic, Tango's Clustering Index and Chen's time interval technique.

Scan Statistic

The scan statistic was originally developed by (Naus, 1965). It is most easily described in the discrete form and we will use the notation of (Glaz & Balakrishnan, 1999). Suppose we have a sequence of N integer valued random variables. By moving a window of fixed size m along the sequence, where $2 \leq m \leq N-1$, we can count the sum of the observations that fall in that window as $Y_t = \Sigma_{i=t}^{t+m-1}$ as shown in Figure 3.1. Let S_m, the discrete scan statistic, be the maximum sum found in the window as shown in equation 3.1:

$$S_m = max_{1 \leq t \leq N-m+1} Y_t \tag{3.1}$$

We can then set up a likelihood ratio test to detect a local unusual cluster among the N observations. Under the null hypothesis, the sequence of N random variables are all drawn from the same distribution F_0. The alternative hypothesis we test for states that the observations within the window have some other distribution F_1 while the observations outside the window are still distributed as F_0. Using the S_m statistic, we can decide whether or not to reject the null hypothesis.

Algorithm	**Implementation Complexity**	**Computational Complexity**	**Baseline Smoothing**	**Cyc**	**LTT**	**RT**	**Parameter Tuning**	**Forecasts**	**Example Data**
Scan Statistic	Moderate	Moderate	None	×	×	×	Low	×	NNDSS records
Tango's Clustering Index	Moderate	Moderate	None	×	×	×	Low	×	Trisomy data
Chen's Interval Sets	Low	Low	None	×	×	×	Low	×	Hodgkins disease data
Shewhart Chart	Low	Low	None	×	×	×	Low	×	NNDSS records
Historical Limits (Median)	Low	Low	Minimal	×	×	×	Low	×	NNDSS records
CUSUM	Low	Low	None	×	×	×	Low	×	Malformations, influenza, salmonella
EWMA	Low	Low	None	×	×	×	Low	×	NNDSS records
Serfling	Low (Any Stats package)	Low	None	√	√	×	Low	√	P&I deaths, P&I ICD-9 codes
ANOVA Regression	Low (Any Stats Package)	Low	None	√	√	√	Low	√	ED data
Log Linear Regression	Moderate (Any Stats package)	Moderate	Yes	√	×	×	High	√	Lab reports
GLMM	Low (SAS)	High	None	√	√	√	Low	√	Ambulatory care encounter records
Compound Smoothing	Low (Minitab)	Moderate	Yes	×	×	×	Low	×	Salmonella and Shigella lab reports
AR and MA	Low	Low	None	×	×	√	Low	√	ED data
ARIMA	Low (Matlab)	Moderate	None	√	√	√	Low	√	P&I deaths, Salmonella reports, NEDSS records, Pediatric ED records
Nobre et al. Change-point Detection	Moderate (Matlab)	Moderate	Minimal	√	√	×	High	√	NEDSS measles reports
Baron Change-point Detection	High	High	Perhaps possible through prior	√	×	×	Low	×	Influenza deaths
MWAR	High (Matlab)	High	Low	√	√	√	Moderate	√	Grocery sales data
WAD	Low (Matlab)	Moderate	High	√	√	×	Low	√	ED data

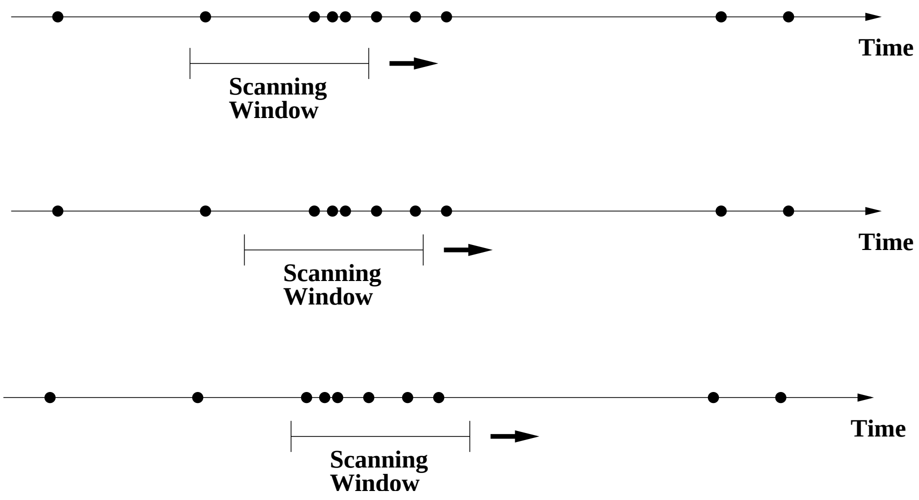


Figure 3.1: An illustration of the Scan Statistic

The key to the hypothesis test is determining $P(S_m \geq k)$, where k is a constant that controls the Type I error. The exact probability has been calculated for specific distributions for a certain range of parameters. In the majority of cases, however, accurate approximations have been developed (Glaz & Balakrishnan, 1999).

The scan statistic also has a continuous version where a point process is observed along an interval $(0, T]$ and the scan statistic looks for the maximum number of events that occur in a moving window $(t, t + w)$ of fixed size w, where $0 < w < T$. Like the discrete case, $P(S_w \geq k)$ is a necessary component of the hypothesis test and this probability is usually computed using approximations (Glaz & Balakrishnan, 1999).

Tango's Clustering Index

Tango (Tango, 1984) developed a clustering index as a test for temporal and cyclical clustering. This clustering index is designed to be reliable when the total number of counts is small. Suppose we have counts $N_1, \ldots, N_m$ over m consecutive equal length time intervals. If we normalize the counts by $N = \Sigma_{i=1}^{m} N_i$, we can obtain a vector $r = (r_1, \ldots, r_m)$ of relative frequencies. The clustering index C is described as the quadratic form:

$$C = r'Ar \tag{3.2}$$

In equation 3.2, the elements a_{ij} of the A matrix have the property that $a_{ii} = 1$ and a_{ij} is a monotonically non-increasing function of the temporal distance between the ith and jth interval eg. $d_{ij} = |i - j|$. Tango uses the formula $a_{ij} = exp(-d_{ij})$. The clustering index C takes the maximum value 1 whenever the vector r has a 1 in one position and zeros everywhere else.

Under the null hypothesis, there is no clustering in time and (Tango, 1984) derives the distribution of the clustering index under the null as:

$$Pr(C \leq c) = \sum_{i=0}^{\infty} \alpha_i Pr\{\chi^2_{m-1+2i} < (c - h)\beta^{-1}\} \tag{3.3}$$

The formulae for α_i, β, and h will not be given here. The important point is that although equation 3.3 requires an infinite sum, a finite number of terms is needed for the sum to convergence up to an ϵ. Of course, the more terms are needed, the longer the calculation takes. The p-value of the clustering index can be obtained by $1 - Pr(C \leq c)$.

Tango's clustering index was evaluated on trisomy data in (Tango, 1984), along with a chi square test and a scan statistic. Whenever the clustering appears in the middle of the sequence of counts, the clustering index is more powerful than the other two techniques. The author explains that this is expected because the chi square test does not take the temporal sequence into account while the scan statistic only looks at the maximum clustering frequency and may ignore other indications of clustering. However, when clustering happens toward one end of the sequence of events, the clustering index may not be as powerful as the other algorithms.

Chen's time interval technique

Chen's algorithm detects increases in rates of chronic diseases by measuring the time interval between two diagnoses. This algorithm is ideally suited for rarely occurring diseases. In this algorithm's framework, suppose the baseline disease rate is π_0 and the total population in region i is N_i. Furthermore, assume that the number of new diagnoses over some time unit in geographic region i is Poisson distributed with mean $\lambda_{i,0} = N_i\pi_0$. If the random variable W is the time interval between diagnoses, then W is exponentially distributed with mean $\lambda_{i,0}$.

The quantity we are interested in is the probability that an interval w is less than W under baseline conditions and under epidemic conditions. Expressing W as a multiple k of the expected interval $E(w) = \frac{1}{\lambda_{i,0}}$, we get $Pr(w < \frac{k}{\lambda_{i,0}}) = 1 - exp(-k)$ for the baseline disease rate. Now suppose an epidemic occurs in the region and the baseline disease rate is scaled by γ ie. $\pi_1 = \gamma\pi_0$. Under these conditions, $Pr(w < \frac{k}{\lambda_{i,0}}) = 1 - exp(-k\gamma)$

For small scale surveillance, whenever a new case is diagnosed, the last n intervals will be analyzed using this algorithm. The increase is considered significant if the last n intervals in the sequence are all shorter than a threshold value, which is expressed as $kE(w)$. The values k, and n can be determined using formulas given the value of γ and an estimate M of the number of cases over the entire surveillance period. The authors note the similarity between this method and CUSUM, which will be described in Section 3.1.2.

For large scale surveillance over f geographic regions, the authors recommend not fixing k in advance. Instead, the formula $k = -ln(1 - P_0(n_i')^{\frac{1}{n_i'}})$ is used where n_i' is the observed number of cases in geographic region i and $P_0(n_i')$ is the fixed probability of the false alarm, which is determined in advance and usually set to 0.05. An overall alarm is raised when the number of regions with alarms being raised is significant assuming a binomial distribution with parameters P_0 and f for the number of geographic regions.

3.1.2 Statistical Quality Control

In statistical quality control, some statistic of a production process is monitored to determine if the process is in control. Since quality control algorithms essentially perform surveillance on a signal, many of these techniques have been transplanted to the field of biosurveillance.

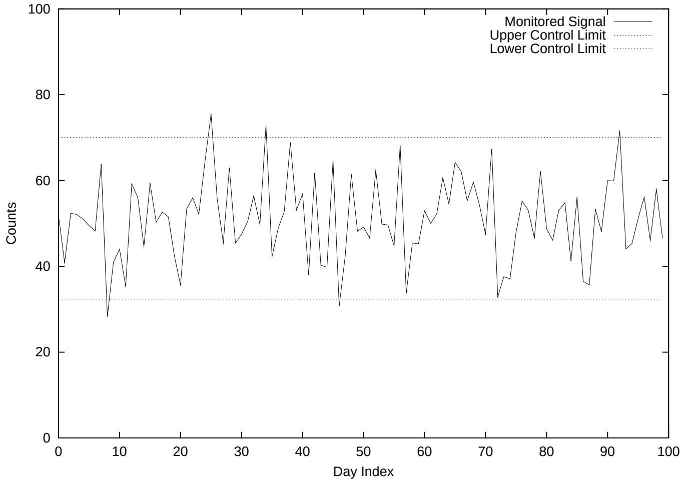


Figure 3.2: An example of a Shewhart Control Chart on a set of observations drawn from a normal distribution with mean 50 and standard deviation 10

Shewhart Control Charts

Shewhart control charts, which are also simply called control charts, are used to monitor some quality characteristic of a production process. As described by (Banks, 1989), a control chart consists of a center line (CL) which is between an upper control limit UCL) and a lower control limit (LCL). The quality characteristic is plotted on the chart. If any points fall outside either the upper or the lower control limits, the process is considered to be out of control. There are many variations on control charts but they have the basic form below (Banks, 1989):

$$\begin{aligned} UCL &= E(C) + k\sqrt{Var(C)} \\ CL &= E(C) \\ LCL &= E(C) - k\sqrt{Var(C)} \end{aligned} \tag{3.4}$$

In Equation 3.4, C is the quality characteristic, $E(C)$ is the expected value of C and $Var(C)$ is the variance of C. The constant k is typically set to 3 in most control charts. Whenever $E(C)$ and $Var(C)$ are not known, they need to be estimated from the data. Figure 3.2 illustrates a control chart on a set of observations drawn from a normal distribution with mean 50 and standard deviation 10. The control limits are obtained by estimating the mean and standard deviation from the 100 observations.

The upper and lower control limits are set at two standard deviations above and below the mean respectively.

Historical Limits

The historical limits algorithm (Stroup et al., 1989), used for detecting anomalies in notifiable diseases surveillance data, is similar to a control chart. This algorithm operates on disease reports aggregated over a four week period called a "month", ending with the current week under investigation. In order to remove seasonal fluctuations, a baseline distribution is formed from 15 months worth of data, obtained by considering the current month and its two adjacents months in a five year window of historical data. For example, if the current month is November of the year 2003, then the baseline distribution is obtained by taking data from October, November and December of the years 1998 to 2002 and data from October 2003. This algorithm uses the ratio of the current month's value divided by the mean of the 15 baseline months. If this ratio falls outside the historical limits, then an anomaly is identified. The historical limits are shown in Equations 3.5 and 3.6 as described in (Hutwagner et al., 2003). In the equations below, μ is the mean of the 15 baseline values, σ_x is their standard deviation, and x_0 is the current month's value.

$$\frac{x_0}{\mu} > 1 + 2 * \frac{\sigma_x}{\mu} \tag{3.5}$$

$$\frac{x_0}{\mu} < 1 - 2 * \frac{\sigma_x}{\mu} \tag{3.6}$$

The authors point out that using the mean makes this algorithm sensitive to outliers in the baseline. They suggest a variation on the algorithm which uses the median and is more robust to extreme values. However, by using the median, the confidence intervals need to be obtained through bootstrapping. The 15 baseline values are also too small a sample for the asymptotic bounds of the bootstrap to hold and a larger data set needs to be used.

CUSUM

Another technique used for biosurveillance that has its origin in statistical quality control is the cumulative sum (CUSUM) method. CUSUM has been extensively used in biosurveillance systems, including surveillance of malformations in children (Weatherall & Haskey, 1976), influenza surveillance (Tillett & Spencer, 1982), detection of salmonella outbreaks (Hutwagner et al., 1997) and in the Early Aberration Reporting System (Hutwagner et al., 2003). CUSUM was originally developed to detect changes in the quality of the output of a continuous production process (Page, 1954). While a Shewhart control chart can be used for this purpose, they are intended to detect a large shift from the mean. CUSUM control charts are more sensitive to small shifts from the mean and are able to detect them more quickly than a Shewhart control chart.

As its name suggests, CUSUM maintains a cumulative sum of deviations from a reference value r, such as the mean. Suppose we have a time series where at time i, we have measurement X_i. Furthermore, assume that X_i has a normal distribution with mean μ_0 and variance σ^2. The CUSUM calculation S_i is as follows:

$$\begin{aligned} S_1 &= X_1 - r \\ S_2 &= (X_2 - r) + (X_1 - r) = (X_2 - r) + S_1 \\ &\vdots \\ S_m &= \sum_{i=1}^{m}(X_i - r) = (X_m - r) + S_{m-1} \end{aligned}$$

From the equation above, if the X_m values are close to the mean, then the S_m values will be around 0. However, once a shift from the mean occurs, the S_m values will either increase or decrease quickly. CUSUM charts used in biosurveillance are usually only concerned with a positive shift from the mean. As a result, this procedure is a one-sided test and we can rewrite S_m as:

$$S_m = max(0, (X_m - r) + S_{m-1}) \tag{3.7}$$

In the equations above, we have used the mean as the reference value r. Typically the reference value consists of the mean plus or minus a slack value or allowance, which is called K. We can rewrite Equation 3.7 as:

$$S_m = max(0, X_m - (\mu_0 + K) + S_{m-1}) \tag{3.8}$$

In Equation 3.8, any values within K units of μ_0 are effectively ignored. The allowance K is usually set to be the midpoint between the in-control process mean μ_0 and the out-of-control process mean μ_1, expressed in terms of the standard deviation σ as shown in Equation 3.9. The value of K is chosen to reflect the magnitude of the shift to be detected, defined as a multiple of σ.

$$\begin{aligned} K &= \frac{\delta}{2}\sigma = \frac{|\mu_1 - \mu_0|}{2} \\ \text{where } \delta &= \frac{|\mu_1 - \mu_0|}{\sigma} \end{aligned} \tag{3.9}$$

Alerts are raised whenever S_m exceeds a threshold or decision interval, which is called H in the literature. The value of H is determined from the value of K and the Average Run Length (ARL) of the in-control process. The ARL is the average number of timesteps i before an alert is raised. Whenever the process is operating as expected, the ARL should be large. However, when a shift in the mean occurs, the ARL should be small in order to detect the shift as quickly as possible. The ARL can be determined through a variety of approximating equations, which are summarized in (Montgomery, 2001). In general, a reasonable value for H is 5σ according to (Montgomery, 2001).

EWMA

Another statistical quality control technique that is similar to CUSUM is the Exponentially Weighted Moving Average (EWMA) algorithm (Roberts, 1959). EWMA is better at detecting small shifts than a Shewhart control chart, with its performance being similar to that of the CUSUM algorithm although it is easier to set up and operate (Montgomery, 2001). Suppose we have a measurement X_i at time i. Then the EWMA statistic Z_i is:

$$Z_i = \lambda X_i + (1 - \lambda) Z_{i-1} \text{ where } 0 < \lambda \leq 1 \tag{3.10}$$

The starting value Z_0 is set to the desired process mean μ_0 or to the average of some initial data ie. $Z_0 = \bar{X}$. If we unravel the recursion in 3.10 as shown in 3.11, we can see that the EWMA algorithm weights all previous samples, with the weights summing to one.

$$\begin{aligned} Z_i &= \lambda X_i + (1-\lambda)[\lambda X_{i-1} + (1-\lambda)Z_{i-2}] \\ &= \lambda X_i + \lambda(1-\lambda)X_{i-1} + (1-\lambda)^2 Z_{i-2} \\ &\vdots \\ &= \lambda\sum_{j=0}^{i-1}(1-\lambda)^j X_{i-j} + (1-\lambda)^i Z_0 \end{aligned}$$

Assuming that the X_is are independent random variables with variance σ^2, the variance of Z_i can be computed as:

$$\sigma_{z_i}^2 = \sigma^2\left(\frac{\lambda}{2-\lambda}\right)[1-(1-\lambda)^{2i}] \tag{3.11}$$

With this variance, we can establish control limits for an EWMA chart as shown below.

$$\begin{aligned} UCL &= \mu_0 + L\sigma_{z_i} \\ LCL &= \mu_0 - L\sigma_{z_i} \end{aligned}$$

Choices of the L and λ parameter can be made based on the desired ARL. (Lucas & Saccucci, 1990) contains ARL tables for choices of L and λ. In practice, values of $0.05 \leq \lambda \leq 0.25$ and $L = 3$ work well. According to (Montgomery, 2001), EWMA can be considered as a non-parametric monitoring procedure since it makes no assumptions about the distribution that X_i is drawn from. EWMA has been used for surveillance on NNDSS data (Williamson & Hudson, 1999).

3.1.3 Regression

A natural approach to monitoring a time series would be to try to fit a model to the series using regression and then forecasting the current day's value. If the observed value is significantly different from the forecast, then an alert is raised. The regression models described below are typically additive linear models with extra terms to account for trends in the data.

Serfling Method

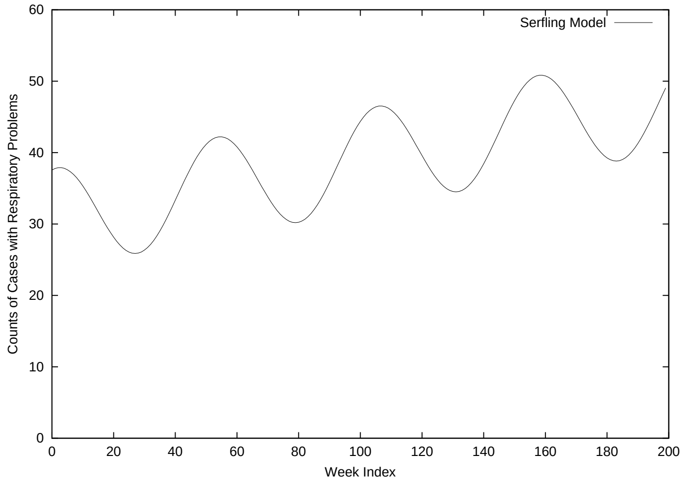


Figure 3.3: A plot of $y(t) = 30.66 + 0.083t + 1.50\text{sin}(2\Pi\frac{t}{52}) + 6.90\text{cos}(2\Pi\frac{t}{52})$

The Serfling Method (Serfling, 1963) was developed to determine the severity of pneumonia and influenza (P&I) in a city by comparing current P&I mortality levels against a baseline that captures both a secular trend and seasonal trends. The baseline is modelled through the general formula below:

$$\hat{Y} = a + b * t + \sum c_i * cos\theta + \sum d_i * sin\theta \tag{3.12}$$

The linear term in equation 3.12 accounts for the secular trend while the sine and cosine terms reflect the seasonal fluctuations. θ is a linear function of t, which is the time period. The coefficients are obtained through regression on a training data set. (Simonsen et al., 1997) include a quadratic term to account for deviations from the linear trend. Figure 3.3 is an example of a model produced by the Serfling method when applied to weekly counts of cases involving respiratory problems. (Tsui et al., 2001). The equation for this model is $y(t) = 30.66 + 0.083t + 1.50\text{sin}(2\Pi\frac{t}{52}) + 6.90\text{cos}(2\Pi\frac{t}{52})$.

An alert threshold is established at 1.64 standard deviations above the baseline. The probability of exceeding this threshold at least once in a non-epidemic year is 0.75 under the model described by (Serfling, 1963). In order to avoid false positives due to noise, an epidemic increase is only signalled if the threshold is exceeded for two consecutive weeks or more.

The Serfling method can produce an inaccurate baseline if there are epidemic periods in the training data. As a result, a variety of methods have been proposed to deal with this problem. The original paper (Serfling, 1963) describes a technique where using only parts of years with low seasonal P&I incidence, the secular trend is determined and subsequently removed from the data. The seasonal trend is then estimated from the modified data. Finally, the secular trend component that was removed is added back in. In (Tsui et al., 2001), the authors smooth the baseline by first fitting the regression model using the raw data. Then, the baseline is computed along with a one-sided 95% confidence interval. Any outliers beyond this confidence interval are removed. The coefficients of the regression model are then recalculated based on the cleaned data set.

ANOVA Regression

When the data exhibits temporal trends, we can supplement linear regression with some extra covariates to produce an algorithm that we will refer to as ANOVA regression. For example, if the amount data varies depending on the day of week, the covariates we would add to the regression would be six binary variables – isMonday, isTuesday, isWednesday, isThursday, isFriday and isSaturday. Similarly, to account for seasonal effects, we can add three binary covariates. If data that are near each other in time tend to have similar measurement, which would indicate some sort of recent trend, then we can add a covariate that measures the count from yesterday. ANOVA regression is a fairly capable detector when applied to a univariate time series with these trends, as was shown in the DARPA challenge (Buckeridge et al., 2004). A similar approach was also taken by (Lazarus et al., 2002), except instead of using linear regression, they used a Generalized Linear Mixed Model.

Log Linear Regression

(Farrington et al., 1996) describe a log linear model used by the Communicable Disease Surveillance Center (CDSC) for monitoring weekly laboratory reports of counts of organisms associated with infectious diseases in England and Wales. The model assumes that the weekly organism count is distributed with mean μ_i and variance $\phi\mu_i$, where ϕ is a dispersion parameter. The mean μ_i incorporates a linear time trend, as shown in Equation 3.13, where the t_i term corresponds to the week index.

$$log\ \mu_i = \alpha + \beta t_i \tag{3.13}$$

In order to account for seasonal fluctuations, the log linear model is only fit to data that come from similar periods in time to t_i. Two parameters are used to restrict the historical data that is used. The b parameter indicates how many years in the past are considered for the baseline while the w parameter specifies a window size that determines which weeks in the baseline years are included.

The algorithm begins by estimating initial values for μ_i and ϕ. Residuals are calculated and the data points are reweighted so that those with high residuals are given a lower weight. This step is necessary in order to alleviate the effects of outbreaks in the baseline counts. Then, the model is refitted to obtain an estimate of ϕ on the modified data. The threshold values are calculated using a $\frac{2}{3}$ power transformation, which is used to correct skewed distributions due to low counts while leaving distributions with larger counts unaffected. Finally, the organisms are ranked in order of exceedance score X:

$$X = \frac{y_0 - \hat{\mu_0}}{U - \hat{\mu_0}} \tag{3.14}$$

In Equation 3.14, y_0 is the current weekly count, $\hat{\mu_0}$ is the mean from the log linear regression, and U is the upper threshold value. An exceedance score greater than one raises an alarm that further investigation is needed. This model is fairly complex and has a large number of parameters, some of which were not described in this brief overview, that need to be tuned in order for this model to perform well.

Generalized Linear Mixed Models

In their system for monitoring ambulatory-care encounter records, (Lazarus et al., 2002) use a generalized linear mixed model to estimate the daily counts for each census tract for each syndrome. Although this algorithm does perform space-time surveillance, the algorithm is applied to each census tract independently. The census tracts are only considered together when accounting for multiple hypothesis testing.

The GLMM form of logistic regression is given by $logit(p_{it}) = x_{it}\beta + b_i$, where x_{it} are the covariates, β consists of the fixed effects, and b_i is a random effect with mean 0 and variance σ_b^2. The covariates account for day of week, month of year, holidays, and a secular time trend. Under this model, $E(y_{it}|b_i) = n_{it}p_{it}$ where y_{it} is the number of ambulatory-care encounters for census tract i on day t, n_{it} is the total number of people in census tract i on day t, and p_{it} is the probability of an encounter with a diagnosis in the syndrome monitored for that census tract and day. The distribution of y_{it} is assumed to be a binomial with $p = p_{it}$ and $n = n_{it}$. In GLMMs, the response variable is assumed to be the result of fixed and random effects. Typically, random effects are used to model any extra variation not already accounted for in the model.

This algorithm first estimates the parameters of the model from historical data. Then, the GLMM is applied to data from the current day in order to calculate $\hat{p}_{it}$. Once $\hat{p}_{it}$ has been determined, the binomial distribution of y_{it} can be used to calculate the probability $Pr(Y_{it} \geq y_{it})$. This probability represents the probability of seeing as extreme a number of counts on that day. In order to compensate for multiple hypothesis testing, this probability is subsequently multiplied by the total number of census tracts. Finally, the number of days between seeing counts as extreme is reported by simply inverting the result.

3.1.4 Time Series Methods

In (Choi & Thacker, 1981), the authors describe several drawbacks to using the Serfling regression model, which was the predominant technique for predicting expected deaths due to pneumonia and influenza. Specifically, the regression model assumes independence over the sequence of observations, thereby making it difficult to model observations that are correlated to each other in time. This behavior makes the Serfling method unable to account for trends that persist in the data for an extended time period and to any recent trends that appear. The authors propose using a more accurate technique for forecasting pneumonia and influenza mortality based on time series analysis.

In addition to (Choi & Thacker, 1981), many other papers have used time series methods to detect anomalies in time in health-care data. The field of time series analysis has been extensively researched (Box & Jenkins, 1976; Chatfield, 1989; Hamilton, 1994) over the years. Techniques have been developed to handle time series with different characteristics, such as data with correlation between data points in time, seasonality, cyclic components, and non-stationarity. These advantages have made time series models naturally applicable to the task of biosurveillance. In this section, we will only be providing a very brief description of the models that are commonly used in biosurveillance. The descriptions of the models below have been summarized from (Chatfield, 1989) and will use the same notation.

Let Z_t be a sequence of random variables in which each Z has mean 0 and variance σ_Z. Furthermore, suppose the Z's are uncorrelated across time ie. $Cov(Z_t, Z_{t+k}) = 0$ for $k \neq 0$. The process Z_t is called a white noise process.

A qth order moving average process, abbreviated as MA(q), is made up of a weighted sum of Z terms. In Equation 3.15, X_t is an MA(q) process and the β's are constants corresponding to the weights.

$$X_t = \beta_0 Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q} \tag{3.15}$$

An autoregressive process of order p, which is also called an AR(p) process, has the following form:

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} \tag{3.16}$$

The term autoregressive refers to the fact that X_t is formed by regressing on its past values in time.

More processes can be represented if we allow a combination of MA and AR processes. An ARMA(p,q) process mixes a pth order AR process with a qth order MA process as shown in Equation 3.17:

$$X_t = \alpha_1 X_{t-1} + \ldots + \alpha_p X_{t-p} + Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q} \tag{3.17}$$

The AR, MA, and ARMA processes described so far model stationary time series. If a time series is non-stationary in the mean, then differencing the time series can produce a stationary process. Define the differencing operator ∇^d as

$$\begin{aligned}
\nabla x_{t+1} &= x_{t+1} - x_t \\
\nabla^2 x_{t+2} &= \nabla x_{t+2} - \nabla x_{t+1} = (x_{t+2} - x_{t+1}) - (x_{t+1} - x_t) = x_{t+2} - 2x_{t+1} + x_t \\
&\vdots
\end{aligned}$$

We can then fit an ARMA(p,q) model to the differenced series to produce an autoregressive integrated moving average or ARIMA process. Let $W_t = \nabla^d X_t$. Then an ARIMA process of order (p,d,q) is defined in Equation 3.18, where the parameter d is typically set to 1.

$$W_t = \alpha_1 W_{t-1} + \ldots + \alpha_p W_{t-p} + Z_t + \beta_1 Z_{t-1} + \ldots + \beta_q Z_{t-q} \tag{3.18}$$

If the time series contains a seasonal component that repeats every s observations, then a seasonal ARIMA or SARIMA model is appropriate. In addition to using simple differencing as described above, SARIMA removes seasonal fluctuations by performing seasonal differencing ∇_s. For example, the differenced time series W_t could be formed by:

$$\begin{aligned}
W_t &= \nabla \nabla_{12} X_t \\
&= \nabla_{12} X_t - \nabla_{12} X_{t-1} \\
&= (X_t - X_{t-12}) - (X_{t-1} - X_{t-13})
\end{aligned}$$

Due to the non-stationarity inherent in health-care data as well as the presence of various temporal trends, SARIMA and ARIMA models are the main time-series methods used in biosurveillance (Choi & Thacker, 1981; Watier et al., 1991; Williamson & Hudson, 1999; Reis & Mandl, 2003). The typical use of time-series methods involves using the model to forecast the current day's signal value and then raise an alert if that value exceeds a threshold.

Alert thresholds for time series forecasts are typically calculated by assuming a normal distribution for the forecast errors. With this assumption, we can establish a confidence interval using a multiple of the standard deviation of the one step forecast errors. Other thresholding methods use Statistical Quality Control (Williamson & Hudson, 1999) methods such as Shewhart and exponentially weighted moving average control charts on forecast errors to establish alert thresholds. Typically, alarm thresholds are based only on the current day's forecast error. In a novel approach, (Reis et al., 2003) suggest using a weighted combination of the last week's forecasts errors to improve detection. The authors experiment with four different weighting schemes, which they call detection filters: one that only considers the current day's error, one that weighs all days equally, one that assigns linearly increasing weight to more recent days, and one that assigns twice as much weight to a particular day than the day before. An alarm is triggered if the weighted sum of the last seven day's forecasts error exceeds a threshold, which was determined retrospectively. The results of these experiments indicate that different filters are more suitable for detecting different sizes of disease outbreaks.

While time-series methods have many useful properties, there are several difficulties in using them. First of all, time-series methods can sometimes miss a slow spreading outbreak that results in a gradual increase in health-care cases. In this case, a time-series model would end up incorporating the subtle increase into its forecast and completely overlook the outbreak (Reis & Mandl, 2003). Secondly, since these models depend on previous observations, a phase shift is introduced relative to the original data and detection of an epidemic can be delayed (Zhang et al., 2003). Thirdly, even though the parameters for a given time series model can be estimated from data (Box & Jenkins, 1976), the user still needs to know which model to choose. Model selection involves either a moderately expensive parameter search or input from a human being. Even after a model is selected, the model is not adaptive and remains accurate as long as future data fits the model well. Should new properties emerge in the time series due to non-stationarity, then a new model needs to be selected.

Time series methods used for biosurveillance are also unable to handle epidemic periods in the historical data used for model fitting. As a result, these epidemic periods need to be smoothed through a preprocessing step. A common solution is to learn a model on the non-epidemic parts of the data and replace the epidemic periods with forecasted values from the learned model (Choi & Thacker, 1981; Watier et al., 1991). A new model is then learned using the modified time series. (Reis & Mandl, 2003) smooth their data by using a trimmed mean seasonal model. Starting with the original pediatric ED data, the authors subtract the overall daily mean, then the mean for the current day of the week, and finally they remove the trimmed mean for the current day of the year. The trimmed mean is obtained by removing the most extreme values at the high and low end. The residuals are then fit with an ARIMA model. Wavelets are another preprocessing technique which we will devote an entire section to in Section 3.1.7.

3.1.5 Compound Smoothing

(Stern & Lightfoot, 1999) developed an early warning system for outbreaks of Salmonella and Shigella based on laboratory reports. Their detection system consists of two phases. In the first phase, five years worth of data are used to produce a baseline of expected counts for each serovar and for each geographic region using monthly counts as a data point. The data points for each year are smoothed using a compound smoothing technique called 4253H (Velleman & Hoaglin, 1981). This yields five yearly baselines, which are then combined into a single baseline by replacing the value for each month by the median of the five yearly values.

The 4253H algorithm is based on a series of smoothing operations that take the current point under consideration along with its $n-1$ closest neighbors and replaces the n data points with the median of that set. The number of data points replaced by their median is called the span of smoother. The 4253H algorithm successively performs smoothing operations of spans 4, 2, 5 and 3, using the output of one as the input to the next operation. Finally, the H step refers to the Hanning running average, which weights the observation d(t) at time t as $d(t) = \frac{1}{4}d(t-1) + \frac{1}{2}d(t) + \frac{1}{4}d(t+1)$. Once the baseline values are smoothed, the residuals are obtained and they too are smoothed by the same process. The smoothed residuals are subsequently added back to the smoothed baseline. In the second phase of the algorithm, the warning threshold is established as two standard deviations above the baseline. The standard deviation is calculated from the differences between the raw data and the smoothed baseline.

3.1.6 Change point Detection

Detection of a maximum, minimum or inflection point in disease occurrence

(Nobre & Stroup, 1994) describe a time series changepoint as a maximum, minimum or inflection point which corresponds to an increase, decrease or change in trend of the occurrence of a disease. (Nobre & Stroup, 1994) detect changepoints by monitoring the numerical first derivative of the signal, which is equivalent to the forecast error in their model. In this model, the signal is assumed to consist of a stochastic trend component plus noise. The stochastic trend is fit using a polynomial model in which the coefficients are first fit using historical data and then updated using exponential smoothing as new data points arrive. Exponential smoothing is equivalent to EWMA, which was discussed in Section 3.1.2. The forecast error is used to calculate a probability index function, which is used to indicate the probability that the numerical derivative is significantly different than zero. Since epidemiologists are typically interested when there is a significant increase in the acceleration of disease cases, the probability index function sets an upper limit above which the numerical derivative is known to be significantly larger than zero. This technique appears to give a large number of false alarms, especially since it essentially just tracks extreme increases in the rate of case occurrences. While some of these false alarms can be reduced through appropriate parameter settings, these parameters are data dependent and require extensive tuning in order for the algorithm to perform optimally.

Detection of a change in distribution

Changepoint detection is more commonly associated with detecting when the distribution of a sequence of random variables shifts to another distribution at some point in time. For example, we could be interested in when the mean of a set of random variables shifts from μ_0 to μ_1. The CUSUM algorithm, which detects such a shift, is an example of a changepoint detection algorithm.

In (Baron, 2002), the author presents the problem of early detection of influenza epidemics as a Bayes sequential changepoint problem. The onset of an influenza epidemic is typically signalled by rapidly increasing mortality rates beyond what is expected. One of the interesting points about this algorithm is that since frequent changes of temperature are known to increase the probability of influenza, the algorithm can incorporate the standard deviation of average daily temperatures as a prior for when the change point happens.

The model used by (Baron, 2002) is a linear expression shown in Equation 3.19, where Y_t is the proportion of deaths caused by influenza-like illnesses during a particular week t, m_t is a general trend, s_t is a seasonal component, and Z_t is an error term. Also defined is the variable X_t, which is simply the differenced series $Z_t = Z_{t-1}$. During non-epidemic periods, X_t is assumed to be a Gaussian with mean 0 while a pre-epidemic trend would cause X_t to have a non-zero mean. The change point parameter ν indicates the time t when the pre-epidemic trend begins.

$$Y_t = m_t + s_t + Z_t \tag{3.19}$$

In order to solve the Bayes sequential changepoint problem, a risk function is defined that penalizes false alarms and late detection of the changepoint. The Bayes stopping rule finds the stopping time that minimizes the risk function. Determining the Bayes stopping rule requires an iterative numeric algorithm to approximate the payoff function, making the calculation computationally intense. This complexity requires the prior distributions for the change point parameter and the loss function to be simple. An alternative method is shown in (Baron, 2002) that is based on asymptotically pointwise optimal (APO) stopping rules (Bickel & Yahav, 1967; Bickel & Yahav, 1968). The APO approach has a closed form expression that is more computationally feasible and as a result, more complex priors can be incorporated into the model. While this algorithm is computationally tractable, the computational complexity is higher than the other biosurveillance algorithms presented so far. In the APO algorithm, the posterior probability Π_t needs to be computed for each time t. The formula for Π_t involves a large number of sums and products.

3.1.7 Wavelets

The Wavelet Transform is commonly used in the field of signal processing. A closely related technique to the Wavelet Transform is the Fourier Transform, which can determine the frequencies inherent a signal by decomposing the signal into sinusoidal components. The Fourier Transform, however, is unable to return any temporal information as to when the frequencies exist in the signal. On the other hand, the Wavelet Transform preserves both the temporal and the frequency information of the signal (Polikar, 2001). The key aspect of the Wavelet Transform is its ability to break a signal down into a series of resolutions, each of which corresponds to the contribution to the signal of different frequencies. Each of these resolutions are statistically independent and by adding them together, we can get back the original signal (Tsui, 1996). The decomposition into different resolutions is achieved through a variety of methods, the most common of which is the Discrete Wavelet Transform(DWT). At each resolution level, the coefficients for the DWT are obtained by fitting scaled and translated versions of a special basis function called the mother wavelet (Polikar, 2001). The DWT is an efficient algorithm, requiring linear time in the size of the signal when the samples are spread evenly throughout the signal. The topic of wavelets is rather extensive and we have provided only a brief overview here. Interested readers should refer to (Polikar, 2001) for an excellent overview while (Percival & Walden, 2000; Ogden, 1997) provide a more in depth coverage of the material.

Wavelets are excellent for modelling time series with a variety of difficult issues such as seasonality, long term trends and any recent upward or downward trends. Most importantly, wavelets can handle non-stationary data in a more autonomous and efficient fashion than an ARIMA model. A non-stationary time series is informally defined as series where the mean and variance change over time (Chatfield, 1989). When using an ARIMA model, non-stationary data requires knowning the degree of differencing in addition to modifying the model every time the data no longer fits it. By choosing the right resolutions in the decomposition, the Wavelet Transform can be used to smooth the data and thereby avoid extreme outliers like negative singularities, which are values that are significantly lower than its neighboring values (Zhang et al., 2003). Negative singularities are common in health-care data, such as low over-the-counter sales due to pharmacies closing on holidays or a low number of visits to an ED due to severe weather conditions.

While the Wavelet Transform is an excellent tool for denoising data, it is essentially a preprocessing step and it cannot be used for forecasting. As a result, an additional prediction layer is needed. A common forecasting technique, called a multiresolution-based predictor (Zhang et al., 2003), fits a predictor at each resolution of the Wavelet Transform. Each predictor makes a one-step forecast independently of the other resolutions. The forecast for the next day is the sum of the one-step forecasts at all the resolutions. Prediction algorithms that can be used include neural networks (Aussem & Murtagh, 1997), AR (Goldenberg et al., 2002; Goldenberg et al., 2003) or any time series algorithm like ARIMA. However, since each resolution has a fairly smooth and regular signal, the use of a simple forecasting algorithm like AR would seem to suffice. We will refer to the multiresolution AR as the MWAR algorithm, as named by (Zhang et al., 2003). In the final step, the MWAR forms a threshold by assuming a normal distribution over the forecast error and thus an alert threshold is established using the mean plus a multiple of the standard deviation. The top graph in Figure 3.4 illustrates the different resolutions produced by the wavelet decomposition of the original signal which is shown in the lower graph of the same figure. The star at the end of each resolution is the predicted value after an AR model is fitted to each resolution level. Another forecasting technique is the Wavelet-based Anomaly

Detector (WAD) (Zhang et al., 2003), which uses only the lowest frequency resolution, called the baseline, instead of all the levels. The WAD then subtracts the baseline from the current day's signal level to obtain the residual. The residuals are assumed to follow a normal distribution, whose mean and standard deviation are obtained from historical data. Once again, the mean plus a positive multiple of the standard deviation is used for a threshold.

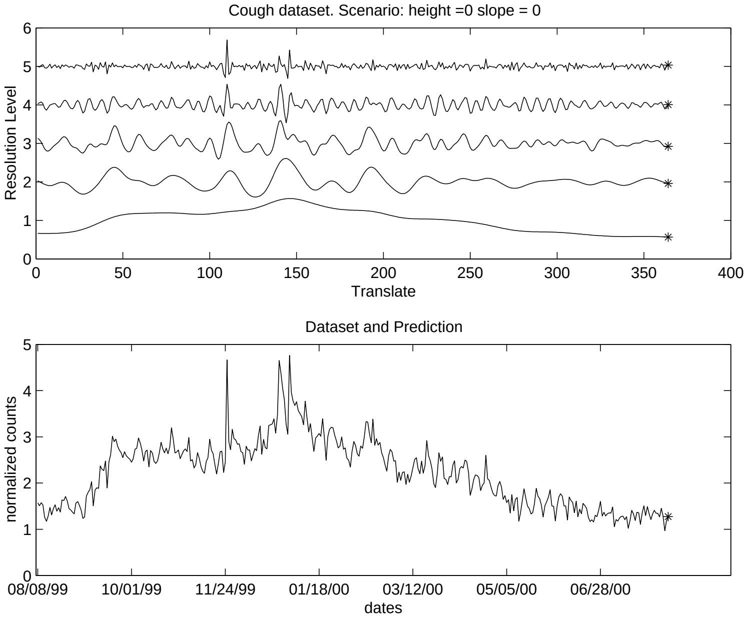


Figure 3.4: The different resolution levels used by MWAR on cough decongestant data with an artificially injected spike (Goldenberg, 2001)

From the results of (Zhang et al., 2003), the WAD algorithm has several features that make it an attractive biosurveillance algorithm. First of all, the WAD is more resistant to negative singularities than MWAR since negative singularities have little effect on the long term trend. In MWAR, when a negative singularity is encountered, the next day forecast by AR will be abnormally low due to the negative singularity. This lowering of the prediction results in a false alarm since the actual signal value is artificially higher than what is expected. In addition, WAD showed improved timeliness over MWAR since it does not introduce a phase shift to the original signal. Due to the nature of AR, it requires order p terms to determine the current day's signal, thereby introducing a phase shift that can be as large as p days. WAD is also much less computationally expensive than MWAR since only one wavelet transform is required to obtain the baseline. MWAR, on the other hand, requires $n+1$ wavelet transforms where n is the maximum level of resolution and it requires fitting and applying an AR model to each level. Finally, WAD needs very little parameter tuning since it is a non-parametric model.

The WAD algorithm, however, does have two main limitations. If a long term outbreak exceeds the baseline trend, then the fluctuations from the outbreak will be removed along with the baseline. As a result, WAD will miss this outbreak. However, as pointed out by the authors of (Zhang et al., 2003), most bio-terrorist attacks are expected to be short and sudden. Secondly, removal of the long term trend may not yield a normal distribution of the residual errors.

3.2 Space Surveillance

By its very definition, surveillance requires a time component. As a result, pure spatial surveillance involves accumulating data over some time interval, removing the time information, and then searching for areas of unusually high incidence of events. These events can be represented either as case events, where the individual locations are known, or as aggregated counts associated with a geographic region. Analysis of such data is typically concerned with finding clusters of events. Figure 3.5 illustrates a taxonomy of disease clustering techniques.

Disease clustering techniques can be divided into non-specific and specific tests (Lawson, 2001). Non-specific tests, also known as tests of general clustering (Besag & Newell, 1991), obtain some statistic of the overall clustering of a disease in a geographic region. Hence, a general clustering test can inform a person that some degree of clustering exists overall but the specific locations of the clusters are not known. In contrast, specific tests identify the locations of the clusters that merit investigation. Although general clustering tests can seem somewhat uninformative for the purposes of surveillance, they are helpful for exploratory analysis. As a result, we will briefly discuss some of these tests. Readers interested in a thorough treatment of disease clustering should consult (Lawson, 2001; Wakefield et al., 2000; Diggle, 2000; Lawson & Kulldorff, 1999; Tango, 1999).

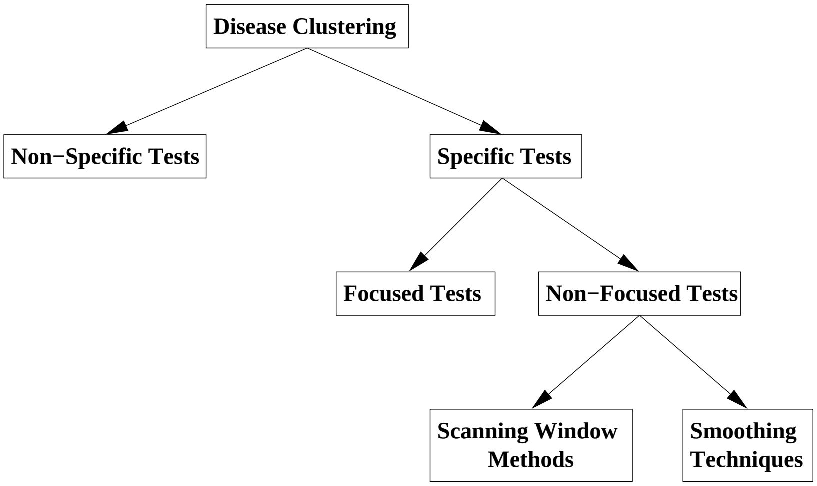


Figure 3.5: Taxonomy of spatial surveillance techniques

3.2.1 General Clustering Tests

Statistical methods in spatial epidemiology commonly model spatial phenomena as the combination of first and second order effects (Bailey & Gatrell, 1995; Lawson, 2001). First order effects cause a trend in the spatial process ie. the mean of the process changes over the entire geographic area. Second order effects are responsible for local effects in which neighboring areas have similar values for the process. These local effects are referred to as spatial autocorrelation or spatial dependence. General clustering tests are typically concerned with measuring the amount of spatial autocorrelation in the data, since a high degree of spatial autocorrelation would indicate localized clustering. The majority of these test take the form shown in Equation 3.20 (Marshall, 1991), where x_{ij} and y_{ij} are some measure of similarity between points i and j.

$$T = \sum_i \sum_j x_{ij} y_{ij} \tag{3.20}$$

The most well-known tests of spatial autocorrelation for tract counts are Moran's I (Moran, 1950) and Geary's C (Geary, 1954). Equation 3.21 shows the formula for Moran's I, where y_i is the value in area i and W is a matrix of spatial proximity. Using similar notation, Equation 3.22 is the formula for Geary's C. From these two equations, the two major differences are that the sample variance replaces the variance in Geary's C and Moran's I uses the distance from the mean as the measure of similarity rather than the actual distances between pairs of points. For complete spatial randomness, the expected value of Moran's I is $\frac{-1}{n-1}$, which is approaches zero for large n. Geary's C is approximately 1 for random data. Table 3.2.1, taken from (Lee & Wong, 2001), illustrates the correlation nature of spatial data when Moran's I and Geary's C fall within a range of values.

$$\text{Moran's I} = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{s^2 \sum\limits_{i \neq j}\sum w_{ij}} \text{ where } s^2 = \frac{\sum\limits_{i=1}^{n}(y_i - \bar{y})^2}{n} \tag{3.21}$$

$$\text{Geary's C} = \frac{\sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} w_{ij}(y_i - y_j)^2}{2\sum\limits_{i \neq j}\sum w_{ij}\sigma^2} \text{ where } \sigma^2 = \frac{\sum\limits_{i=1}^{n}(y_i - \bar{y})^2}{n-1} \tag{3.22}$$

Spatial Patterns	**Geary's C**	**Moran's I**
Clustered pattern	$0 < C < 1$	$I > E(I)$
Random pattern	$C \sim 1$	$I \sim E(I)$
Dispersed pattern	$1 < C < 2$	$I < E(I)$

For case events, the most commonly used method for measuring spatial autocorrelation is the K function (Ripley, 1976). One of the assumptions of the K function is that the process is stationary and isotropic over a small enough scale. Stationarity means that the process has a constant mean and variance. A stationary process is isotropic if the covariance between two points depends solely on the distance and not the direction in which it is separated (Bailey & Gatrell, 1995). Informally, the K function is the expected number of events within distance h of an arbitrary event, scaled by some factor. Equation 3.23 is the formula used for an estimate of the actual K function. In this equation, R is the small scale area under consideration, n is the number of events in R, and $I_h(d_{ij})$ is an indicator function that is 1 if the distance between points i and j is less than or equal to h and 0 otherwise. The K function results are compared to a baseline of a random spatial process, which yields $K(h) = \pi h^2$.

This comparison is achieved through a plot of the L function, given in Equation 3.24, against h. High values in this plot signal the presence of clustering while low values indicate spatial repulsion.

$$\hat{K}(h) = \frac{R}{n^2}\sum_{i \neq j}\sum I_h(d_{ij}) \tag{3.23}$$

$$\hat{L}(h) = \sqrt{\frac{\hat{K}(h)}{\pi}} - h \tag{3.24}$$

The literature for general clustering tests is enormous and we will simply mention other methods. Other tests for general clustering of case events include (Cuzick & Edwards, 1990), (Diggle & Chetwynd, 1991) and (Anderson & Titterington, 1997). Algorithms for aggregated counts include (Whittemore et al., 1987), (Tango, 1995), (Oden, 1995) and (Assuncao & Reis, 1999).

3.2.2 Specific tests

In contrast to general clustering tests, specific tests identify the locations of the clusters that merit investigation. Following the taxonomy of (Lawson, 2001; Besag & Newell, 1991), specific tests can be further subdivided into focused and non-focused tests. Focused testing requires the locations of the clusters to be specified a priori. Hence, these tests are concerned with the degree of disease clustering around a location that is typically a health hazard like a nuclear power plant. In prospective surveillance, the high incidence locations are not known in advance and focused testing is not appropriate. We will concentrate instead on non-focused testing, which does not require prespecified locations of interest and is a subclass of disease clustering algorithms that is suitable for surveillance.

Scanning Window Algorithms

The Geographical Analysis Machine (GAM) (Openshaw et al., 1988) was one of the earliest scanning window algorithms. In this approach, a series of overlapping circular areas centered on a regular grid over the study area are used to find high incidence areas. For each circle, a Monte Carlo test is done to determine if the number of disease cases in the circle is significant at the 0.002 level. If the circle passes the significance test and the number of cases in the circle is greater than 2, then its circumference is drawn on the map. When the procedure is completed, a map with disease clusters should have numerous circles encompassing these clusters.

Apart from the massive computational costs of doing a Monte Carlo test for each circle, this technique has been criticized for not controlling for multiple hypothesis testing (Besag & Newell, 1991). An alternative to GAM, applied to counts associated with a centroid of an administrative zone, was proposed in (Besag & Newell, 1991). Given a zone A_0 and its centroid, an ordering of the other administrative zones is induced based on the distances of their centroids to the chosen centroid. The Besag and Newell algorithm accumulates the counts of cases in these ordered zones, moving outwards from A_0 until the number of cases exceeds a threshold k. Let M be the number of zones required to exceed the threshold. If M is small, then clustering occurs around A_0. In fact, the significance of each "cluster" can be computed since $Pr(M \leq m)$ can be approximated using a Poisson distribution. The choice of k is arbitrary and the authors admit that this method is an ad hoc method but still useful as an exploratory tool.

Besag and Newell's technique examines the number of neighboring areas required to accumulate k number of cases. Another option would be to examine the number of cases over a constant population by aggregating the neighboring areas until the desired population R is reached. (Turnbull et al., 1990) propose such a procedure, relying on a fraction of the total count of an area whenever adding the population of that area exceeds R. Following the terminology of (Turnbull et al., 1990), we will refer to the aggregated areas of constant population as a ball. The statistic of interest is M_R, the maximum number of cases over all the balls. Under the null hypothesis of randomness, we can compute a p-value for the area with M_R cases by using a randomization test to form the null distribution of M_R. One of the awkward points with this method is determining what value of R to use. In (Turnbull et al., 1990), the authors use 4 different R values to evaluate their results.

Cuzick and Edwards (Cuzick & Edwards, 1996) modified their k nearest neighbor test of general clustering (Cuzick & Edwards, 1990) to determine the locations of potential clusters. Suppose a circular region is centered on each case and the radius is fixed such that the expected number of cases, E_j, within the region j is as close to k as possible. If Y_j is the actual number of cases in this circle and n_1 is the total number of cases, then the U_k statistic is defined as:

$$U_k = \sum_{j=1}^{n_1} (Y_j - E_j) \tag{3.25}$$

Both the mean and variance under the null hypothesis can be calculated and the significance of each circular region can be obtained.

Perhaps the most widely used algorithm for spatial biosurveillance is the Spatial Scan Statistic (Kulldorff, 1997), which has been used in NYC DOH (Das et al., 2003) and ESSENCE (Burkom, 2003). We have already discussed the one dimensional scan statistic for detecting anomalies in time in Section 3.1.1. The spatial scan statistic is simply the two dimensional analog.

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, where each entity in the population takes one of two states, or a Poisson model where the counts are produced by an inhomogeneous Poisson process. Whenever c is small in relation to n, the two models are extremely similar to each other (Kulldorff, 1997).

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 use window shapes of circles (Kulldorff, 1999b), ellipses (Kulldorff et al., 2003), and rectangles (Neill & Moore, 2003). 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 3.26 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. Specific versions of equation 3.26 for either the Bernoulli or the Poisson model can be found in (Kulldorff, 1997).

$$S_{\mathcal{W}} = max_{W \epsilon \mathcal{W}} \frac{L(W)}{L_0} \tag{3.26}$$

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.

At first glance, the algorithm seems to be intractable since scanning windows of all possible sizes, a potentially infinte number, need to be considered in order to compute the maximum term in the likelihood ratio. However, as (Kulldorff, 1997) points out, the computation time will be finite since the number of observed points is finite and the likelihood always decreases as we increase the size of the scanning window for a fixed count inside. As a result, the likelihood only needs to be recalculated whenever a new point enters the scanning window as it increases its size. This fact allows many different efficient algorithms to be created.

In (Kulldorff, 1999a), Algorithm 14.3.6 describes a version of the spatial scan statistic that uses variable sized, circular windows with their centers at the grid points. Suppose we have G grid points, R Monte Carlo replications, and M population points eg. aggregated counts at the centroids of a census tract. Each grid point contains a sorted list of the distances to each population point. For each grid point, create a circle for each of the M population points encountered on the list and compute the likelihood for each scanning circle. Once the maximum likelihood is obtained over all the circles, repeat the previous step for the R Monte Carlo replications. The overall complexity of Algorithm 14.3.6 is $O(GMlogM) + O(RGM)$. This algorithm has been implemented in the widely used in version 1.0 of the SaTScan software package (Kulldorff et al., 1997) although version 3.1 is currently available (Kulldorff & Information Management Systems Inc., 2003). (Neill & Moore, 2003) propose a faster implementation of the Spatial Scan Statistic that does not calculate all possible circles.

Smoothing techniques

Another class of space surveillance algorithms involves using smoothing techniques to form a spatial surface where areas of high disease incidence can be identified. The surface created is usually the relative risk, which is defined as the ratio of observed counts to expected counts in an area (Lawson, 2001). By incorporating expected counts into relative risk, we can account for the underlying population at risk for the disease. The risk surface is formed using a variety of smoothing techniques based on parametric and non-parametric models. For surveillance, however, little is known beforehand about what diseases we might encounter and a non-parametric approach might be more appropriate since it makes fewer assumptions about the spatial distribution of the data (Lawson, 2001).

We will discuss three different smoothing methods for creating the relative risk surface – kernel density estimation, kernel binary regression, and Generalized Additive Models (GAM). Before introducing these algorithms, we define the problem as was given in (Kelsall & Diggle, 1998). The task of computing the risk surface can be viewed as a ratio of bivariate densities, one for the observed distribution of cases and one for the expected counts ie. the control data. Let the data consist of $x_1, \ldots, x_{n1}$, which is a randomly sampled proportion q_1 of observed cases from a Poisson process with intensity $\lambda_1(x)$. Similarly, let $y_1, \ldots, y_{n2}$ be a randomly sample proportion q_2 of controls from a Poisson process with intensity $\lambda_2(x)$. The proportions q_1 and q_2 are unknown. Conditioning on n_1 and n_2, the data can be considered as random samples from probability densities $f_1(x)$ and $f_2(x)$, which are proportional to $\lambda_1(x)$ and $\lambda_2(x)$ respectively. The log risk function is given as:

$$\rho(x) = log\left\{\frac{\lambda_1(x)}{\lambda_2(x)}\right\} \tag{3.27}$$

Instead of using the log risk function, the log density function, defined in Equation 3.28, is used to generate the relative risk surface. The log density function differs from the log risk function by a constant.

$$r(x) = log\left\{\frac{f_1(x)}{f_2(x)}\right\} \tag{3.28}$$

Using kernel density estimation, (Kelsall & Diggle, 1995; Bithell, 1990) obtain estimates of the densities $f_1(x)$ and $f_2(x)$. At point x, the estimate $\hat{f}_h(x)$ is computed as:

$$\hat{f}_h(x) = n^{-1} \sum_{i=1}^{n} h^{-2} K\{h^{-1}(x - X_i)\} \tag{3.29}$$

In Equation 3.29, a bivariate Gaussian density is typically used as the kernel function K(). The bandwidth h is determined through a cross-validation procedure described in (Kelsall & Diggle, 1995). Under the null hypothesis, $r(x) = 0$. The significance of the results can be calculated by a randomization test which randomly labels n_1 points as cases and n_2 points as controls.

An alternative approach is to use binary regression. Suppose binary labels $y_1, \ldots, y_n$ are assigned to the data points $x_1, \ldots, x_n$ such that $y_i = 1$ if x_i is a case and $y_i = 0$ if x_i is a control. Conditional on x_i, y_i are realizations of mutually independent Bernoulli random variables Y_i with:

$$P(Y_i = 1 | X_i = x) = p(x) = \frac{q_1 \lambda_1(x)}{q_1 \lambda_1(x) + q_2 \lambda_2(x)} \tag{3.30}$$

From Equation 3.30, we can obtain:

$$logit\{p(x)\} = \rho(x) + c_2 \tag{3.31}$$

$p(x)$ differs from $\rho(x)$ by a constant, just as in the case of kernel density estimation where $r(x)$ differs from $\rho(x)$ by a constant, although they are not the same. When the kernel regression estimator in Equation 3.32 is used to estimate $\hat{p}_h(x)$, we find that $\text{logit}\{\hat{p}_h(x)\} = \hat{r}_h(x) + c_3$ ie. the kernel regression approach and the kernel density estimation method are equivalent up to an additive constant. The bandwidth for this kernel regression method is found using a specialized weighted least squares cross-validation procedure. Kelsall and Diggle prefer kernel regression over the density estimation because the former leads to a better criterion for choosing the bandwidth (Kelsall & Diggle, 1998). P-values can be computed through a Monte Carlo test that generates replicated data similar in distribution to the original data and respecting the null hypothesis. The test statistic used is $\hat{s}(x)$ in Equation 3.33.

$$\hat{P}_h(x) = \frac{\sum_{i=1}^{n} K_h(x - x_i) y_i}{\sum_{i=1}^{n} K_h(x - x_i)} \tag{3.32}$$

$$\hat{s}(x) = \hat{\rho}(x) - \frac{1}{n}\sum_{i=1}^{n} \hat{\rho}(x_i) \text{ where } \hat{\rho}(x) = \text{logit}\{\hat{p}_h(x)\} \tag{3.33}$$

Kernel smoothing is also used by (Lawson & Williams, 1993). In their work, kernel density estimation is first used to smooth the case-control background hazard $\hat{h}(X_i)$. Then, the excess intensity function $\hat{\lambda}(X_i)$ is also smoothed using kernel regression as shown in Equation 3.34 where s is the bandwidth as chosen by cross-validation.

$$\hat{\lambda}(X) = \sum_{i=1}^{n} \hat{h}^{-1}(X_i) K\left(\frac{X_i - X}{s}\right) \tag{3.34}$$

Another possible option for forming the relative risk surface is a generalized additive model (GAM). GAMs are usually used whenever covariates, such as age and gender, need to be included in the regression. This technique can be considered a surveillance method for anomalies in the Space-Person dimension although it is used in (Kelsall & Diggle, 1998) as a purely spatial technique for comparison against kernel density estimation and kernel regression. Suppose we have covariates u and spatial coordinates x. Furthermore, let the probability that the binary response variable Y takes the value 1 be some function $p(x, u)$. Equation 3.35 shows a GAM with a logit link function:

$$\text{logit}\{p(x, u)\} = u'\beta + g(x) \tag{3.35}$$

This equation seems like a regular regression expression except for the term $g(x)$, which is simply a smooth function of x. The algorithm for fitting a GAM uses an iteratively weighted additive model procedure using kernel regression for the data (y_i, x_i, u_i) (Hastie & Tibshirani, 1990). Under the special case of no covariates ie. $\text{logit}\{p(x)\} = g(x)$, we can obtain a GAM estimate of $p(x)$. As (Kelsall & Diggle,

1998) point out, this estimate of $p(x)$ has no direct algebraic relationship to the corresponding kernel regression estimate.

Computing the p-value surface for a GAM relies on the $\hat{s}(x)$ statistic in Equation 3.33, except $\hat{\rho}(x) = \hat{g}(x)$. However, generating the data through Monte Carlo simulations is somewhat more complicated. In order for the generated data to respect the null hypothesis and be of the same distribution as the original data, the probability of an individual being a case must depend on all the covariates while being independent of the spatial coordinates. Overall, GAM works well when covariates need to be taken into account. However, it is a computationally intense procedure and simpler methods could be used whenever covariates are not needed.

3.3 Space-Time Surveillance

With the need for timeliness of detection (Wagner et al., 2001b), several algorithms have begun to exploit both the spatial and temporal characteristics of disease outbreaks in order to decrease detection time. One obvious approach is to split the data into subsets where each subset only contains data from one particular geographic unit. Then, an algorithm that detects temporal anomalies, like those in Section 3.1, is run on each subset of data. Together, this set of algorithms performs a crude spatial-temporal monitoring tool. This approach, however ignores spatial information, such as the proximity of the geographic units, which may improve the performance of the detection algorithm. In addition, if the region of interest consists of many geographic units, then the detection algorithm needs to be run many times, once for each geographic unit's data. Overall, there are fewer algorithms for space-time surveillance as compared to the previous two sections.

3.3.1 Clustering in Space-Time

General clustering techniques, which summarize the overall degree of clustering, also exist for spatio-temporal data. Once again, the T statistic in Equation 3.20 is used in almost all of these tests. The two most well-known tests for space-time interaction between case events are the Knox test (Knox, 1964) and Mantel's test (Mantel, 1967). In the Knox test, the x_{ij} values in the T statistic are the spatial adjacency, which is set to 1 if the cases are less than a threshold spatial distance apart. Likewise, the y_{ij} values in the T statistic are the temporal adjacency with their values being 1 if the cases are less than a threshold temporal distance apart and 0 otherwise. If space-time interaction is present, the value of this test statistic will be large. The null hypothesis under the Knox test is independence of space and time. Its distribution can be approximated by randomizing the time locations of the data points, hence a p-value can be calculated. Mantel's test is almost identical to the Knox test except the actual space and time distances are used instead of whether or not they exceed the threshold value. Another test that has been developed is a general K function for spatio-temporal data (Diggle et al., 1995). As was mentioned earlier, these tests can measure the level of global clustering in the region but are unable to pinpoint the disease cluster locations.

3.3.2 Koch and McKenna

(Koch & McKenna, 2001) consider each geographic region r as the center of a region-cluster, where a region-cluster, for example, can be defined as the surrounding neighbors of r. For a given time interval, the number of cases in each region is assumed to be Poisson distributed with a mean that can be estimated from historical data. The authors propose a hypothesis test where under the null hypothesis, at least one region in a region cluster has a non-significant time cluster while in the alternative hypothesis, each region in the region cluster has a significant time cluster. Let Φ be the entire set of regions and $R(r)$ be the region cluster with region r as its center. Furthermore, if $y_{i,\tau}$ is the number of cases in region i at time τ, then a very conservative p-value of the entire map is defined as:

$$Pval(\Phi) = \underset{r \epsilon \Phi}{min}(\underset{i \epsilon R(r)}{max}\ Pval(y_{i,\tau})) \tag{3.36}$$

3.3.3 Spatial CUSUM

CUSUM algorithms have been extended to incorporate spatial information in order to produce a space-time surveillance algorithm. In (Raubertas, 1989), the author uses CUSUM to monitor anomalies over neighborhoods. A matrix S is defined where $s_{il} = 1$ if points i and l are within a cutoff distance and $s_{il} = 0$ otherwise. Let x_{ij} be the observed cases in a geographic unit i at time period j. The observed count of cases in a neighborhood is defined in Equation 3.37:

$$x'_{ij} = \sum_{l=1}^{s} s_{il} x_{lj} \tag{3.37}$$

The neighborhood CUSUM statistic is then defined in Equation 3.38 with k'_i being the neighborhood reference value.

$$w'_{ij} = max(0, w'_{i,j-1} + (x'_{ij} - k'_i)) \tag{3.38}$$

The effectiveness of this approach lies in the definition of the neighborhoods. If the majority of units in a neighborhood experience an anomaly, then the sensitivity of the detector will increase. On the other hand, unusual behavior localized to one geographic unit will likely be hidden when the data is accumulated for the entire neighborhood. The choice of neighborhood size is difficult to know *a priori*. Another problem with this approach is the difficulty in accounting for multiple hypothesis testing, which can occur at two levels. For a given neighborhood size, each neighborhood has a hypothesis test. Furthermore, each neighborhood can have an arbitrary size. Raubertas proposes a solution for the former problem but not the latter. Multiple hypothesis testing for a fixed neighborhood size is addressed by controlling the neighborhood baseline ARL called ARL_0^*, which is the time till the first alarm for the entire neighborhood. For small and large neighborhood sizes, approximations for ARL_0^* can be made. However, a trial and error process through computer simulations are necessary for intermediate neighborhood sizes.

Another spatial CUSUM algorithm is proposed in (Rogerson, 1997) where CUSUM is used to monitor Tango's clustering statistic (Tango, 1995), which is a generalization of the clustering index discussed in Section 3.1.1 applied to spatial data. Recall that a cluster in time was detected whenever Tango's clustering index is large. Likewise, a cluster in space is detected whenever Tango's clustering statistic C_G is large. By using CUSUM to monitor for a shift from the expected value of C_G, changes in spatial patterns over time can be noticed.

Let $C_{G,i}$ be Tango's clustering statistic based on the first i observations. CUSUM is then applied to the statistic Z_i, which is defined as:

$$Z_i = \frac{C_{G,i} - E[C_{G,i}|C_{G,i-1}]}{\sigma_{C_{G,i}|C_{G,i-1}}} \tag{3.39}$$

This statistic does not have a normal distribution. However, by accumulating n observed values of Z_i together, we can form their mean $\bar{Z}_{(n)}$, which is normally distributed. The value of n needs to be small in order to improve timeliness of detection yet large enough to result in a normal distribution for $\bar{Z}_{(n)}$. In (Rogerson, 1997), n is set to 4. The reference value k is set to be a multiple of $\sigma_{\bar{Z}_{(n)}}$, which is equal to $\frac{1}{\sqrt{n}}$, and the corresponding value of h is obtained from a standard CUSUM table.

Rogerson extends his approach in (Rogerson, 2001) by using a local Knox statistic in place of Tango's clustering statistic in 3.39. Define $n_{st}(i)$ to be the number of points that are close to observation i in both time and space. Let the local Knox statistic $N_{st}(i)$ be equal to $n_{st}(i)$. The distribution of the local Knox statistic under the null hypothesis of independence of time and space can be computed through a hypergeometric distribution or a normal approximation. Using the normal approximation, an adjusted z-score can be computed as shown in Equation 3.40, with $E\{N_{st}(i)\}$ and $V\{N_{st}(i)\}$ being the mean and variance respectively of the local Knox statistic under the normal approximation.

$$Z_i = \frac{n_{st}(i) - E\{N_{st}(i)\} - 0.5}{\sqrt{V\{N_{st(i)}\}}} \tag{3.40}$$

Once again, CUSUM is used to monitor the Z_i statistic, where Z_i is computed for each new observation i.

3.3.4 Space-Time Scan Statistic

The Space-Time Scan Statistic is an extension of the spatial scan statistic (Kulldorff, 1999b; Kulldorff, 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. This procedure is computationally intensive and is intended for disease-specific surveillance (Kulldorff, 2001). Furthermore, the space-time scan statistic has low power against "stretched out" clusters but it is successful at detecting fairly compact clusters (Kulldorff, 1999b).

3.4 Space-Time-Person Surveillance

The algorithms in the previous sections deal with finding anomalies in one, two or three dimensions. Case event health-care data used for surveillance is currently available which has greater than 3 dimensions. While time will always be a single dimension, space information for cases includes home location, work location and hospital location. Since cases involve people, additional information such as age, gender, race and symptom information can also be included. This rich multivariate data can of course be aggregated into groups of interest and a time, space, or space-time algorithm could be run on each group. However, algorithms that consider such multivariate data as a whole are very scarce.

To our knowledge, only three algorithms have taken this multivariate approach. All of these algorithms ignore proximity information among spatial units by simply treating each geographic region like a zipcode as a value for a categorical attribute. The first such algorithm is WSARE, which is the focus of this thesis. WSARE takes a baseline data set and a recent data set and determines which groups in the recent data set are most anomalous when compared to those in the baseline. A somewhat similar approach was taken by (Brossette et al., 1998) for hospital infection control and public health surveillance. Like WSARE, they use association rules to characterize and compare groups. However, (Brossette et al., 1998) lacks the additional steps of a randomization test and FDR that are present in WSARE. In addition, unlike WSARE 3.0, they do not account for expected temporal fluctuations such as seasonal and day of week effects.

Cooper's BCD algorithm (Buckeridge et al., 2004) is a changepoint detection algorithm that takes a true multivariate approach to disease detection. BCD operates on data consisting of a variable X to be monitored along with other variables W which predict X. For instance, if we are monitoring ED chief complaint data, X could be a binary attribute stating whether or not the case involved respiratory problems while W could be variables such as age, gender, and home zip code. Like WSARE, BCD takes recent data, which we will call D_r, and compares it to data from a baseline period, which we will refer to as D_s for data from a "safe period". The BCD algorithm first computes the LR statistic shown in Equation 3.41, where the hypothesis $H_{noChange}$ assumes that the data in D_r and D_s are distributed identically and the hypothesis H_{change} assumes that a changepoint occurs ie. at some point in time, the distribution of D_r changed.

$$LR = \frac{P(D_r|H_{change})}{P(D_r|H_{noChange})} \tag{3.41}$$

In order to calculate the probabilities in Equation 3.41, we need to first define time points t_0, for when we start collecting D_r, and t_{now} for the current time. Moreover, define $H_{change}(t)$ as the hypothesis that from time t_0 to t inclusive, the data is drawn from the same distribution as D_s while from time $t+1$ to t_{now}, the data is possibly drawn from a different distribution. From this definition, $H_{change}(t_{now}) = H_{noChange}$.

BCD uses Bayesian networks to compute $P(D_r|H_{change(t)})$. Let the Bayesian network $B = (\theta, S)$, where S is a directed acyclic graph and θ is a set of parameters for the Bayesian network. The Bayesian network structure is learned from data, using some function $s(D)$ which returns a structure S maximizing $P(D|S)$ over the predictor variables W of X. We can now calculate the probability $P(D_r|H_{change(t)})$ as shown in Equation 3.42, where the notation $D_r(t_i, t_j)$ indicates a subset of the data in D_r from time t_i to t_j. The integral in this equation can be solved using a closed form expression as shown in (Cooper & Herskovits, 1992). From Equation 3.42, we can then compute $P(D_r|H_{change})$ as shown in Equation 3.44, where n is the number of changepoints from t_0 to t_{now-1} inclusive.

$$P(D_r|H_{change}(t)) = P(D_r(t_0,t)|s(D_s), D_s) \int P(D_r(t, t_{now})|\theta) P(\theta|s(D_r(t, t_{now})))d\theta \tag{3.42}$$

$$P(D_r|H_{change}) = \sum_{t=t_0,t_{now}-1} P(D_r|H_{change}(t))P(H_{change}(t)) \tag{3.43}$$

$$\text{where } P(H_{change}(t)) = \frac{1}{n} \tag{3.44}$$

P-values for BCD are calculated by obtaining a set of LR statistics for some time interval during the safe period. The value of the LR statistic for the current time is then ranked against this set of LR statistics and the p-value is obtained accordingly.

3.5 Other Work Related to WSARE

On the surface, WSARE may seem like an anomaly detection algorithm such as those in (Bishop, 1994; Hamerly & Elkan, 2001), yet its behavior is quite different as we will discuss in Chapter 4. Instead, WSARE is similar in flavor to algorithms that find differences between data sets. One of these algorithms develops a global measure of the deviation between two data sets (Ganti et al., 1999). In another method, (Chakrabarti et al., 1998) propose using a minimum description length approach to monitor the degree of surprise of a temporal pattern in market basket data. The work by (Dong & Li, 1999) finds emergent patterns, which is defined as an itemset in which the ratio of support between the two data sets exceeds some threshold.

Out of all the techniques for finding data set differences, the most similar approach to WSARE is taken in contrast set mining (Bay & Pazzani, 1999). Contrast set mining involves finding association rules, called constrast sets, that are best at distinguishing two or more groups. Unlike WSARE, contrast set mining searches over a larger space of association rules because rules with m components are potentially allowed, where m is the total number of attributes. However, in order to make the algorithm tractable, constrast set mining relies heavily on a variety of pruning rules. One of these pruning rules is to ignore association rules that yield counts that invalidate the use of a Chi-Square test. WSARE, on the other hand, is able to handle such association rules with Fisher's exact test and in fact, some of these rules may be interesting from a surveillance perspective. Another difference between WSARE and contrast set mining is that in contrast set mining, multiple hypothesis testing problems are controlled with a Bonferroni correction while in WSARE, a randomization test is used. Finally, temporal trends are not taken into account in contrast set mining.

Chapter 4

WSARE 2.0

4.1 Introduction

Multidimensional data with a temporal component is available from numerous disciplines such as medicine, engineering, and astrophysics. This data is commonly used for monitoring purposes by a detection system. These systems inspect the data for anomalies and raise an appropriate alert upon discovery of any deviations from the norm. For example, in the case of an intrusion detection system, an anomaly would indicate a possible breach of security (Lane & Brodley, 1999; Eskin, 2000; Maxion & Tan, 2001).

We would like to tackle the problem of early disease outbreak detection in a similar manner. In our situation, we have real-time access to a database of emergency department (ED) cases from several hospitals in a city. Each record in this database contains information about the individual who was admitted to the ED. This information includes fields such as age, gender, symptoms exhibited, home location, work location, and time admitted. (To maintain patient confidentiality, the regulations described in Chapter 2 were followed ie. personal identifying information, such as patient names, addresses, and identification numbers were not in the data set used in this research.) Clearly, when a severe epidemic sweeps through a region, there will 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. Furthermore, most existing detection algorithms work solely with the number of ED visits over time while ignoring other information available in ED records. We would like to take advantage of all available information, particularly the temporal, spatial, demographic, and syndromic aspects of ED records (Wagner et al., 2001c) to improve the timeliness of detection.

Although we have posed our problem in an anomaly detection framework, the majority of anomaly detection algorithms are inappropriate for this domain. In this section, we will illustrate the shortcomings of traditional anomaly detection techniques in our task of early epidemic detection.

A simplistic first approach would be to report an ED case as an anomaly if it has a rare value for some attribute. As an example, we would signal an anomaly if we encountered a patient over a hundred years old. While this method detects the outliers for a single attribute, it fails to identify anomalies that occur due to combinations of features which by themselves might not be abnormal but together would certainly be unusual. For instance, the first technique would not find anomalies in cases where the patients were male and under the age of thirty but exhibited symptoms that were associated with a disease that affects primarily female senior citizens. Fortunately, there are plenty of anomaly detection algorithms that can identify outliers in multi-dimensional feature space. Typically these detection algorithms build a probabilistic model of the "normal" data using a variety of techniques such as neural nets (Bishop, 1994) or a mixture of naive Bayes submodels (Hamerly & Elkan, 2001)

However, even that kind of sophisticated outlier detection is insufficient for our purposes. Outlier detection succeeds at finding data points that are rare based on the underlying density, but these data points are treated in isolation from each other. Early epidemic detection, on the other hand, hinges on identifying anomalous groups, which we will refer to as *anomalous patterns*. Specifically, we want to know if the recent proportion of a group with certain characteristics is anomalous based on what the proportion is normally. Traditional outlier detection will likely return isolated irregularities that are insignificant to the early detection system.

We might then argue that aggregate daily counts of a single attribute or combination of attributes should be monitored in order to detect an anomalous group. For instance, we could monitor the daily number of people appearing in the ED with respiratory problems, thereby collapsing a multivariate data set into a univariate one. A naive univariate detector such as a Shewhart control chart (Montgomery, 2001) would determine the mean and variance of the monitored signal over a training set which is assumed to capture the normal behavior of the system. Then, a threshold would be established based on these values. Whenever the daily count exceeds this threshold, an alert is raised. This technique works well if the monitored features are known. However, the spatial, temporal, demographic, and syndromic signatures of diseases are simply too wide a space for us to know a priori what features to monitor. We could well miss some combination of features that would indicate an outbreak of a particular disease. Thus, we need a multivariate algorithm that is able to detect anomalous patterns rather than pre-defined anomalies.

At this point, we would like to reiterate the fact that our task involves non-specific disease detection. If we know a priori which diseases to monitor, we can improve the timeliness of detection by monitoring specific features of the disease. For example, if we were vigilant against an anthrax attack, we can concentrate our efforts on ED cases involving respiratory problems. With non-specific disease detection, however, we need to resort to monitoring health-care data for any irregular patterns.

Our approach to this problem uses a rule-based anomaly pattern detector. Each anomalous pattern is summarized by a rule, which in our current implementation consists of one or two components. Each component 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 Age_Decile = 4. One important benefit to a rule-based system is that the rules are easily understood by a non-statistician.

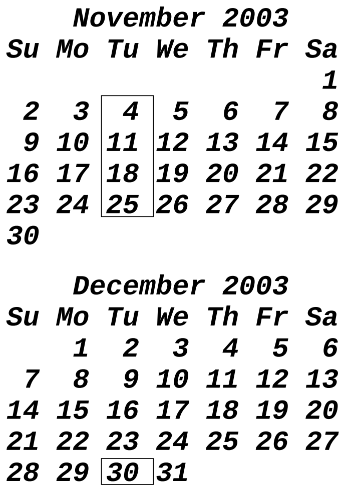
Nevertheless, we need to be wary of the pitfalls of rule-based anomaly pattern detection. Since we are finding anomalous patterns rather than isolated anomalies, we will be performing multiple hypothesis tests. When multiple hypothesis tests are performed, the probability of a false positive becomes inflated unless a correction is made (Benjamini & Hochberg, 1995). In addition, as we add more components to a rule, overfitting becomes a serious concern. A careful evaluation of significance is clearly needed. Furthermore, temporal health-care data used for disease outbreak detection are frequently subject to “seasonal” variations. As an example, the number of influenza cases is expected to be higher during winter than summer. Additionally, the number of ED visits vary between weekends and weekdays. The definition of what is normal will change depending on these variations.

4.2 Rule-based Anomaly Pattern Detection

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. In order to define an anomaly, we need to establish the concept of something being normal. Our algorithm is intended to be applied to a database of ED cases and we need to account for environmental factors such as weekend versus weekday differences in the number of cases. Consequently, baseline behavior is assumed to be captured by the events occurring on the days that are exactly five, six, seven, and eight weeks prior to the day under consideration. This range of days was chosen because WSARE 2.0 performed better with this setting on our evaluation data than with a baseline of one, two, three, and four weeks prior. We would like to emphasize that the definition of this baseline period can be easily modified to another time period without major changes to our algorithm. This baseline period must be chosen to be from a time period similar to the current day. Often this is 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.

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 five to eight weeks ago 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 4.1. These dates are all from Tuesdays in order to avoid day of week variations.

WSARE 2.0 operates on discrete data sets with the aim of finding rules that characterize significant patterns of anomalies. This algorithm is intended to be used in combination with other detection algorithms, preferably those that are designed to monitor specific diseases. WSARE's role in this suite of detectors is to act as a safety net, so that any irregular patterns in health-care data that are not found by



```
    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 4.1: The baseline for WSARE 2.0 if the current day is December 30, 2003

the other detectors will trigger an alert with WSARE. An overview of the WSARE algorithm will be given followed by a more detailed example.

4.2.1 Overview of WSARE

The best rule for a day is found by considering all possible one and two component rules over events occurring on that day and returning the one with the best "score". For computational reasons, we limit the maximum number of components in WSARE rules to be two. This restriction does not contradict the multivariate nature of WSARE because WSARE still searches over all the attributes to find the combination that forms the best scoring rule. In contrast, a univariate detection algorithm would need to know what combination of attributes to monitor beforehand. The score of a rule is determined by comparing the events on the current day against events in 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 case features and date are independent. The randomization-based p-value takes into account the effect of the multiple testing that went on during the rule search. If we were running the algorithm on a day-by-day basis we would end at this step. However, if we were looking at a history of days and we wanted 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 & Hochberg, 1995) to determine which of the p-values are significant. The days with significant p-values are returned as the anomalies.

4.2.2 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 1000 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 five, six, seven, and eight weeks prior to the current day. This baseline data set was chosen to avoid variations due to the day of the week, such as more people appearing at the ED on Mondays as opposed to Saturdays, and also to be recent enough that any current trends could be incorporated. 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, retrieve the records belonging to DB_i. We first consider all possible one-component rules. For every possible feature-value combination, obtain the counts C_{recent} and $C_{baseline}$ from the data set DB_i. As an example, suppose the feature under consideration is the Age_Decile for the ED case. There are 9 possible Age_Decile values, 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 4.1.

4.2.3 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 differentiate this p-value from the p-value that is obtained later on from the randomization test. We use the Chi Squared test for independence of variables whenever the counts in the contingency table do not violate the validity of the Chi Squared 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 Squared test is an approximation to Fisher's exact test when counts are large. Running Fisher's exact test on Table 4.1 yields a score of 0.00005058, which indicates that the count C_{recent} for cases matching the rule Age_Decile = 3 are significantly different from the count $C_{baseline}$.

	C_{recent}	$C_{baseline}$
$Age\_Decile = 3$	48	45
$Age\_Decile \neq 3$	86	220

Table 4.1: A Sample 2x2 Contingency Table

4.2.4 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 feature-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

BR_i^2, however, may not be an improvement over BR_i^1. We need to perform further hypothesis tests to determine if the presence of either component has a significant effect. This can be accomplished by determining the scores of having each component through Fisher's exact test. If we label BR_i^2's components as C_0 and C_1, then the two 2-by-2 contingency tables for the Fisher's exact tests are shown in Table 4.2.

Once we have the scores for both tables, we need to determine if they are significant or not. We use the standard α value of 0.05 and considered a score to be significant if it is less than or equal to α. If the scores for the two tables are both significant,

Records from Today matching C_0 and C_1	Records from Other matching C_0 and C_1
Records from Today matching C_1 and differing on C_0	Records from Other matching C_1 and differing on C_0

Records from Today matching C_0 and C_1	Records from Other matching C_0 and C_1
Records from Today matching C_0 and differing on C_1	Records from Other matching C_0 and differing on C_1

Table 4.2: 2x2 Contingency Tables for a Two Component Rule

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.

4.2.5 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 feature-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. For example, if we have a 3 component rule with components C_0, C_1, and C_2, there are 6 2x2 tests we need to perform as shown in Table 4.3. In general, we need $2\sum_{i=1}^{\lfloor \frac{n}{2} \rfloor} \binom{n}{i}$ such tests.

Clearly, having this many tests is computationally intense 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 4.4, where C_n refers to the last component added and C_{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 matching C_0 AND C_1 AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_0 and differing on C_1 AND C_2	Records from Other matching C_0 and differing on C_1 AND C_2

Records from Today matching C_0 AND C_1 AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_1 AND C_2 and differing on C_0	Records from Other matching C_1 AND C_2 and differing on C_0

Records from Today matching C_0 AND C_1 AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_1 and differing on C_0 AND C_2	Records from Other matching C_1 and differing on C_0 AND C_2

Records from Today matching C_0 AND C_1, AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_0 AND C_2 and differing on C_1	Records from Other matching C_0 AND C_2 and differing on C_1

Records from Today matching C_0 AND C_1 AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_2 and differing on C_0 AND C_1	Records from Other matching C_2 and differing on C_0 AND C_1

Records from Today matching C_0 AND C_1 AND C_2	Records from Other matching C_0 AND C_1 AND C_2
Records from Today matching C_0 AND C_1 and differing on C_2	Records from Other matching C_0 AND C_1 and differing on C_2

Table 4.3: 2x2 Contingency Tables for a Three Component Rule

4.2.6 Finding the p-value for a rule

The algorithm above for determining scores is extremely prone to overfitting. 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 turn to a

Records from Today matching C_{n-1} and C_n	Records from Other matching C_{n-1} and C_n
Records from Today matching C_{n-1} and differing on C_n	Records from Other matching C_{n-1} and differing on C_n

Records from Today matching C_{n-1} and C_n	Records from Other matching C_{n-1} and C_n
Records from Today matching C_n and differing on C_{n-1}	Records from Other matching C_n and differing on C_{n-1}

Table 4.4: 2x2 Contingency Tables for an N Component Rule

randomization test in which the date and each ED case features are assumed to be independent. In this test, the case features 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 ie. 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 $CPV_i$ ie.
```

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

It is clear from this procedure that 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 features.

4.2.7 Using FDR to determine which p-values are significant

This algorithm can be used on a day-to-day basis similar to an online algorithm 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 compare the CPV values for each day in the history. Comparison of multiple CPV values 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 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 at least one false positive 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 & Hochberg, 1995), which we will refer to as BH-FDR. The BH-FDR method is more desirable than the Bonferroni correction because it has a higher power and still has reasonable control over the number of false positives.

BH-FDR first requires the selection of a desired false discovery rate α_{FDR}, where $0 \leq \alpha_{FDR} \leq 1$. The false discovery rate is defined as the number of times the null hypothesis was falsely rejected divided by the total number of tests in which the null hypothesis was rejected. The procedure then takes the list of original p-values from the hypothesis tests and returns a list of p-values whose corresponding null hypotheses should be rejected. We will call these p-values "significant". The BH-FDR procedure guarantees that the false discovery rate is less than or equal to α_{FDR}.

The BH-FDR algorithm first sorts the observed p-values in increasing order, yielding $p_{r_1} \leq p_{r_2} \leq \ldots \leq p_{r_m}$. Define the index i^* to be $i^* = max\left\{i : p_{r_i} \leq \frac{i}{c_m m}\alpha\right\}$. The null hypothesis for the tests corresponding to p-values $p_{r_1}, \ldots, p_{r_{i^*}}$ are rejected. If i^* does not exist, then no null hypotheses are rejected. If the original p-values are obtained from independent hypothesis tests, then $c_m = 1$. On the other hand, if the tests are dependent, then $c_m = \sum_{l=1}^{m} \frac{1}{l}$, resulting in a more conservative version of BH-FDR under arbitrary dependence. We incorporate the FDR method into our rule-learning algorithm by first providing an α_{FDR} value and then using FDR to find the cutoff threshold for determining which p-values are significant.

4.3 Computational Issues

The bottleneck in the entire WSARE procedure is clearly in the randomization test. If implemented naively, it can be extremely computationally intense. In order to speed up this portion of the algorithm, we implemented the following computational tricks.

4.3.1 Racing

We invoke the old idea of "racing" (Maron & Moore, 1997) during the randomization procedure. 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 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 better score than the best scoring rule over the original unrandomized data set). Using a normality assumption on the distribution of CPV, we can estimate the standard deviation σ_{CPV} and form a 95% confidence interval on its final value. This is achieved using the interval $CPV_i^j \pm \frac{1.96\sigma_{CPV}}{\sqrt{n}}$. If the lower half of this interval 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.

4.3.2 Early computation cutoff for p-value calculation

The only information that is actually required during each iteration of the randomization test is whether the score of the best rule found over the randomized data set is higher than the original score. Computing the score using Fisher's exact test requires a summation over a range of parameter values for the hypergeometric distribution. Since we do not need to know the exact score value, only whether or not it is higher than the original score, we can cut the computation short once the summation exceeds the original score.

4.3.3 Differential Counting

Suppose that there are M attributes and all the attributes have an arity of K. In addition, let there be N_T records for today and N_B records for the baseline period. Note that N_T is usually 4 to 20 times smaller than N_B. Also, assume that there are Q iterations of the randomization test. The complexity of the randomization test step is $O(QKM(N_T + N_B))$.

In order to explain the complexity, recall that at iteration j of the randomization test, we need to search for the best scoring rule over the $DB_i^{(j)}$. Searching for the best rule requires scoring every possible rule, of which there are $KM + KM$ since we are building the rules using a greedy search and we allow each rule to have a maximum of 2 components. 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 $2KM(N_T + N_B)$. With Q iterations, the overall complexity, written in big O notation, is $O(QKM(N_T + N_B))$.

For the special case where the maximum number of components for a rule is two and we are building the rules greedily, we can improve the speed of this algorithm through a space-time tradeoff. Let C_{Total} be a hash that stores for each possible rule, the number of records that match that rule over the union of rows used for the "recent" and "baseline" periods of the data set ie. DB_i. We will use the notation that $C_{Total}(r)$ retrieves the count of the number of records matching a rule r on DB_i. Furthermore, let $C_{Today}(r, j)$ be the number of records matching rule r on the "today" rows for randomization test iteration j and let $C_{Baseline}(r, j)$ be the number of records matching rule r on the "baseline" rows for randomization test iteration j. We can use the subtraction $C_{Baseline}(r, j) = C_{Total}(r) - C_{Today}(r, j)$ to obtain $C_{Baseline}(r, j)$.

For the entire randomization test, C_{Total} needs to be built only once since the counts it stores are unaffected by randomization. The cost of building C_{Total} is $O(K^2M^2(N_T + N_B))$. On each iteration of the randomization test and for each possible rule, we need to obtain $C_{Today}(r, j)$ and $C_{Baseline}(r, j)$. $C_{Baseline}(r, j)$ requires a constant amount of work and we can ignore its cost. Obtaining $C_{Today}(r, j)$ requires a pass through N_T records. Over all rules, the cost per randomization test iteration is $2KMN_T$ since rules have at most 2 components and they are constructed greedily. With Q randomization test iterations, we write the overall cost for the entire randomization test using big O notation as $O(K^2M^2(N_T+N_B))+O(QKMN_T)$. Theoretically, this optimization allows us to perform the randomization test approximately 10 times faster, given fairly representative values of $K = 2$, $M = 20$, $Q = 1000$, $N_T = 500$, and $N_B = 10000$.

4.4 Conclusion

The three main innovations of WSARE 2.0 are:

1. Turning the problem of "detect the emergence of new patterns in recent data" into the question "is it possible to learn a propositional rule that can significantly distinguish whether records are most likely to have come from the recent past or more distant past?"

2. Incorporating several levels of significance tests into rule learning in order to avoid several levels of overfitting caused by intensive multiple testing

3. Examining the interesting domain of early outbreak detection by means of machine learning tools

Chapter 5

WSARE 3.0

5.1 Introduction

Early disease outbreak detection systems monitor a variety of surveillance data for any irregularities due to the onset of an epidemic. The observed data are assumed to be the sum of cases from background activity, which we will refer to as the baseline, plus any cases from current outbreaks. Under this assumption, outbreak 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 is to estimate the baseline distribution using data from historical data. 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 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 5.1, 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 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.

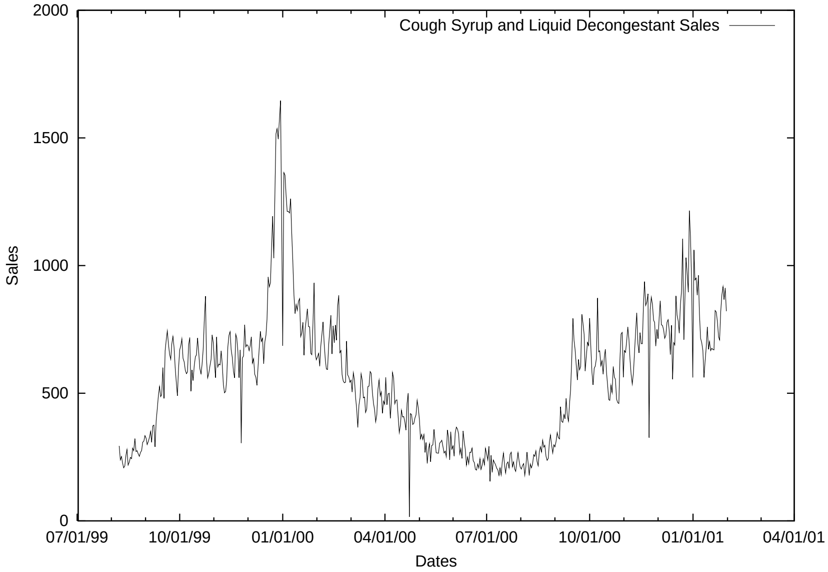


Figure 5.1: Cough syrup and liquid decongestant sales from (Goldenberg et al., 2003)

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, 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 the previous chapter, 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 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 chapter, 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 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.

5.2 WSARE 3.0

Before proceeding with a description of WSARE 3.0, we will give a brief summary of the previous chapter which described how WSARE 2.0 operates. 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 Chapter 4 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 behavior 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 large data set sampled from the Bayesian network. We will describe this step in detail below, while the other parts of the algorithm are identical to the steps in WSARE 2.0.

5.2.1 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 on in Chapter 6, 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). An overview of Optimal Reinsertion will be discussed in section 5.2.2. 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 $P(X_1, ..., X_n \mid \text{Season} = \text{Winter}, \text{ Day of Week} = \text{Weekday})$ 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. 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.

5.2.2 Optimal Reinsertion

The task of learning a Bayesian network structure from data has been well studied over the years (Heckerman et al., 1995; Cooper & Herskovits, 1992). However, this problem has been shown to be NP-Hard (Chickering et al., 2003; Chickering, 1996a), and although a variety of clever algorithms have tackled some of the issues (Chickering, 1996b; Heckerman et al., 1995; Friedman & Goldszmidt, 1997; Xiang et al., 1997;

Friedman et al., 1999; Elidan et al., 2002; Hulten & Domingos, 2002), the computational cost is still quite daunting. In general, most structure learning algorithms find a directed acyclic graph (DAG) D that maximizes a scoring function which we will call $DagScore(D)$. One essential feature of $DagScore(D)$ is that it can be decomposed into components, one for each node in the graph. In equation 5.1, $Parents(i)$ are the parents of node i and $NodeScore(Parents(i) \to i)$ is the score of the node i given its parents. Most scoring functions like BIC (Schwartz, 1979), BD (Cooper & Herskovits, 1992), BDE (Heckerman et al., 1995), and BDEU (Buntine, 1991) have this decomposable characteristic.

$$DagScore(D) = \sum_{i=1}^{m} NodeScore(Parents(i) \to i) \tag{5.1}$$

Hillclimbing is the simplest structure learning algorithm. Starting with some initial DAG, a series of changes such as adding an arc, deleting an arc, or reversing an arc are made. The best scoring structure is kept and the process repeats itself until some stopping criterion. One of the problems involve with hillclimbing is getting trapped in local optima. Solutions to this problem include TABU search, multiple restart and simulated annealing.

Optimal Reinsertion is a larger scale search operator than the simple series of modifications done by hillclimbing. Figure 5.2 illustrates the process of Optimal Reinsertion. Starting with a DAG D_{Old}, a target node T is chosen. All arcs pointing towards or away from T are removed, isolating T. The final step involves searching for the best possible way to reinsert T into the DAG. Not all potential reinsertions are possible since a number of constraints, such as acyclicity, must be satisfied.

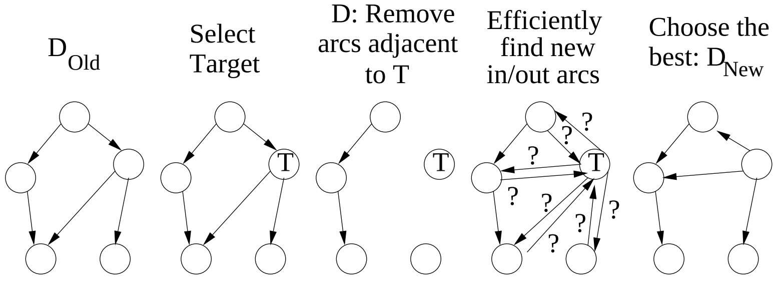

D_Old
Select Target
D: Remove arcs adjacent to T
Efficiently find new in/out arcs
Choose the best: D_New
T
T
T
?

Figure 5.2: Optimal Reinsertion

The outer loop for Optimal Reinsertion generates a random ordering of the nodes in the DAG. The Optimal Reinsertion operation is then called on each node in turn. This process repeats until a complete pass through the random ordering results in no change to the current DAG. However, at an even higher level above, the outer loop is called a fixed number of times. Before each call, a randomly corrupted version of the best DAG found so far is used as the initial structure. This step is used to jolt the algorithm out of any local optima. The best DAG found so far is returned as the result of the structure learning algorithm.

Of course, Optimal Reinsertion is an extremely expensive operator if implemented naively. There are numerous computational tricks and clever data structures that are used to make Optimal Reinsertion an efficient algorithm. First of all, we can speed up the structure search by restricting the number of possible conditional probability tables (CPTs) that we look at. With the $maxParams$ parameter, any conditional probability tables with more than $maxParams$ non-zero entries are ignored. We expect large CPTs to be infrequent since they would incur large penalties in the scoring function. The value of $maxParams$ is typically set to be 100.

One crucial aspect of Optimal Reinsertion is the need for fast calculation of $NodeScore(Parents(i) \rightarrow i)$. By creating a cache for all nodes i and for a large number of possibilities for parents of i, we can access the $NodeScore$ in time independent of the number of records and the number of attributes. In addition, since certain parent combinations for a node i can create CPTs that violate the $maxParams$ restriction, we can reduce the size of the cache by only storing parent and node combinations that respect the $maxParams$ limit. Creating the cache, however, requires computing all the entries once, which if done naively can be expensive. If there are R records in the data set, m attributes, and $maxParams = 2^k$, then searching over all contingency tables of dimension k requires $R\binom{n}{k}$ operations. Our solution to this problem is to use AD-search (Moore & Schneider, 2002) which is an extension of AD-trees (Moore & Lee, 1998). With AD-search, the cost is $R\sum_{j=0}^{k}\lambda^j\binom{m}{j}$. Here, λ is a data set specific quantity that decreases as inter-dependence between attributes increases. Empirically, λ is between 10^{-2} and 10^{-1}.

The final but extremely important optimization deals with finding the optimal set of parents and children for the isolated target node T during Optimal Reinsertion. This step can be intractable due to the extensive number of combinations of parent and children sets. However, we resort to two optimizations in order to make this step feasible. First of all, given a parent set PS, we can cheaply find the optimal children set CS to associate with PS. Secondly, a branch and bound procedure can be used to prune the search for parent sets PS once we can prove that no further parent sets down a search path can improve on the current best score.

We have given a brief overview of the Optimal Reinsertion algorithm in this section. Interested readers are encouraged to refer to (Moore & Wong, 2003) for more details. When compared to conventional hillclimbing, Optimal Reinsertion has been demonstrated in (Moore & Wong, 2003) to be much faster and less local optima prone. Optimal Reinsertion also finds superior Bayesian network structures in terms of the scoring function and in terms of generalization to future data.

5.2.3 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 in 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.

Our solution to new data sources 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 $N_h = \sum_{h=0}^{H} n_h$. 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.

Chapter 6

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 validated our algorithms mainly on simulated data. Two simulators were implemented for the purpose of evaluating WSARE. The first of these simulators was the gridworld simulator, which was a crude Simcity-like world where diseases could be spread between individuals, through restaurants with contaminated food, or through areas containing an aerosolized anthrax release. The performance of WSARE 2.0 was evaluated on data from the gridworld simulator. Since the gridworld simulator did not incorporate any deliberate temporal fluctuations, we used data from a second simulator, which will be called the city Bayesian network (CityBN) simulator, as a testbed for WSARE 3.0. The CityBN simulator was based on two Bayesian networks that introduced temporal fluctuations based on a variety of factors.

In addition to simulated data, we had two sources of real world data. The DARPA

2003 challenge data set (Buckeridge et al., 2004) contained actual ED and pharmacy sales data from five major cities. The outbreak periods were labelled by a team of epidemiologists and a scoring mechanism for detection algorithms was developed. We report the results of WSARE on this challenge data, even though the official evaluation criteria for this data was inappropriate for WSARE, as will be discussed later. We also ran WSARE on ED data from an actual city. Due to the fact that epidemiologists had not analyzed this data set for known outbreaks, we are only able to provide annotated results from the runs of WSARE.

The final section in this evaluation chapter contains timing results which illustrate the benefits of the computational optimizations we implemented to speed up WSARE. These optimizations include greedy rule search, differential counting, racing, and early p-value calculation cutoff.

6.1 The Gridworld Simulator

6.1.1 Simulator Description

The gridworld simulator is intended to simulate (to a first approximation) the effects of an epidemic on a population. The world in this simulator consists of a spatial grid in which there are three types of objects – places, people, and diseases. These three objects interact with each other in a daily routine for a fixed number of days. Each of these objects will be described in detail below.

Places The three types of places in the simulator include homes, businesses, and restaurants. Their roles are evident from what they represent in real life. People reside in homes, work in businesses and eat in restaurants.

People Each person in the simulation has a specified gender and age. Genders for the population are distributed uniformly between male and female while ages follow a normal distribution with mean 40 and standard deviation of 15. People have a home location, a work location, a list of restaurants that they eat at and a list of homes of friends that they like to visit. The locations of work, restaurants, and friends' homes are chosen to be in close proximity to a person's home. On each day, a schedule is randomly generated for each person. In this schedule, people sleep at home until it is time to go to work. They go to work, stop for a lunch break at a restaurant, and then return to work. After work, they spend some time at home before going to a restaurant for dinner. Following dinner, they visit a random selection of friends at their houses. Finally they return home to sleep.

Diseases Diseases are the most complex objects in the simulator as they are designed to allow the creation of a large variety of disease models. People, places and grid cells can all serve as infection agents since they can all carry a disease. With infected places, we can create diseases that spread by a contaminated food supply while with infected grid cells, we can model airborne infections. Associated with each disease is a spontaneous generation probability which corresponds to how likely the disease is to appear in the population at each timestep. Typically, this probability is extremely small. Each disease also progresses through several stages at different rates. At each stage, the infected person can exhibit a variety of symptoms. The current simulation chooses randomly from a list of symptoms at each stage of the disease. At the final stage, an infected agent can either recover or die. The deceased are removed from the simulation.

The entire infection process revolves around the infection probability, which controls how easily an infected person or place can transmit the disease to another person on each timestep. A radius parameter determines how close a person needs to be to catch the disease. The simulator only allows a person to have one disease at a time. Should more than one disease infect a person, the priority of an epidemic arbitrates which disease is assigned to the person. Diseases can be designed to spread from one particular type of agent to another, for example place to person, person to person, or grid cell to person. Additionally, each disease has a specific demographic group that it infects. Whenever it has an opportunity to spread to a person outside of this demographic group, the infection probability is reduced to a small percentage of its original value.

We do not have hospitals in the simulation. Instead, when people exhibit a certain symptom, we create an ED case by adding an entry to a log file. This setup assumes that everyone who develops this monitored symptom will go to an ED. The log file entry contains information such as the person ID, the day, the time, the current location of the person, the home location of the person, and any demographic information about the individual. Most importantly, we add to each entry the actual disease carried by that person, though this last piece of information is hidden from the detection algorithm.

6.1.2 Simulation Settings

Our results were obtained by running the simulator on a 50 by 50 grid world with 1000 people, 350 homes, 200 businesses, and 100 restaurants. The simulation ran for 180 simulated days with the epidemic being introduced into the environment on the 90th day. There were nine background diseases that spontaneously appeared at random points in the simulation. At certain stages, these background diseases caused infected people to display the monitored symptom. These background diseases had low infection probabilities as they were intended to provide a baseline for the number of ED cases. The epidemic, on the other hand, had a higher priority than the background diseases and it had a relatively high infection probability, making it spread easily through its target demographic group.

The epidemic that we added to the system will be referred to as Epidemic0. This disease had a target demographic group of males in their 50s. Additionally, the disease was permitted to contaminate places. Epidemic0 had 4 stages with each stage lasting for two days. The disease was contagious during all four stages. At the final stage, we allowed the person to recover instead of dying in order to keep the total number of people in the simulation constant. Epidemic0 also exhibited the monitored symptom with probability 0.33 on the third stage, probability 1.0 on the final stage, and probability 0 on all other stages.

6.1.3 Evaluation of performance

We evaluated WSARE 2.0 as if it ran once every day. Thus, for each day in the simulated data, we ran WSARE 2.0. We compare the performance of WSARE 2.0 to that of a control chart detection algorithm that considered a day as anomalous when the daily count of ED cases for the monitored symptom exceeded the upper control limit (UCL). The control chart algorithm was chosen because it is a standard detection algorithm used in many fields. The control chart was allowed to train on the ED case data from day 30 to day 56 in the simulation to obtain the mean μ and variance σ^2. The UCL was calculated by Equation 6.1.3, in which Φ^{-1} is the inverse to the cumulative distribution function of a standard normal.

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

In order to illustrate the control chart algorithm, suppose we train on the data from day 30 to 89. The mean and variance of the daily counts of the monitored symptom on this training set are determined to be 20 and 8 respectively. Given an alarm threshold of 0.05, we calculate the UCL as $20 + 1.96 * \sqrt{8} = 25.54$. After training, the control chart algorithm is run over all the days of data from day 56 to day 179. Any day in which the daily count of the particular symptom exceeds 25.54 is considered to contain anomalous events.

We also ran two other univariate detection algorithms for comparison purposes. The first of these algorithms was a moving average algorithm that makes a forecast based on the weighted average of the aggregate counts from the last seven days. This moving average algorithm was selected because it is a common algorithm for analyzing univariate time series data. The other algorithm was an ANOVA regression which simply performed a linear regression on the aggregate daily counts and added the count from yesterday as an additional covariate. We chose to use an ANOVA regression because it is a simple way to account for recent trends in the data.

For WSARE 2.0, if the p-value for the best scoring rule on a particular day was lower than the alarm threshold, then an alarm was raised. Due to the fact that WSARE 2.0 relies on data from 5 to 8 weeks prior to the current day, it detected outbreaks starting from day 56 until day 179. WSARE 2.0 estimated p-values through 400 iterations of the randomization test.

In order to evaluate the performance of the algorithms, we plot an Activity Monitoring Operating Characteristic (AMOC) curve (Fawcett & Provost, 1999), which is similar to an ROC curve. On an 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.

In the gridworld simulation, a false positive was considered to be any alarm raised before day 90. Detection time was considered to be the number of days after the outbreak began ie. day 90, that the first alarm was raised. If no alarm was raised after the start of the outbreak, we set the detection time to 14 days.

6.1.4 Results from Simulated Data

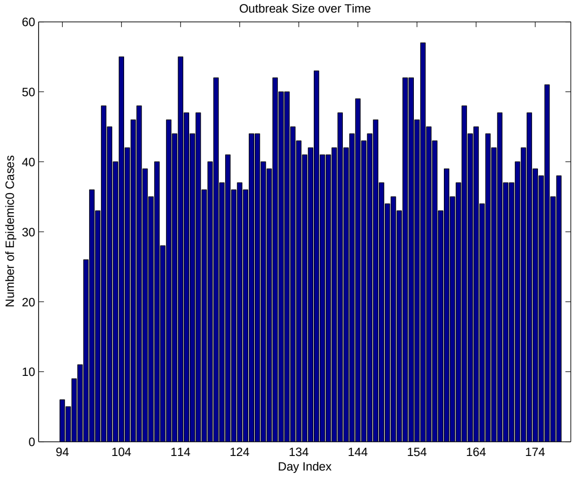


Figure 6.1: The number of Epidemic0 cases over time in the gridworld data

Figure 6.1 plots the number of cases of Epidemic0 per day after the disease is added to the simulator for one of the 100 data sets generated by the simulator. This plot is representative of the nature of the outbreak in the other data sets. As Figure 6.1 shows, the disease was a large scale epidemic which infected a large portion of the population. The challenge with the simulated data was not to detect the epidemic, which would be an easy task in most cases, but to detect the epidemic as early as possible.

Figures 6.2 and 6.3 plot the AMOC curves for all of the algorithms. In Figure 6.2, the error bars for detection time and false positives are shown over a range of alarm thresholds starting at 0 and ending at 0.2 with an increment of 0.025. Figure 6.3 plots the AMOC curves for both algorithms over the same alarm threshold range but with an increment of 0.001. This graph also fills in the lines to illustrate the asymptotic behavior of the curves. The values obtained for both graphs were generated by taking the average over 100 runs of the simulation. These results indicate that WSARE 2.0 significantly outperformed the other univariate algorithms over the alarm threshold range. At a rate of two false positives a month, WSARE 2.0 possesses almost a full two days advantage in detection time over the best competitor. With a sensitive alarm threshold of 0.199, which would correspond to the lower right hand side of the AMOC curve, WSARE 2.0 was able to detect all of the outbreaks. At the upper left hand side of the AMOC curve, WSARE 2.0 missed 43 of the 100 outbreaks when a highly sensitive alarm threshold of 0.001 is used.

Figure 6.4 is a plot of sensitivity versus (1 - specificity) for our results. Producing an ROC curve requires an explanation of the criteria for true positives, false positives, true negatives and false negatives. For each of the 100 data sets, we have the date of the outbreak along with its duration, where the duration is defined as the number of consecutive days after the outbreak in which at least one reported event is due to anthrax. False positives are defined as any alerts raised before the outbreak date, where an alert is a p-value that is less than or equal to the alarm threshold. True negatives are the number of non-alerts before the outbreak date. Before considering false negatives and true positives, we need to define the outbreak period. If the outbreak begins on day i, then the end of the outbreak is on day $i + duration$. Let the outbreak period be $[i, i + duration)$. The number of true positives is 1 if an alert is raised during the outbreak period and 0 if the outbreak is not detected. Finally, we count false negatives as the number of days beginning with the outbreak date and ending on the first alert during the outbreak period. If the outbreak is not detected, the number of false negatives is set to be the outbreak duration.

The other algorithms performed slightly better than WSARE when (1-specificity) was less than approximately 0.03. This behavior occurred because the threshold for these algorithms was extremely conservative, resulting in few alarms being raised. On the other hand, this conservative behavior produced longer detection times, as can

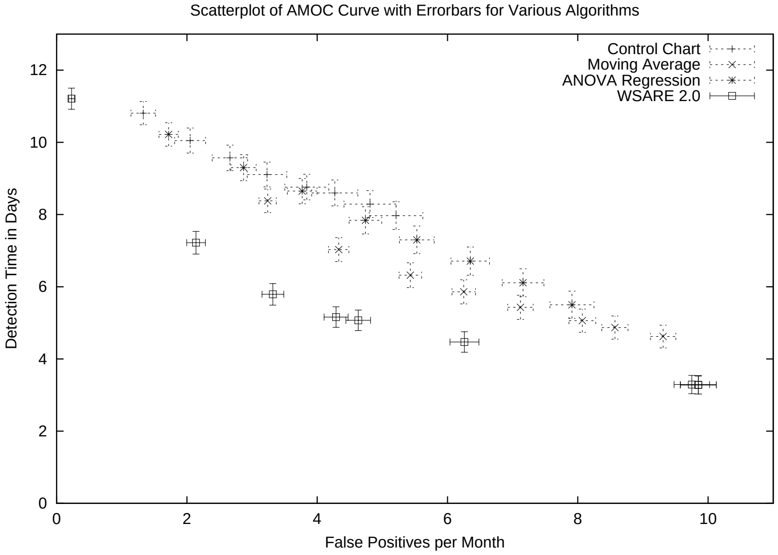


Figure 6.2: AMOC curves with errorbars for gridworld data

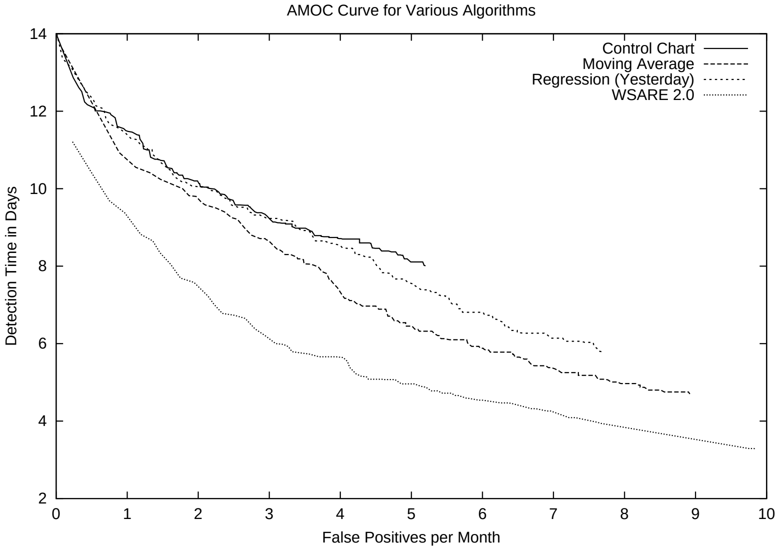


Figure 6.3: AMOC curves for gridworld data

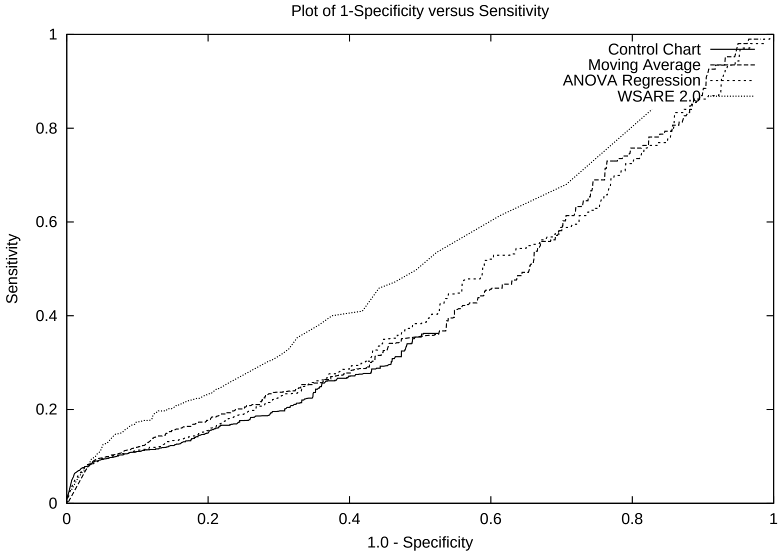


Figure 6.4: ROC curves for gridworld data

be seen in either the AMOC curve, or more false negatives, which resulted in a lower sensitivity overall.

Figures 6.5 illustrate the effects of varying the maximum number of components for a rule from 1 to 3. As the number of components in a rule increases, WSARE 2.0 can be more expressive in characterizing anomalous patterns. On the other hand, WSARE 2.0 also guards against overfitting by requiring each component added to be 95% significant for the two hypothesis tests performed in Section 4.2.5. This criteria makes the addition of components highly unlikely. We expected the performance of WSARE 2.0 to peak around 2 or 3 rules. From Figure 6.5, there was no significant difference between all three variations on data from the gridworld simulation. In fact, the graph for the maximum number of rule components being 2 is not visible since the graph for a maximum number of components equal to 3 is overlaid on top of it. We also explored the difference between greedy rule search and an exhaustive rule search. Figure 6.6 depicts no significant difference between the two types of rule search. We naturally prefer greedy search because it is much faster than exhaustive search.

Another experiment involved increasing the number of iterations of the randomization test from 400 to 999. The results in Figure 6.7 indicate that this too had little difference in the performance of the algorithm. We were also curious as to the effects of changing the scoring function from Fisher's exact test to Informa-

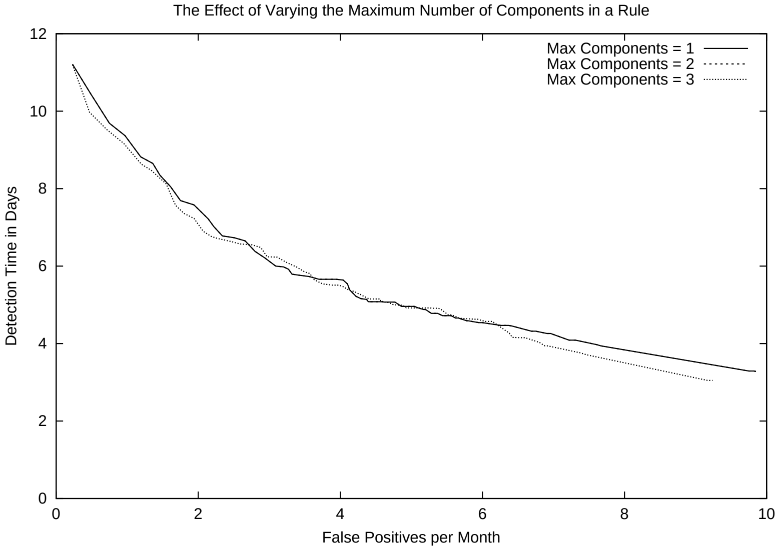


Figure 6.5: The effect of varying the maximum number of components for a rule on the AMOC curve for gridworld data

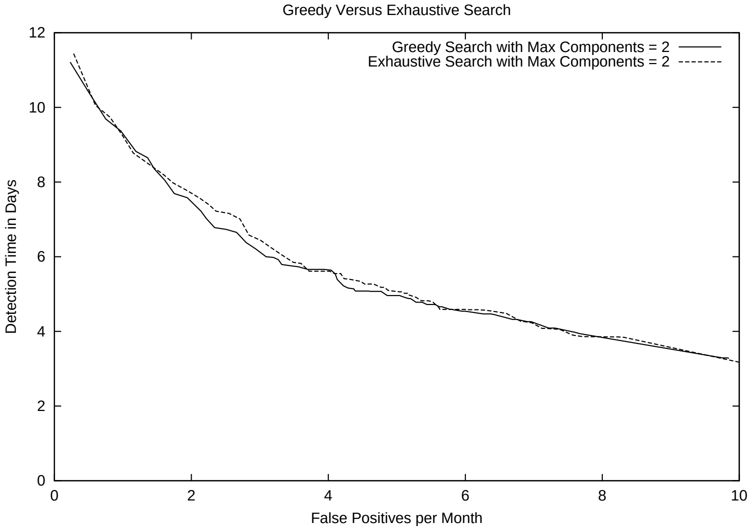


Figure 6.6: AMOC curves for greedy versus exhaustive search on gridworld data

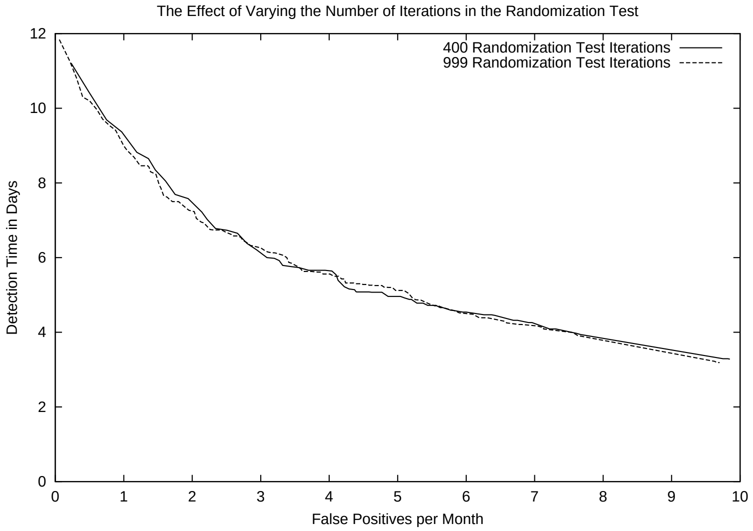


Figure 6.7: The effect of varying the number of iterations in the randomization test on an AMOC curve on gridworld data

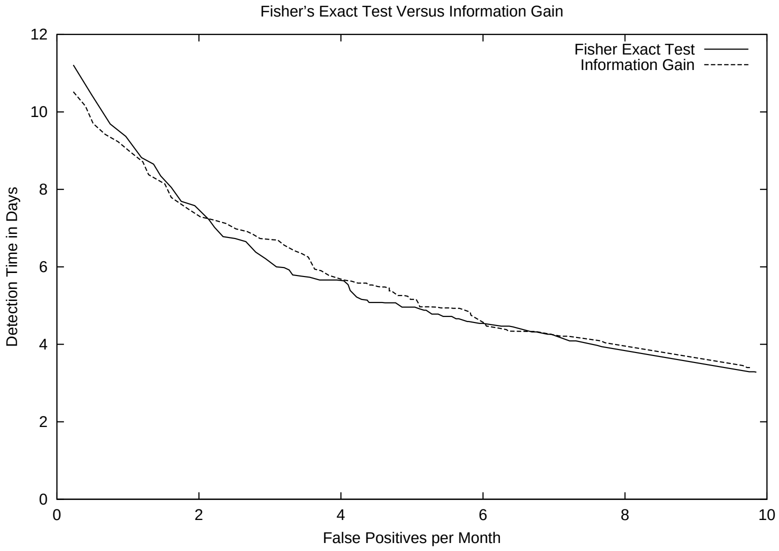


Figure 6.8: AMOC curves for Fisher's exact test versus information gain for gridworld data

tion Gain. Information Gain is commonly associated with decision trees and is defined as $IG(Y|X) = H(Y) - H(Y|X)$, where $H(Y)$ is the entropy of Y and $H(Y|X)$ is the entropy of Y given X. If we let Y = Recent or Baseline and X = Rule Matches or Rule Doesn't Match, we can use Information Gain to tell us how knowing the rule can also help us predict whether the data comes from the recent period or the baseline period. Figure 6.8 indicates that these two scoring functions were nearly identical. We expected this to be the case since the most critical aspect of a scoring function is that it can handle small counts in the cells of the 2-by-2 contingency table and both functions are able to do this.

We now show some of the rules learned by WSARE 2.0. The rules below were obtained from one of the 100 runs of WSARE 2.0 on the simulated data.

```
### Rule 1: Sat Day97 (daynum 97, dayindex 97)
SCORE = -0.00000011 PVALUE = 0.00249875
 33.33% ( 16/ 48) of today's cases have Age Decile = 5 and Gender = Male
  3.85% (  7/182) of other cases have Age Decile = 5 and Gender = Male

### Rule 2: Tue Day100 (daynum 100, dayindex 100)
SCORE = -0.00001093  PVALUE = 0.02698651
 30.19% ( 16/ 53) of today's cases have Age Decile = 5 and Col2 less than 25
  6.19% ( 12/194) of   other cases have Age Decile = 5 and Col2 less than 25
```

In rule 1, WSARE demonstrates that it was capable of finding the target demographic group that Epidemic0 infects. This rule proved to be significant above the 99% level. On the other hand, Rule 2 discovered something that was not deliberately hardcoded into Epidemic0. Rule 2 states that on Day 100, there was an unusually large number of cases involving people in their fifties that were all in the left half of the grid. Since we designed the people in the simulation to interact with places that were in close geographic proximity to their homes, we suspect that the locality of interaction of infected individuals would form some spatial clusters of ED cases. Upon further inspection of the log files, we discovered that 12 of the 16 cases from the current day that satisfied this rule were in fact caused by Epidemic0. This example illustrates the capability of WSARE to detect anomalous patterns that were unexpected.

6.2 The Bayesian Network Simulator

6.2.1 Simulator Description

We evaluated WSARE 3.0 using a simulator based on two Bayesian networks, which we will refer to as the CityBN simulator. The city in this simulator consisted of nine regions, each of which contained a different sized population, ranging from 100 people in the smallest area to 600 people in the largest section, as shown in Table 6.1. We ran the simulation for a two year period starting from January 1, 2002 to December 31, 2003. The environment of the city was not static, with weather, flu levels and food conditions in the city changing from day to day. Flu levels were typically low in the spring and summer but started to climb during the fall. We made flu season strike in winter, resulting in the highest flu levels during the year. Weather, which only took on the values of hot or cold, was as expected for the four seasons, with the additional feature that it had a good chance of remaining the same as it was yesterday. Each region had a food condition of good or bad. A bad food condition facilitated 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 6.1: The geographic regions in the CityBN simulator with their populations in parentheses

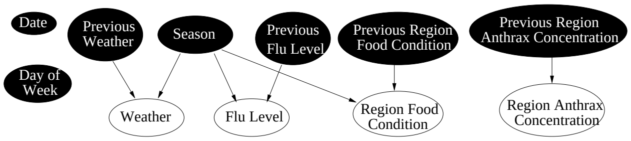

Date
Previous Weather
Season
Previous Flu Level
Previous Region Food Condition
Previous Region Anthrax Concentration
Day of Week
Weather
Flu Level
Region Food Condition
Region Anthrax Concentration

Figure 6.9: City Status Bayesian Network

We implemented this city simulation using a Bayesian network, as shown in Figure 6.9. We 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 had their values set, the values for the white nodes were sampled from the Bayesian network. These records were 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 was chosen randomly for the location of the simulated release. On the date of the release, the Region Anthrax Concentration node was set to have the value of High. The anthrax concentration remained high for the affected region for a randomly chosen length of time.

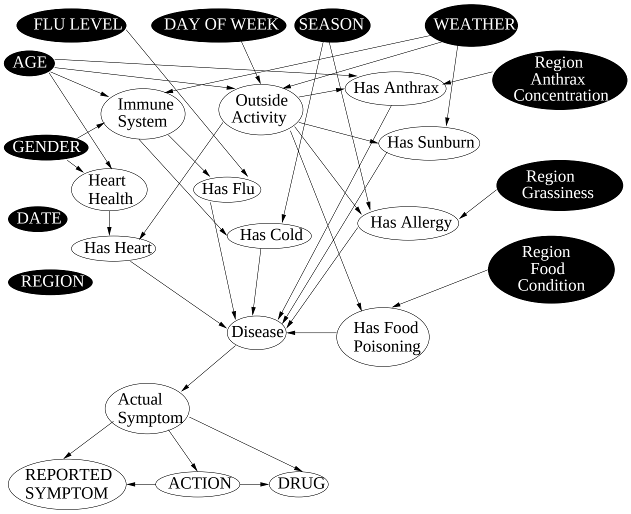


Figure 6.10: Patient Status Bayesian Network

The second Bayesian network used in our simulation produced individual health care cases. Figure 6.10 depicts the Patient Status (PS) network. On each day, for each person in each region, we sampled 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 were then sampled from the PS network. Each individual's health profile for the day was thus generated. The disease node indicated the status of each person in the simulation. A person was either healthy or they could have, in increasing order of precedence, allergies, the cold, sunburn, the flu, food poisoning, heart problems or anthrax. If an individual had more than one disease, the disease

Table 6.2: 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

node picked the disease with the highest precedence. A sick individual then exhibited 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 was 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 were small, ranging from two to four values. If the patient performed any action other than doing nothing, the patient's health-care case was added to the PS data set. Only the attributes in figure 6.10 labelled with uppercase letters were 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 was typically in the range of 30 to 50 records. Table 6.2 contains two examples of records in the PS data set.

6.2.2 Algorithms

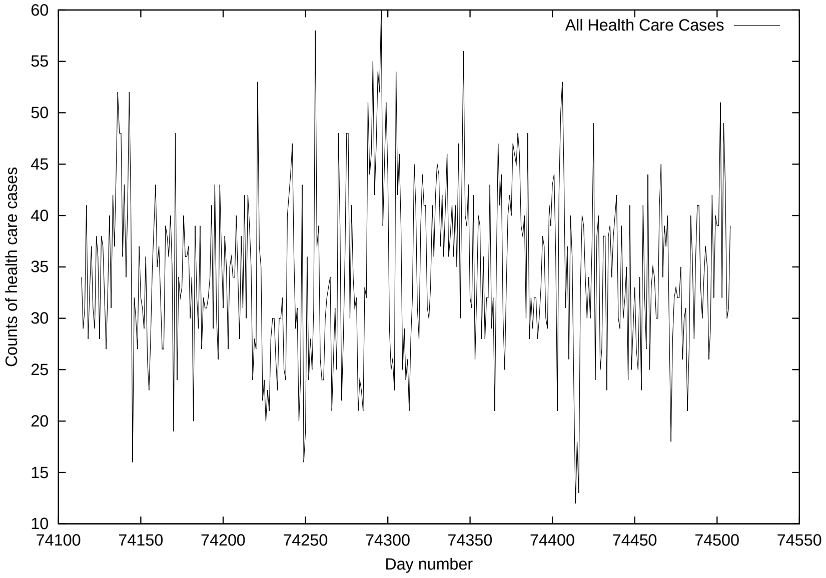


Figure 6.11: Daily Counts of Health-Care Data

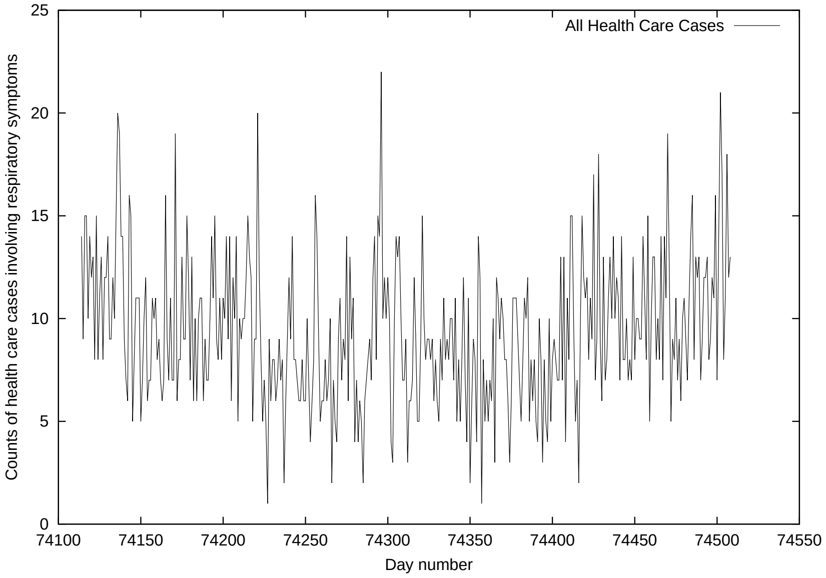


Figure 6.12: Daily Counts of Health-Care Data Involving Respiratory Symptoms

We ran six 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 were not trivially detectable. Figure 6.11 plots the total count of health-care cases on each day during the evaluation period while Figure 6.12 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 occurred on day index 74409, which is clearly not the highest peak in both graphs. Occasionally the anthrax releases affected such a limited number of people that it was virtually undetectable. Consequently, we only used data sets with more than eight reported anthrax cases on any day during the attack period.

The following paragraphs will describe the six detection algorithms that we ran on the data sets.

The Control Chart Algorithm The first algorithm used was the control chart algorithm described in Section 6.1.3. We used a training period of January 1, 2002 to December 31, 2002. Two different types of univariate data were used by this algorithm – one data set was composed of total daily counts and the other on daily counts of cases involving respiratory symptoms.

Moving Average Algorithm We also show the results of using a simple 7 day window moving average algorithm. As with the control chart algorithm, we ran this algorithm on total daily counts and also on daily counts of cases with respiratory symptoms.

ANOVA Regression ANOVA regression is a fairly powerful detector when temporal trends are present in the data, as was shown in the DARPA Challenge (Buckeridge et al., 2004). We used 6 covariates for the days of the week, 3 for the seasons and one for the daily aggregate count from the previous day. As before, this algorithm was run on two versions of the data.

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 was 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 built 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 had 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 was possible that no such records existed in the past with exactly this combination of environmental attributes. If there were fewer than five records in the past that match, WSARE 2.5 could not make an informed decision when comparing the current day to the baseline and simply reported nothing for the current day.

WSARE 3.0 WSARE 3.0 used the same environmental attributes as WSARE 2.5 but built a Bayesian network for all data from January 1, 2002 to the beginning of the current day. 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 allowed WSARE 3.0 to learn a network structure from scratch once every 30 days. On intermediate days, WSARE 3.0 simply updated the parameters of the previously learned network without altering its structure. An example of a Bayesian network learned from the CityBN simulator data is shown in Appendix A along with the contingency tables for each node.

6.2.3 Results

Our evaluation criteria examines the algorithms' performance in terms of detection time versus false positives over alarm thresholds ranging from 0 to 0.2. The lower alarm thresholds yield lower false positives and higher detection times while the converse is true with higher thresholds. Figures 6.13 to 6.15 are plots of the AMOC curves with error bars for the algorithms, using alarm threshold increments of 0.025. In order to avoid too much clutter on each graph, we split the plots up into three graphs. We also do not plot the Moving Average algorithm on Figure 6.13 to avoid clutter and because it performs significantly worse than all the other algorithms.

In order to display the asymptotic behavior of the algorithms, Figures 6.16 to 6.18 plot the AMOC curve with alarm threshold increments of 0.001 and connects the plot points. The optimal detection time was one day, as shown by the dotted line at the bottom of the graph. We added 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 was treated as a false positive. Detection time was calculated as the first alert raised after the release date. If no alerts were raised after the release, the detection time was set to 14 days. We also present a ROC curve for all the algorithms in Figures 6.19 to 6.21.

From Figures 6.13 through 6.21, WSARE 2.5 and WSARE 3.0 outperformed 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 were 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 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 outperformed 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 was 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 were 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

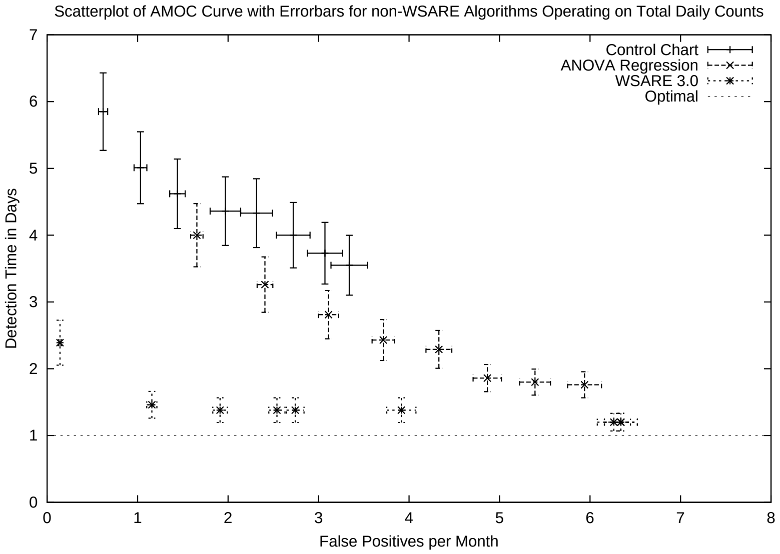


Figure 6.13: AMOC curves with error bars comparing WSARE 3.0 to non-WSARE algorithms operating on total daily counts of cases from the CityBN simulator

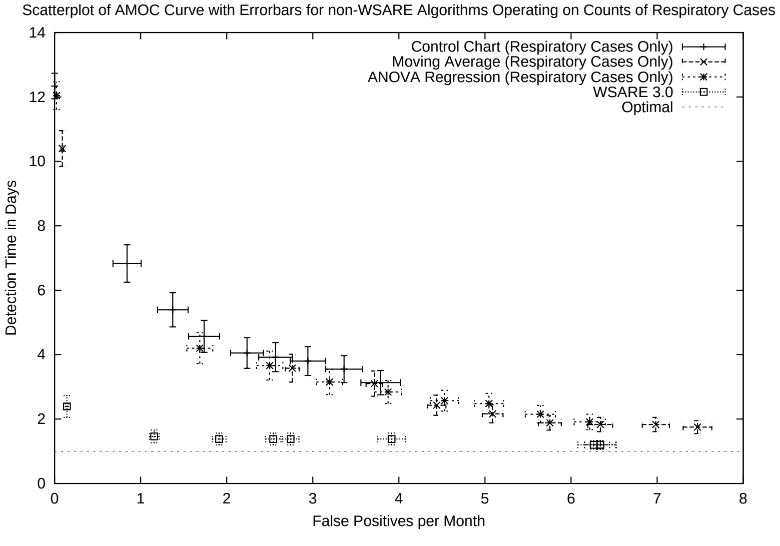


Figure 6.14: AMOC curves with error bars comparing WSARE 3.0 to non-WSARE algorithms operating on counts of respiratory cases from the CityBN simulator

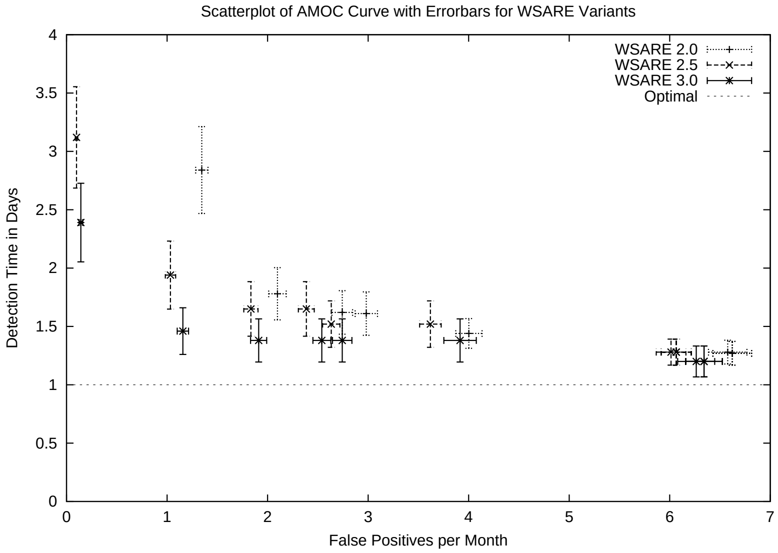


Figure 6.15: AMOC curves with error bars for WSARE variants operating on CityBN data

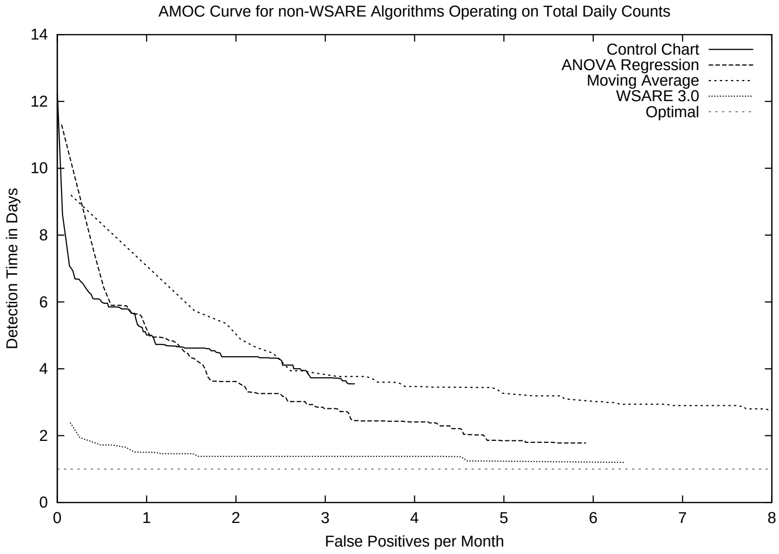


Figure 6.16: AMOC curves comparing WSARE 3.0 to non-WSARE algorithms operating on total daily counts from the CityBN simulator

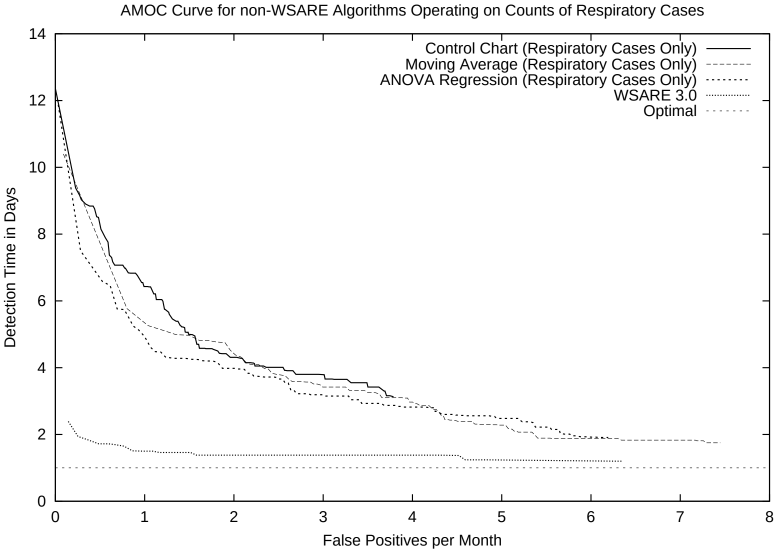


Figure 6.17: AMOC curves comparing WSARE 3.0 to non-WSARE algorithms operating on data with respiratory symptoms only from the CityBN simulator

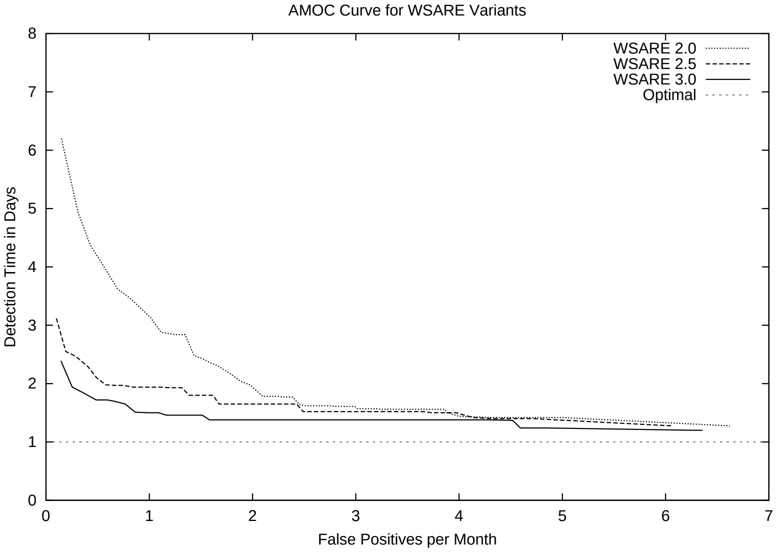


Figure 6.18: AMOC curves for WSARE variants operating on CityBN data

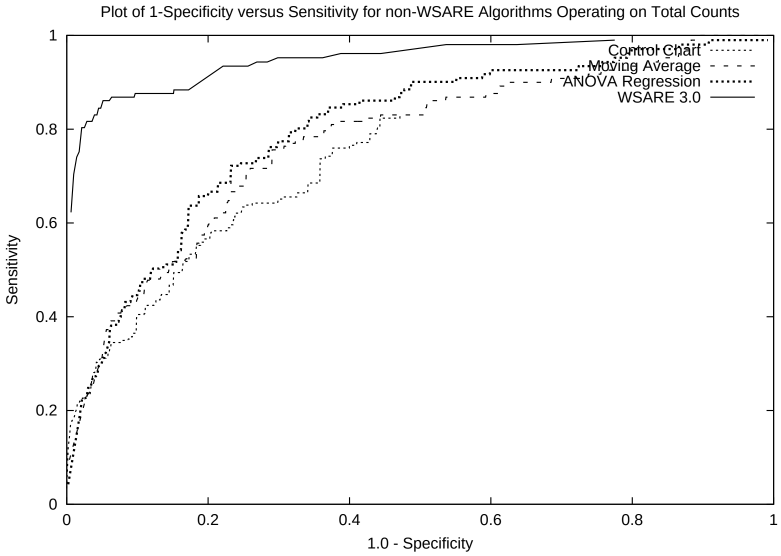


Figure 6.19: ROC curves comparing WSARE 3.0 to non-WSARE algorithms working on total daily counts from the CityBN simulator

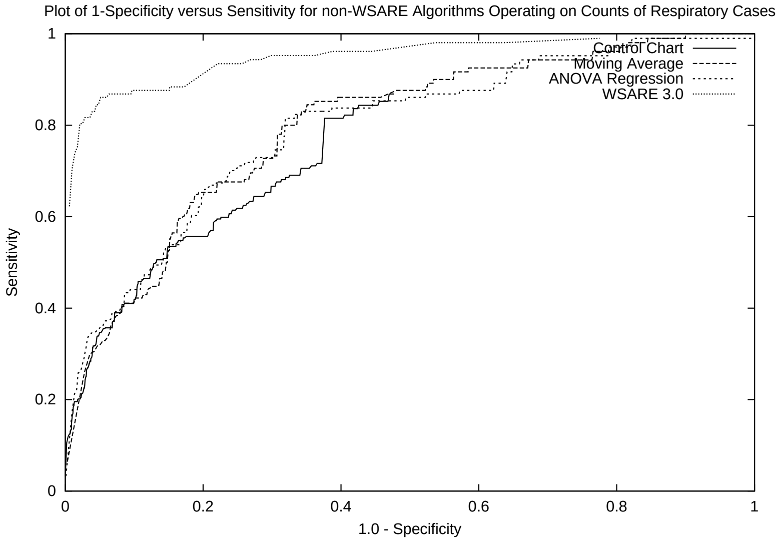


Figure 6.20: ROC curves comparing WSARE 3.0 to non-WSARE algorithms working on counts of respiratory cases from the CityBN simulator

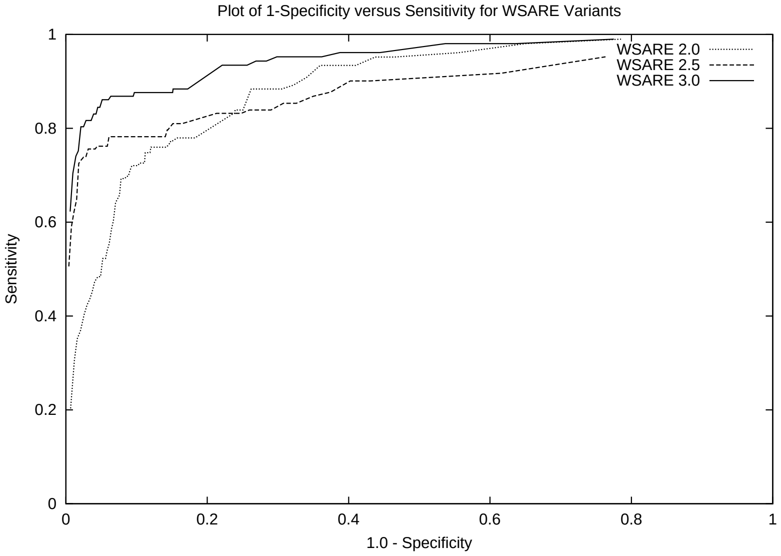


Figure 6.21: ROC curves for WSARE variants operating on CityBN data

to the Bayesian network, this additional information would likely have reduced the false positive count.

Figures 6.24 to 6.27 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 were 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 6.24 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 6.25 is a representative plot of a medium scale outbreak that was most common in the data. The particular outbreak shown in Figure 6.25 was also detected by WSARE 3.0 on the first day for an alarm threshold of 0.005. Small scale outbreaks, as shown in Figure 6.26, were the most difficult to detect. WSARE 3.0 detected the outbreak in Figure 6.26 on the third day with a very insentitive alarm threshold of 0.005. Figure 6.27 contains an outbreak that WSARE 3.0 was unable to detect using a sensitive alarm threshold of 0.03.

We also conducted other experiments to determine the effect of varying certain parameters of WSARE 3.0. One of the experiments involved changing the maximum number of parents allowed per node in the Bayesian network. We tried a limit of 0,

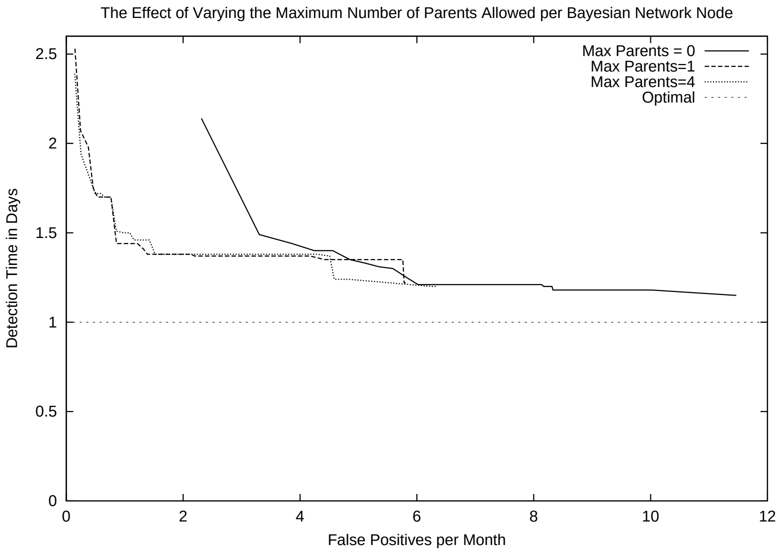


Figure 6.22: The effect on the AMOC curve of varying the maximum number of parents allowed for a Bayesian network node for CityBN data

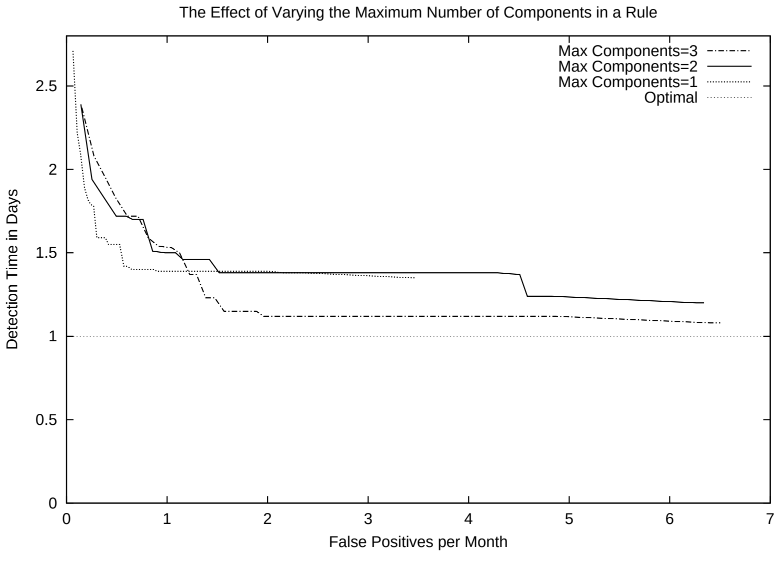


Figure 6.23: The effect of varying the maximum number of components for a rule on the AMOC curve for CityBN data

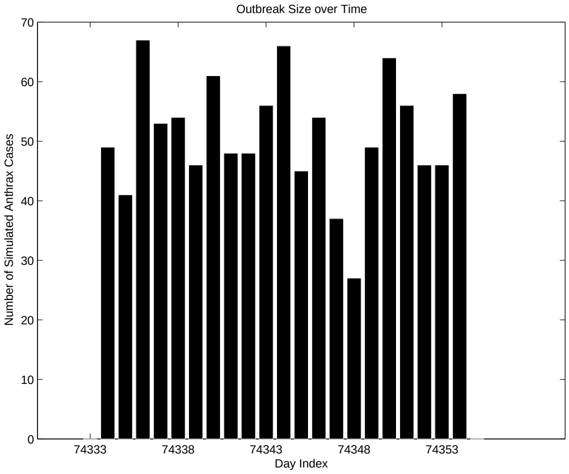


Figure 6.24: An example of a large scale outbreak in the CityBN data

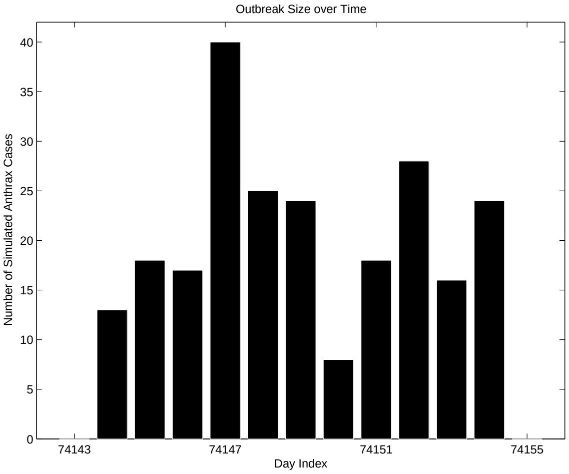


Figure 6.25: An example of a medium scale outbreak in the CityBN data

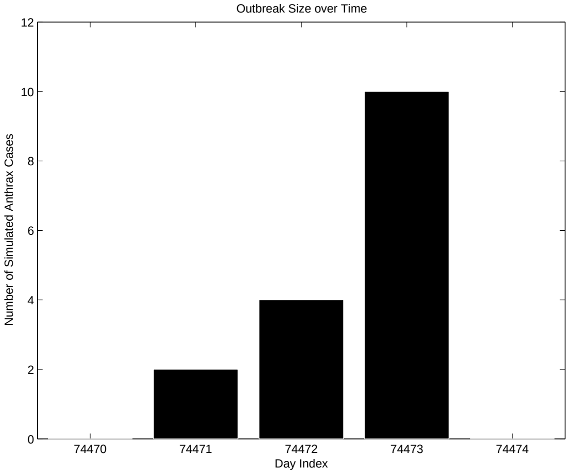


Figure 6.26: An example of a small scale outbreak in the CityBN data

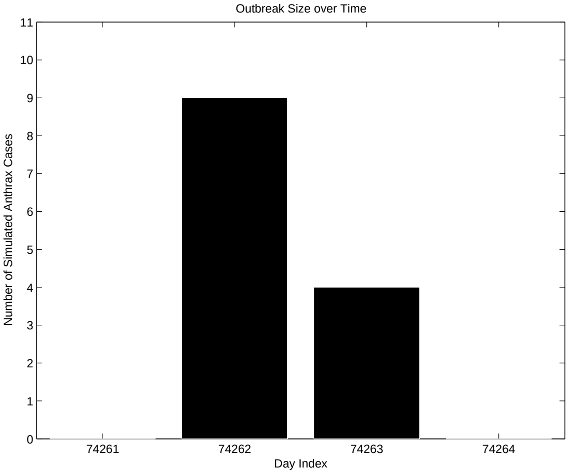


Figure 6.27: An example of an outbreak that was not detected in the CityBN data by WSARE 3.0 with an alarm threshold of 0.03

resulting in a naive, disconnected Bayesian network and a limit of 1, corresponding to a tree structure for the network. The AMOC curves in Figure 6.22 compared the results of these experiments with the default limit of 4. The graph for a network with no parents allowed per node was shifted to the right, indicating that for a given alarm threshold, it produced about two more false positives per month. This behavior was expected since a disconnected network has poor generalization capabilities due to its large bias. The tree Bayesian network and the network with a limit of 4 parents per node demonstrated similar results because the network structures learned from the CityBN simulator data were generally simple, with one parent per node in almost all the structures.

Another experiment involved varying the maximum components allowed per rule from 1 to 3. The variations did 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 3 component limit outperformed the other two variations. As was mentioned before, a 3 component rule is capable of being more expressive in its description of anomalous patterns. Therefore, it should be more successful at finding these patterns and improving the detection time. In addition to varying the number of components per rule, we once again evaluated the performance of greedy versus exhaustive search. As in the gridworld simulator, there seemed to be no significant difference between the two as seen on the AMOC curve in Figure 6.28.

6.3 DARPA Challenge

The challenge data from DARPA included three data sets from five major cities in the US. These three data sets consisted of actual civilian and military ED data as well as pharmacies sales data, with patient confidentiality preserved. These data sets were all taken from the same time period. Of course, since this was real world data, there were outbreaks present although we restricted our attention to outbreaks related to respiratory and gastrointestinal syndromes. The outbreak periods in the data were determined by a team of three epidemiologists, who manually examined plots of the daily number of counts from all three data sources to form their conclusions, without any further attempts to determine the ground truth. Detection algorithms were run on the data starting from September 1, 2002 until May 31, 2003. All three sources of data were allowed to be used in order to improve the performance of the algorithms.

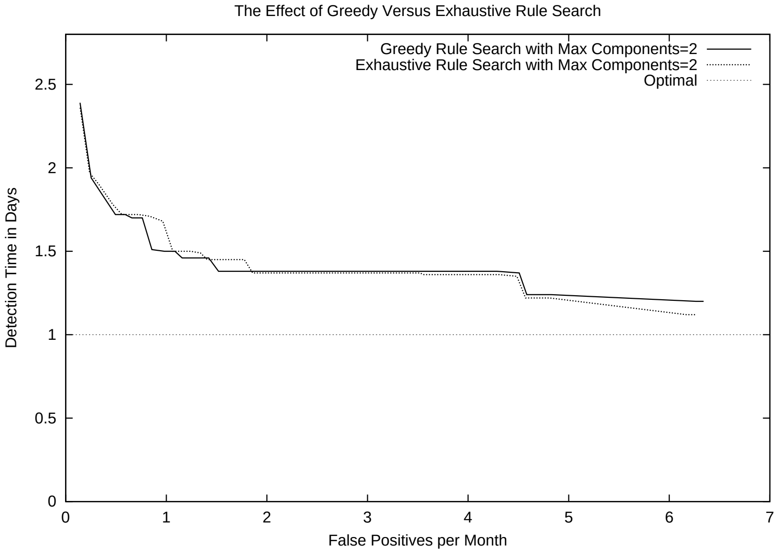


Figure 6.28: AMOC curves for greedy versus exhaustive search for CityBN data

Each algorithm was then scored according to its sensitivity, specificity, and timeliness on the detection of the respiratory and gastrointestinal outbreaks in the data.

Before proceeding, we would like to point out that WSARE is inappropriate for this type of evaluation. The outbreak periods were identified from peaks in a univariate time series plot based on aggregate daily counts. WSARE identifies outbreaks from a totally different perspective. Instead of looking for an absolute change in the total number of cases, WSARE relies on relative changes in ratios of subgroups of the data. For example, WSARE can detect that the fraction of male senior citizens appearing in the ED today is double what it normally is during a baseline period. WSARE cannot detect an absolute change such as the daily total of ED cases reaching a pre-specified threshold. Naturally, we expected WSARE to perform extremely poorly on this data when the scoring criterion was applied since it could monitor the same signal used by the epidemiologists to determine outbreak periods. Furthermore, WSARE may be unfairly penalized for raising too many false positives because the alarms it raised may in fact be public health concerns but they were not validated by the epidemiologists. Nevertheless, we include our results in order to show when WSARE should not be used and also to illustrate some of the problems with WSARE.

WSARE 3.0 was run on only the ED data using the attributes of age, gender, season and day of week. The pharmacy data was not used because we felt that it raised too many alerts. We used season and day of week as environmental attributes. The results are summarized in Tables 6.3 and 6.4 using the scoring techniques provided by the DARPA project. For comparison purposes, we include the results of an ANOVA regression algorithm that performed well on the challenge.

False Alarm Rate	**Regression RESP**	**Regression GI**	**WSARE RESP**	**WSARE GI**
1 per 2 weeks	4.125	2.8	N/A	17.0
1 per 4 weeks	6.625	19.67	N/A	N/A
1 per 6 weeks	7.714	19.67	N/A	N/A

Table 6.3: Timeliness in Days as a Function of Specificity

False Alarm Rate	**Regression RESP**	**Regression GI**	**WSARE RESP**	**WSARE GI**
1 per 2 weeks	8/8	5/7	1/8	4/7
1 per 4 weeks	8/8	4/7	1/8	2/7
1 per 6 weeks	7/8	4/7	0/8	0/7

Table 6.4: Sensitivity (number of outbreaks detected / total number of outbreaks) as a function of Specificity

From the results, WSARE raised too many alarms in order for it to detect the outbreaks with a low false positive rate. WSARE highlighted significant upswings in the ratios of cases involving certain age groups or genders. As was previously stated, some of these were penalized as false positives but they may in fact be important to public health practioners except we have no way of knowing since the epidemiologists that scored the data only looked at aggregate daily counts.

The high false positive count was also be due to characteristics of the data that illustrate some of the problems with WSARE. One of the assumptions with WSARE 3.0 is that given the environmental attributes, the ratios over the baseline period should be reasonably stable since environmental attributes account for most of the variation in the data. There were times, however, when latent factors were responsible for non-stationarity in the data. The challenge data was extremely non-stationary, even when environmental attributes such as the season and day of week were taken into account. Figure 6.29 contains a plot over fall Thursdays of the ratio of the number of patients over the age of 71 to the total daily number of patients. The big gaps in the plot are due to the fact that we are only plotting the ratios for fall Thursdays and we obviously have no such data for winter, spring and summer. Even with day of week and seasonal effects accounted for, the signal still had large fluctuations. Figure 6.30 illustrates the total number of cases from this group. Due to the large sample sizes involved, these fluctuations were not due to chance. The DARPA challenge data has been analyzed and the source of non-stationarity was due to new sources such as doctors, clinics and hospitals being added over time and also because of existing sources not routinely reporting their data during the surveillance period.

When faced with this degree of non-stationarity, a different baseline approach from that of WSARE 3.0 should be taken. Instead of building the baseline from a Bayesian network learned from all historical data, a baseline computed from more recent data should be used. Of course, using more recent data results limits the amount of data available for the baseline and when data becomes scarce, an accurate Bayesian network can be almost impossible to learn from this limited data. One possible way to alleviate some of the non-stationarity using the WSARE framework is to add an extra environmental variable that summarizes the count from the previous day, thereby allowing WSARE to account for some recent information.

Determining the time intervals for cyclical environmental attributes is also another difficulty with WSARE 3.0. In the challenge data, there were strong day of week trends which we were able to account for accurately with an environmental attribute. The seasonal trends, however, were much more difficult since they did not line up with either monthly time intervals or seasonal time intervals. Moreover, the seasonal trend boundaries moved around from year to year. Our best approximation to these trend time intervals was to use the seasons. In general, WSARE 3.0 is unable to determine autonomously the best set of boundaries for these time environmental attributes and a substantial degree of tuning by the user is needed.

One final problem is the lack of baseline smoothing in WSARE. In the challenge data, some data sets had multiple outbreaks contained in them. Furthermore, the training data contained outbreaks that were not identified. WSARE 3.0 incorporated these outbreaks into the baseline without smoothing out the outliers, resulting in an inaccurate baseline. When running WSARE 3.0 on a daily basis, we assume that when an outbreak occurs, we would be notified at some point during its duration. By being aware of the outbreak, we can then add an extra environmental variable that would allow us to account for this outbreak while still performing further surveillance.

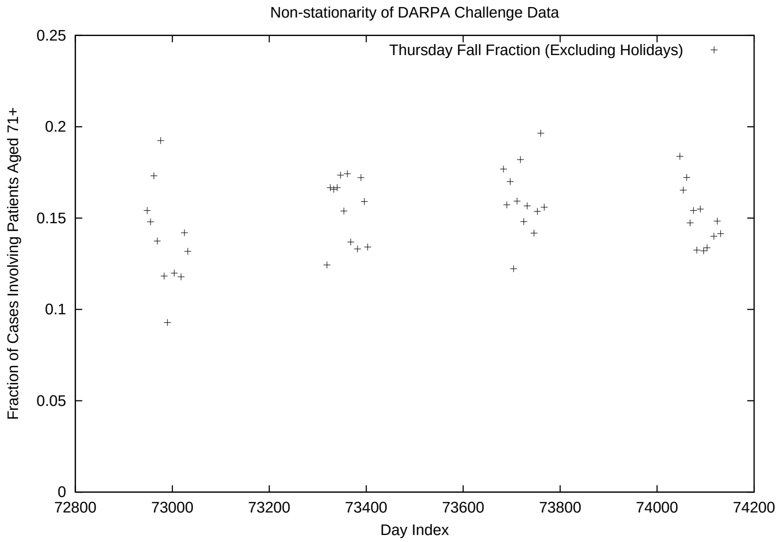


Figure 6.29: The fraction of cases on Thursdays in the fall (excluding holidays) involving patients aged 71 and up in the Seattle ED data

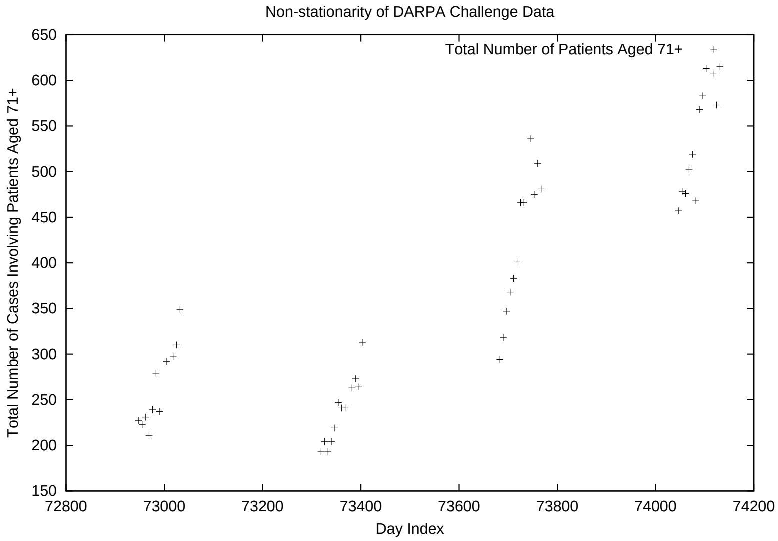


Figure 6.30: The total number of cases on Thursdays in the fall (excluding holidays) involving patients aged 71 and up in the Seattle ED data

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

We also tested 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 included date of admission, coded hospital ID, age decile, gender, syndrome information and both home location and work location on a latitude-longitude grid. This data had a similar phenomenon as the DARPA data in the previous section: new hospitals came online and began submitting data during the time period that the data was collected. We used the solution described in section 5.2.3 to address this problem. WSARE operated 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 was 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 applied 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. Sat 2001-02-13: SCORE = -0.00000004 PVALUE = 0.00000000
  14.80% ( 74/500) of today's cases have Viral Prodrome = True and Encephalitic Prodome = False
  7.42% (742/10000) of baseline have Viral Prodrome = True and Encephalitic Prodrome = False
2. Sat 2001-03-13: SCORE = -0.00000464 PVALUE = 0.00000000
  12.42% ( 58/467) of today's cases have Respiratory Prodrome = True
  6.53% (653/10000) of baseline have Respiratory Prodrome = 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 Prodrome = False
  74.29% (7430/10001) of baseline have Unknown Prodrome = False
5. Thu 2001-12-02: SCORE = -0.00000087 PVALUE = 0.00000000
  14.71% ( 70/476) of today's cases have Viral Prodrome = True and Encephalitic Prodrome = False
  7.89% (789/9999) of baseline have Viral Prodrome = True and Encephalitic Prodrome = 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 Prodrome = True
  2.40% (240/10000) of baseline have Hospital ID = 1 and Viral Prodrome = True
```

Rule 3 was likely due to clerical errors in the data since the rule found 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 data to inspect the text descriptions of the chief complaints for the cases related to these three rules. Rule 2 cases contained 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 had dates in winter, along with the symptoms mentioned, we speculate that this anomalous pattern was likely caused by an influenza strain.

6.5 Annotated Output of WSARE 3.0 on Actual ED Data July 18, 2002 - July 18, 2003

We also evaluated WSARE 3.0 over more recent ED data from July 18, 2002 to July 18, 2003. Unlike the data used in Section 6.4, the majority of cases from July 18, 2002 to July 18, 2003 were missing information about home and work locations. As a result, WSARE could use the spatial attributes for these cases. Only the attributes of date of admission, coded hospital ID, age decile, gender and syndrome information were used, all of which respect patient confidentiality as described in Chapter 2. This data also had new hospitals appearing throughout the surveillance time period. Once again, we used the approach in Section 5.2.3 to deal with this problem. WSARE operated on this data using day of week and month as environmental attributes. Below, we report the significant rules as found by FDR with $\alpha_{FDR} = 0.1$.

```
1. 2002-07-18: SCORE = -3.71674e-07 PVALUE = 0.00249377
  3.52941% (18/510) of today's cases have Age Decile = 4 and Viral Prodrome = True
  1.14% (114/10000) of baseline cases have Age Decile = 4 and Viral Prodrome = True
2. 2002-08-21: SCORE = -3.01133e-06 PVALUE = 0.00249377
  3.10345% (18/580) of today's cases have Age Decile = 8 and Viral Prodrome = True
  1.09% (109/10000) of baseline cases have Age Decile = 8 and Viral Prodrome = True
```

```
3. 2002-08-22: SCORE = -4.15098e-07 PVALUE = 0
  13.9535% (78/559) of today's cases have Viral Prodrome = True and Rash Prodrome = False
  8.10919% (811/10001) of baseline cases have Viral Prodrome = True and Rash Prodrome = False
4. 2002-08-24: SCORE = -8.81048e-06 PVALUE = 0.00997506
  15.4255% (87/564) of today's cases have Viral Prodrome = True
  9.84902% (985/10001) of baseline cases have Viral Prodrome = True
5. 2002-08-30: SCORE = -3.41648e-05 PVALUE = 0.0074813
  12.7886% (72/563) of today's cases have Viral Prodrome = True and Encephalitic Prodrome = False
  8.04% (804/10000) of baseline cases have Viral Prodrome = True and Encephalitic Prodrome = False
6. 2002-08-31: SCORE = -3.70298e-06 PVALUE = 0
  14.7541% (90/610) of today's cases have Viral Prodrome = True
  9.31% (931/10000) of baseline cases have Viral Prodrome = True
7. 2002-09-15: SCORE = -1.33673e-06 PVALUE = 0
  14.711% (84/571) of today's cases have Age Decile = 1
  8.93911% (894/10001) of baseline cases have Age Decile = 1
8. 2002-10-29: SCORE = -2.02341e-106 PVALUE = 0
  14.4128% (81/562) of today's cases have Hospital ID = 1 and Unknown Prodrome = True
  0% (0/10000) of baseline cases have Hospital ID = 1 and Unknown Prodrome = True
```

Rules 8-269 from 2002-10-30 to 2003-07-17 all have Hospital ID = 1 and Unknown Prodrome = True except for Rules 71, 123 and 233 shown below

```
71. 2002-12-31: SCORE = -6.14186e-15 PVALUE = 0
  22.0207% (170/772) of today's cases have Hospital ID = 1 and Unmapped Prodrome = False
  12.6787% (1268/10001) of baseline cases have Hospital ID = 1 and Unmapped Prodrome = False
123. 2003-02-22: SCORE = -1.8728e-19 PVALUE = 0
  18.7721% (159/847) of today's cases have Hospital ID = 1 and Unmapped Prodrome = False
  9.62904% (963/10001) of baseline cases have Hospital ID = 1 and Unmapped Prodrome = False
233. 2003-06-12: SCORE = -2.3117e-30 PVALUE = 0
  17.8707% (141/789) of today's cases have Hospital ID = 1 and Unmapped Prodrome = False
  7.27927% (728/10001) of baseline cases have Hospital ID = 1 and Unmapped Prodrome = False
```

The first six rules indicated an increase in patients with viral symptoms. Although rule 1 seems like an isolated anomalous pattern, rules 2-6 seem to indicate a possible viral outbreak, especially given their temporal proximity. We would like to point out that public health officials may find the age decile groupings, such as those in rules 1 and 2, to be too narrow. A broader categorization such as adult versus child might be more informative. This limitation can be easily removed by simply discretizing the data in a different way.

Rule 7 detected an anomalous upswing in the number of patients between the age of 10 to 19 appearing in the ED. After reviewing the free text chief complaint fields corresponding to these cases, we noticed that the majority of the cases involved head injuries and superficial injuries to shoulders, elbows, wrists, and knees. We would suggest that this increase in cases involving teenage children was due to an incident at a school or perhaps a motor vehicle accident involving a school bus.

Rules 8-269 except for rules 71, 123 and 233 indicated that records from hospital ID 1 had a large upswing in an unknown prodrome. Each case in the ED data set could have fallen into multiple prodrome categories, where a prodrome is a broad grouping for the symptoms exhibited by the patient. These prodrome categories are: diarrhea, viral, respiratory, rash, hemorrhagic, botulinic, encephalitic, other, unmapped, and unknown. The other prodrome was a special category for cases that did not fall into the first seven prodromes while the unmapped prodrome was for symptoms that did not match any of the first eight. The unknown prodrome was for cases without any information about the chief complaint. Prodrome information was represented as 10 binary attributes in the ED data set, where a true value indicates that the case exhibited the symptoms corresponding to that prodrome. After checking the cases corresponding to rules 71, 123 and 233, we noticed that 65-70% of the time, whenever the unmapped prodrome attribute had a value of false, the unknown prodrome attribute for that case also had a value of true. This occurrence was high enough for us to suspect that rules 71, 123 and 233 were identical to rules 8-269. The increase in unknown prodromes at hospital ID 1 was likely due to a change in the categorization of ICD9 codes to prodromes by the RODS system.

6.6 Timing

We also ran tests to measure the speedup gained from greedy rule search, racing, differential counting, and early p-value calculation cutoff. Table 6.5 summarizes the results, sorted in decreasing order of benefit from the optimizations. The row for WSARE 3.0 represents the WSARE algorithm with all of the computational tricks

Algorithm	**Mean duration for each day in seconds**	**95% Confidence Interval**	**Slowdown from original**
WSARE 3.0	14.96	(13.64, 16.29)	N/A
Exhaustive Rule Search	456.14	(408.39,503.90)	30.49
No Racing	71.9	(70.95, 72.77)	4.81
No Differential Counting	23.11	(20.95, 25.27)	1.54
No Early P-value Calculation Cutoff	15.02	(13.69, 16.35)	1.00

Table 6.5: A summary of timing results

in place. For each row in the table above, we removed the stated optimization and measured the time taken to compute the p-value for the best scoring rule on each day. These timing statistics were taken over 365 days on data from the CityBN simulator using WSARE 3.0 in which the Bayesian network structure was learned from scratch every 30 days. A 1.0 Ghz Pentium 4 workstation was used to obtain these results.

The greatest time savings were gained using greedy search instead of exhaustive search. If we have M attributes all with arity K, then there are $O(KM)$ rules to search over using greedy search. Exhaustive search for two component rules is clearly more expensive at a cost of $O(K^2M^2)$ rules to search over. We have shown in sections 6.1.4 and 6.2.3 that the AMOC curves for exhaustive search were almost identical to those of greedy search. Thus, we feel justified in using this approximation.

By avoiding the full set of randomization test iterations, racing sped up WSARE by a factor of nearly 5 times. As was mentioned previously, the racing optimization was frequently used since typical days did not contain outbreaks and had an insignificant p-value. Differential counting improved the speed by a factor of 1.5. This was not as substantial a gain as we would have hoped, likely because of the time involved in setting up the data structures for this optimization. Finally, the speedup from an early cutoff during the p-value calculation was almost negligible.

Chapter 7

Improvements to the Randomization Test

Hypothesis testing is a standard procedure in statistics in which we need to choose between a null hypothesis H_0 and an alternative hypothesis H_1. This decision is based on a test statistic, which is a value calculated from the data that has two distinct probability distributions under the two competing hypotheses. If the test statistic is in the rejection region, we reject the null hypothesis in favor of the alternative hypothesis. Equivalently, we can reject the null hypothesis if the p-value of the test statistic is less than a predetermined significance level α, which is typically set at $\alpha = 0.05$. The p-value is defined as the probability of seeing a value at least as extreme as the observed test statistic under the distribution of the null hypothesis. The significance level α controls the probability of a type I error, which occurs whenever the null hypothesis is rejected even though it is in fact true. Since rejection of the null hypothesis in favor of the alternative is usually considered to be making a discovery, type I errors are called false discoveries or false positives.

For a single hypothesis test, the α value does indeed guarantee that the probability of a type I error is less than or equal to α. However, when m hypothesis tests are performed, we no longer have the same guarantee on the probability of a type I error. For instance, suppose we perform 10000 hypothesis tests with $\alpha = 0.05$. The probability of a false discovery is will be $1 - (1 - 0.05)^{10000} \approx 1 >> \alpha$ (Miller et al., 2001). Another way of describing this problem is as follows: even if we perform 10000 hypothesis tests with $\alpha = 0.05$ on randomly generated data, we would still expect to have 500 “discoveries”.

	# not rejected	# rejected	total
# true null hypotheses	U	V	m_0
# non-true null hypotheses	T	S	m_1
total	W	R	m

Table 7.1: A summary table for the multiple hypothesis testing problem as given in (Ge et al., 2003)

Table 7.1 (Ge et al., 2003) summarizes the multiple hypothesis testing problem. There are several variations on the type I error rate when applied to the multiple hypothesis testing. We will only discuss three of these error rates. The Family-wise error rate (FWER) is defined as the probability of at least one type I Error ie. $Pr(V > 0)$ and is used by the Bonferroni correction (Bonferroni, 1936). The Benjamini and Hochberg FDR procedure (Benjamini & Hochberg, 1995) controls the false discovery rate (FDR), which is the expected ratio of the number of type I errors over the number of times we reject the null hypothesis. FDR is defined as 0 whenever there are no rejections. The formula for FDR is shown in Equation 7.1. Finally, the positive false discovery rate (pFDR) (Storey, 2003) is the FDR given that at least one hypothesis is rejected as shown in Equation 7.2. Readers interested in the difference between 7.1 and 7.2 are referred to (Storey, 2003; Benjamini & Hochberg, 1995).

$$FDR = E\left[\frac{V}{R}|R > 0\right] Pr(R > 0) \tag{7.1}$$

$$FDR = E\left[\frac{V}{R}|R > 0\right] \tag{7.2}$$

In order to compensate for multiple hypothesis tests that are performed while searching for the best rules, WSARE requires a randomization test. This randomization test builds a histogram of the null hypothesis distribution of the test statistic. Under the null hypothesis, the date and the other case features are assumed to be independent. The test statistic we calculate is the best scoring rule found in each of the randomized data sets. The compensated p-value returned is the probability of having a best score as extreme as the score found on the original “recent” and “baseline” data sets. The randomization test controls the FWER since the p-value is an estimate of the probability of seeing at least one type I error under the null hypothesis of independence of date and case features for the test statistic.

There are two main concerns with the randomization test portion of the algorithm. First of all, as was mentioned in Section 4.3, the randomization test is the bottleneck in the WSARE procedure. If there can be a faster method to account for the multiple hypothesis testing problem, then WSARE will run much faster. Secondly, the randomization test only reports the p-value of the best scoring rule found on a particular day. Although the best scoring rule may be the strangest event from WSARE's perspective, it may not be interesting to a public health official. Thus, by reporting only the p-value of the best scoring rule on a particular day, WSARE may hide some other anomalous patterns that are more interesting to public health practioners.

7.1 Faster methods to account for multiple hypothesis testing

Perhaps the best known technique to account for multiple hypothesis testing is the Bonferroni method. Suppose we have performed m tests and we would like to control the probability of a type I error, ie. rejecting the null hypothesis when it is true, at some level α. The Bonferroni method requires us to reject the null hypothesis for all hypothesis tests where the p-value is less than $\frac{\alpha}{m}$ (Bonferroni, 1936). In other words, any tests satisfying this condition would yield significant results.

The Bonferroni correction is a very conservative way to control the FWER. The Bonferroni correction takes into account neither the p-values of the individual tests nor the ranks of the test statistics. Let us define the adjusted p-value for hypothesis H_i as in (Ge et al., 2003):

$$\tilde{p}_i = inf\,\{\alpha :\ H_i \text{ is rejected at FWER} = \alpha\} \tag{7.3}$$

The Bonferroni correction then creates a conservative lower bound for the adjusted p-value of each test. While the Bonferroni correction does guarantee the FWER to be less than α, it loses power as the number of tests increases ie. the probability of maintaining the null hypothesis when it should in fact be rejected will approach one as m increases (Miller et al., 2001).

Implementing the Bonferroni correction is straightforward. It requires a constant time multiplication to all the p-values as opposed to the lengthier process of a randomization test. We replaced the Bonferroni correction for the randomization test and tested this variation on the Bayesian network simulator. The AMOC curve for the results, as shown in Figure 7.1 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. We expect that as the number of hypothesis tests increases, due to the number of attributes in the data increasing or by increasing the arity of each attribute, the power of the Bonferroni correction will decrease, resulting in poorer performance on an AMOC curve. In general, even though the Bonferroni correction is extremely fast, we would prefer a more powerful method to adjust for multiple hypothesis testing.

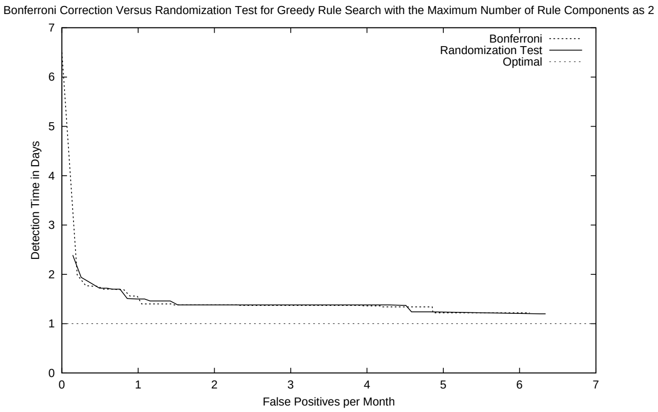


Figure 7.1: Asymptotic behavior of the Bonferoni correction versus the randomization test version

Another alternative is the Benjamini and Hochberg False Discovery Rate (BH-FDR) procedure (Benjamini & Hochberg, 1995), which was used in section 4.2.7 to account for multiple hypothesis testing due to comparison of p-values over multiple days when analyzing historical output from WSARE. Similarly, we can use BH-FDR to handle the multiple hypothesis testing problem due to comparison of rule scores which are produced by many hypothesis tests on a particular day. Unlike the randomization test which only returns the best scoring rule and its adjusted p-value, BH-FDR will produces a sorted list of the most significant rules for a particular day along with a guarantee of an expected false discovery rate of α_{FDR}. In addition, BH-FDR is a much faster procedure than the randomization test.

7.2 Reporting more than the best scoring rule for the day

As was previously mentioned, reporting only the best scoring rule for the current day can mask other potentially interesting anomalous patterns on that day. Consider the following scenario where the best scoring rule for today reveals an upswing in ED cases involving patients with ages between 80 and 90 from a sparse populated zipcode. While this rule is significant from WSARE's standpoint, suppose that after public health officials look at it, they determine that it is truly a statistical anomaly and not a public health threat. Since it is the best scoring rule, the next four best scoring rules individually have a lower score and are not reported by WSARE. However, suppose that the next four best rules do pick up on a food-borne outbreak and indicate that there is an unusually high proportion of ED cases from 4 neighboring zipcodes. Public health officials would be interested in this spatial cluster but they are not alerted to its presence.

The obvious solution would be to report the top n rules for a particular day. Nevertheless, we still need to associate a p-value for each of these rules. An extremely expensive solution would be to run a separate randomization test for each of the n rules. Let $\hat{p}_i$ be defined as the compensated p-value result from the randomization test in which the test statistic is the number of times the ith best rule found in the randomized data sets has a better score than the ith best rule for the original data set. Thus, in our computationally intensive solution, we could calculate $\hat{p}_1, \hat{p}_2, \ldots, \hat{p}_n$. A more feasible solution would be to associate some p-value with the entire set of n rules. This can be somewhat accomplished if WSARE reports the top n rules and then perform a randomization test in which the p-value estimate is $\hat{p}_n$. This p-value estimates the probability of seeing a set of n rules in which the nth best rule has a score as good as the score of the nth best rule on the original data set under the null hypothesis. Besides the difficulty in interpreting this p-value description, a major drawback to this approach is the fact that n needs to be known. We typically do not know a priori how many significantly anomalous rules there will be on a particular day. Suppose that in reality, there are 4 significantly anomalous events but since we set n to be 5, we "accidentally" include an insignificant fifth event which ends up making the p-value of the set insignificant as well.

Our solution to reporting multiple rules on a particular day is to use the BH-FDR procedure developed by Benjamini and Hochberg (Benjamini & Hochberg, 1995), which was described in detail in Section 4.2.7. Let an adjusted p-value for BH-FDR be defined as in Equation 7.3:

$$\tilde{p}_i = inf \left\{\alpha : H_i \text{ is rejected at FDR} = \alpha\right\} \tag{7.4}$$

For each null hypothesis test that is rejected by FDR, the adjusted p-value for that test can be calculated as in Equation 7.5, where $p_{r_1} \leq p_{r_2} \leq \ldots \leq p_{r_m}$ are the observed p-values sorted in increasing order and c_m is a value based on the dependence of the hypothesis tests.

$$\tilde{p}_{r_i} = \underset{k=i,\ldots,m}{min} \left\{ min(\frac{mc_m}{k} p_{r_k}, 1) \right\} \tag{7.5}$$

Note that this adjusted p-value is not a traditional p-value in the sense that it measures the probability of a type I error under the null hypothesis. Instead, it is the minimum FDR at which a p-value can be considered significant. This value is similar to q-values which are described in the next section. Public health officials that are accustomed to seeing traditional p-values may be unfamiliar with these adjusted p-values and may misinterpret them. As a result, when the significant rules for a day are reported using this BH-FDR modification, WSARE simply lists the rules without their p-values but in order of decreasing significance.

On a day with a disease outbreak, BH-FDR should in theory return a relatively large list of significant p-values along with their corresponding rules. Public health officials can use this list with some certainty knowing that the expected proportion of false discoveries in this list is α. Figure 7.2 illustrates the results of using this BH-FDR procedure on one of the data files from the CityBN simulation with $\alpha_{FDR} = 0.01$. On day 74420, the anthrax release occurred in this simulation, resulting in a large number of significant p-values being reported by WSARE after that day.

Another possibility that we explored was the use of q-values (Storey, 2003). Q-values can be considered as the pFDR parallel to a p-value. Following the notation of (Ge et al., 2003), let us define the rejection region Γ_α for significance level α. We can consider p-value(t) to be the minimum Type I error probability α over all rejection regions Γ_α containing the observed test statistic value t. Likewise, q-value(t) is defined as the minimum pFDR that can occur over all rejection regions Γ_α containing t. Informally, q-value(t) is the pFDR when we make t significant. Q-values can be

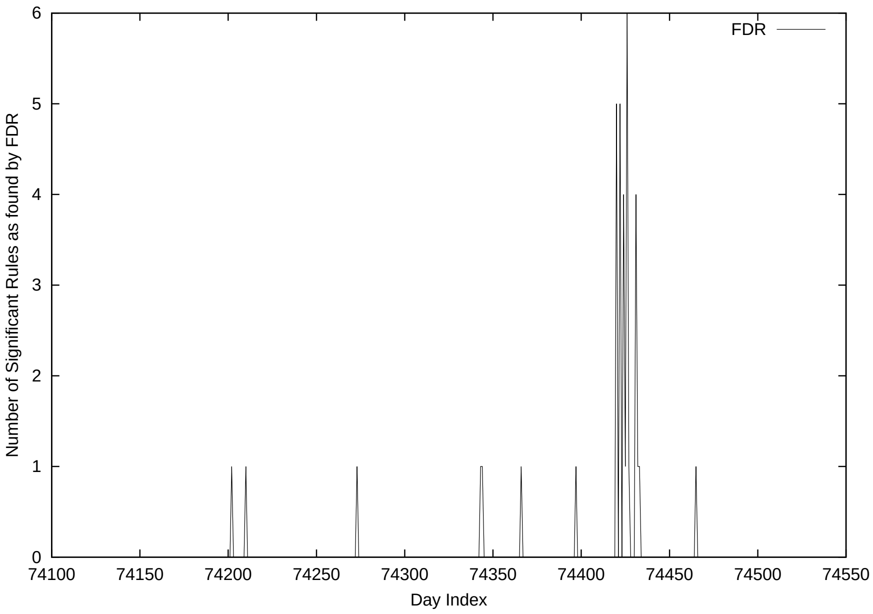


Figure 7.2: Number of significant p-values per day for CityBN data

estimated using either a fast cubic spline interpolation algorithm (Storey & Tibshirani, 2003) or a slower but more robust bootstrap algorithm (Storey et al., 2002).

The most attractive feature of the q-value procedure is that it can be shown to be a more powerful test than BH-FDR (Storey, 2002). If we define $\pi_0 = \frac{m_0}{m}$, where m_0 is the number of true null hypotheses and m is the total number of hypothesis test, as defined in 7.1, then BH-FDR controls for FDR at level $\frac{\alpha_{FDR}}{\pi_0}$ while the q-value procedure controls it at α. Since $\pi_0 \leq 1.0$ and $\alpha \leq \frac{\alpha_{FDR}}{\pi_0}$, the q-value procedure is more powerful than BH-FDR because it rejects the same number of null hypotheses as BH-FDR but at a lower significance level.

We were unable to use the q-value procedure because WSARE violates the assumptions of either independence among the null hypothesis p-values or weak dependence as defined in (Storey et al., 2002). Under the null hypothesis, the p-values and hence rules of WSARE are strongly dependent. For example, the p-value from a rule such as "Gender = Male AND Age = Senior" will be correlated with the rules such as "Gender = Male" and "Age = Senior". Furthermore, the benefit of using the q-value procedure over BH-FDR is relative to π_0. In a situation where $\pi_0 \approx 1$, there is no benefit over the BH-FDR procedure. For scenarios like DNA microarray analysis, we expect a large proportion of null hypotheses to be rejected and hence π_0 should be relatively low. On the other hand, for WSARE, there is little benefit since we expect π_0 to be relatively high most of the time.

Chapter 8

An Efficient Algorithm for Clustering with Clutter

The most common kinds of clustering algorithms tend to fall into two categories – mixture models and graph theoretic approaches such as hierarchical single linkage clustering. The best choice is task and data dependent. However, cases do exist where both of these styles perform poorly, such as in Figure 8.2, which has too much noise for the clustering algorithms to find the appropriate clusters. When single linkage clustering was asked to find five clusters in the example data set, it joined together nearly all the points, as shown in Figure 8.3. A mixture of Gaussians, optimized by EM, was also unsuccessful, as the results in Figure 8.4 indicate. Even though we permitted a uniform background component and started the iteration with hand picked Gaussians, EM was unable to correctly model the shape of the overdense area. Furthermore, seven Gaussians were required by EM, which is more than the three clusters that make up the region of overdensity. On the other hand, the Cuevas-Febrero-Fraiman algorithm, to be described shortly, successfully finds the correct number of clusters as shown in Figure 8.5.

Situations with a combination of noisy backgrounds and clusters that can not be modeled by parametric forms such as Gaussians are common enough in the physical sciences, such as astrophysics (Reichart et al., 1999), that a new approach is needed. As an example, consider the data set shown in Figure 8.6. This data set is a two dimensional projection taken from the Early Data Release of the Sloan Digital Sky Survey (The Sloan Digital Sky Survey, 2000). Each point is a galaxy containing many billions of stars. Astrophysicists are interested in clustering galaxies, but a noisy

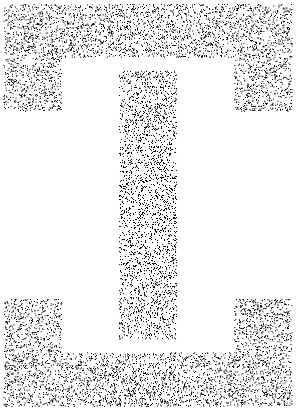


Figure 8.1: The anchor shape

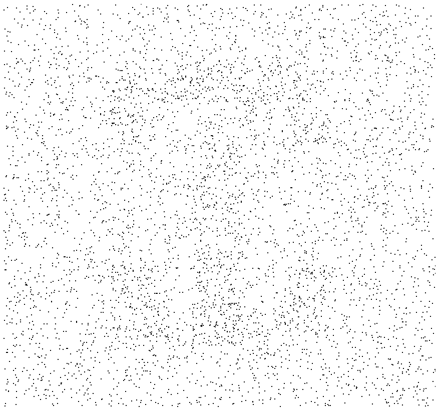


Figure 8.2: Example data set with a central anchor shape ma de of 3 components and 90% clutter

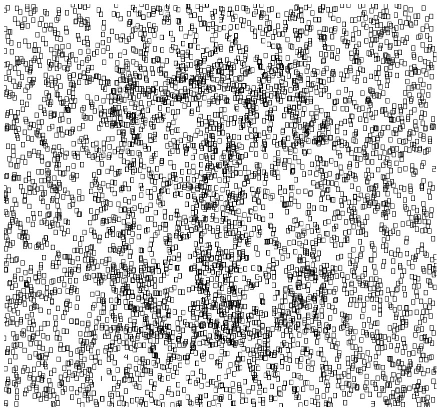


Figure 8.3: Clusters as found by Single Linkage Clustering. All points marked with a square are in one big cluster. Clusters 2 thro ugh 5 are tiny and not in useful places

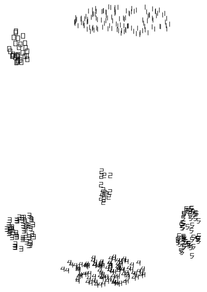


Figure 8.4: Clusters as found by a mixture of Gaussians with a uniform background component for clutter removal

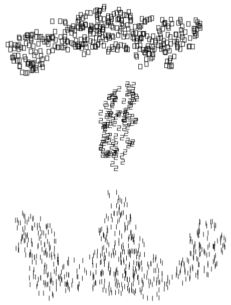


Figure 8.5: Clusters as found by CFF algorithm

background of field galaxies interferes with traditional clustering techniques. Figure 8.7 demonstrates the clusters found by using the Cuevas-Febrero-Fraiman algorithm. Other examples include detection of minefields (Dasgupta & Raftery, 1998), detection of seismic faults (Dasgupta & Raftery, 1998), robust estimation of covariance (Byers & Raftery, 1998), and identification of foci of activation in the brain (Pereira et al., 2001).

This type of clustering can also be extended to perform spatial surveillance of disease outbreaks. Suppose we are provided with the spatial locations of cases belonging to the disease of interest. We naturally expect some isolated cases over a geographic area, which we can consider as a noise. The cases that are of greater interest to epidemiologists are those that are within close proximity to each other. If we plot the case locations on a map, we can use density estimation to find the high density areas which would correspond to the areas with disease clusters. For chronic diseases, the low amount of noise on the map would not be a problem for a standard density estimation technique. In the case of syndromic surveillance data however, there is a potential for more noise in the spatial distribution since the data is non-specific. Moreover, we might consider more attributes than just the locations of cases. The addition of temporal, demographic, and symptom-related attributes would increase the dimensionality of the data. The Cuevas-Febrero-Fraiman algorithm, along with the optimizations described in this chapter, is ideal for finding clusters against a noisy background in two or more dimensions.

Simply using the Cuevas-Febrero-Fraiman algorithm on the locations of disease cases does not take into account the underlying at-risk population. As was mentioned in Section 3.2.2, the relative risk surface is a more accurate representation of the disease incidence in an area. In (Kelsall & Diggle, 1995; Bithell, 1990), the relative risk surface was formed by two kernel density estimates – one to model the expected distribution of cases and the other for the observed distribution. The areas of high risk can be found by finding the areas with high values of the ratio of the observed distribution to the expected distribution. Locating these high risk areas is similar to the process of finding areas of high density. Although we do not describe an efficient algorithm for finding the relative risk surface in this chapter, we present the optimizations to the Cuevas-Febrero-Fraiman algorithm as a first step and possible extensions are left for future work.

The algorithm proposed by Cuevas, Febrero and Fraiman (Cuevas et al., 2000),

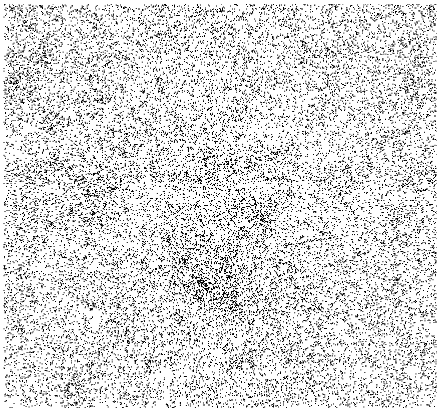


Figure 8.6: A data set of galaxies from the Sloan Digital Sky Survey

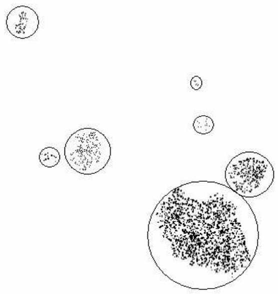


Figure 8.7: The seven clusters found by CFF and manually circled

while originally developed to estimate the number of clusters, can also be applied to the problem of finding clusters against a noisy background as we will show. We will hereafter abbreviate the Cuevas, Febrero and Fraiman algorithm to be the CFF algorithm.

8.1 The CFF Algorithm

One common definition, originated by Hartigan (Hartigan, 1975), of the number of clusters in a data set is the number of connected components of the level set $\{f > c\}$, where f is a density function on $\Re^d$ and c is a positive constant. In (Cuevas et al., 2000), CFF describes a two step algorithm that takes three parameters, a bandwidth h, a density threshold c, and a link-length ϵ:

- **Step One:** Find the set of high density datapoints $\{X_i \,\epsilon\, \text{Data set} \mid \hat{f}_h(X_i) > c\}$, where $\hat{f}_h$ is a kernel density estimator of f with bandwidth h.

- **Step Two:** Construct the graph in which the vertices are the high density datapoints and in which the nodes X_i and X_j are linked together if and only if $|X_i - X_j| \; \leq \epsilon$, where $|X_i - X_j|$ is the Euclidean distance. Find the connected components of this graph.

The second step forms a set of spanning trees with the X_is in which the edges are of length ϵ or less. In fact, the algorithm proposed by CFF forms a Minimum

Spanning Tree since in that algorithm the next point to be considered is the closest point to the members of the current connected component.

While the algorithm is conceptually simple, it suffers from computational issues, especially when there are many datapoints and more than one dimension. Nonparametric density estimation is expensive. Euclidean Minimum Spanning Trees (EMSTs) can be determined in $O(nlogn)$ time in two dimensions using Delauney triangulations (Preparata & Shamos, 1985), where n is the number of datapoints. However, this technique does not scale well as the dimensionality increases. We propose a fast version of the CFF algorithm by addressing the computational issues in both of these steps.

8.1.1 Scaling Step One: Nonparametric Density Estimation

This paper will pay relatively little attention to Step One for two reasons. First, although it is critical to the scalability of the method, it can be addressed using technology to be reported in a separate paper (Gray & Moore, 2002). That paper describes how two ideas can allow near linear time detection of high density points. The first idea uses a dual tree algorithm similar to that described in (Gray & Moore, 2001) while the second idea cuts off the search early without computing exact densities in a manner similar to the single ball-tree approach of (Moore, 2000)

The second reason we will pay no more attention to Step One is that frequently the CFF algorithm is executed with a sequence of hundreds of different values of c and ϵ but with the same bandwidth h and thus the same density estimate $\hat{f}$. For this scenario the speed of Step One is much less critical than Step Two, since provided we can obtain numerical density estimates for all datapoints (again, using a new all-kernel algorithm to be described in (Gray & Moore, 2002)), we must run Step 2 hundreds of times for each execution of Step One.

8.1.2 Scaling Step Two: Connected Components

The CFF algorithm for finding connected components does not address the issue of obtaining the distances between points when forming the Minimum Spanning Tree. The initial improvement over the CFF method is straightforward. We simply exploit recent results in computational geometry for efficient EMSTs. This is based on the GeoMST2 algorithm described in (Narasimhan et al., 2000). One of the key concepts behind GeoMST2 is the use of Well-Separated Pair Decompositions, which were developed by Callahan (Callahan, 1995), (Callahan & Kosaraju, 1995). We will now outline the GeoMST2 algorithm, after which we will introduce two additional speedups that are specific to the CFF algorithm and gain at least one extra order of magnitude speed, especially in higher dimensions.

8.2 Well-Separated Pair Decomposition

Figure 8.14 illustrates a Well-Separated Pair. Let A and B be point sets in $\Re^d$ with bounding hyper-rectangles R_A and R_B respectively. The notation Diam(R_A) indicates the diameter of the hyper-rectangle R_A. MargDistance(A,B) is defined to be the minimum distance between the hyper-rectangles R_A and R_B. The point sets A and B are considered to be well-separated (Callahan & Kosaraju, 1995) if MargDistance(A,B) $\geq$ max{Diam(R_A), Diam(R_B)} (Narasimhan et al., 2000). The *interaction product* between two point sets A and B is defined as:

$$A \otimes B = \{\{p, p'\} |\ p\ \epsilon\ A, p'\ \epsilon\ B,\ \text{and}\ p \neq\ p'\}$$

A set of pairs $(A_1, B_1), \dots, (A_k, B_k)$ is said to be a well-separated realization of $A\ \otimes\ B$ if the following properties (Callahan, 1995) hold:

1. $A_i \subseteq A$ and $B_i \subseteq B$ for all $i = 1, \dots, k$.

2. $A_i \cap B_i = \emptyset$ for all $i = 1, \dots, k$.

3. $(A_i\ \otimes\ B_i)\ \cap\ (A_j \otimes B_j) = \emptyset$ for all i, j such that $i \neq j$

4. $A \otimes B = \bigcup_{i=1}^{k} A_i \otimes B_i$

5. A_i and B_i are well-separated for all $i = 1, \dots, k$.

Notice that $A_1 \otimes B_1, \dots, A_k \otimes B_k$ is a partitioning of the large set consisting of all pairs of points $(a \in A, b \in B)$. Every pair of points exists in exactly one (A_i, B_i). And in addition, every (A_i, B_i) has the property of being well-separated.

If P is a point set in $\Re^d$ then a Well-Separated Pair Decomposition, hereafter abbreviated to WSPD, of P is a set of pairs $(A_i, B_i), \dots, (A_k, B_k)$ which is a well-separated realization of $P \otimes P$. Somewhat astonishingly, a WSPD of a point set P of size n can be constructed with only $O(n)$ elements and in only $O(nlogn)$ time (Callahan & Kosaraju, 1995). This decomposition is performed using a fair split tree (Callahan & Kosaraju, 1995), which is essentially a k-d tree.

Figures 8.9 to 8.12 illustrate a WSPD of the data set with five datapoints shown in Figure 8.8. This decomposition was formed using a fair split tree. The WSPD contains four pairs. The members in each pair of the WSPD are drawn using a large black circle and a light gray circle. The bounding hyper-rectangles of the pairs in the WSPD are also shown.

8.3 GeoMST2

The GeoMST2 algorithm (Narasimhan et al., 2000) is an improvement on the GeoMST algorithm (Callahan & Kosaraju, 1993) and (Eppstein, 1999). Both algorithms use WSPDs in order to address the issue of forming EMSTs in higher dimensions. GeoMST2 differs from its predecessor by reducing the number of Bichromatic Closest Pair (BCP) distance calculations. The BCP distance of two point sets A_i and B_i is defined to be the shortest distance between a and b where $a \epsilon A_i$ and $b \epsilon B_i$. GeoMST2 has an expected running time of $O(nlogn)$.

The algorithm is best explained by looking at the pseudocode in Figure 8.13. Line 1 forms the WSPD of a point set S. The edges of the final EMST are selected from among the pairs of the WSPD. In lines 3-5, GeoMST2 inserts each pair of the WSPD into a priority queue, using the MargDistance as the priority. In order to form a EMST correctly, lines 6-14 must remove the pairs of the priority queue in the order of ascending BCP distance. However, in line 4, the priority used for the queue is the MargDistance, which is a lower bound for the BCP distance and it is also much cheaper to compute. We will shortly explain how GeoMST2 does indeed produce a EMST.

Let the element currently removed be the pair (A_i, B_i). Line 8 determines if A_i and B_i are already connected. If they are connected, then the pair is ignored and we have possibly saved ourselves a BCP calculation. Otherwise, line 9 determines if the BCP distance has been calculated. In the case that the BCP distance has not been computed, Lines 12-14 calculate the BCP pair and distance. The Well-Separated Pair is reinserted into the priority queue with the actual BCP distance as the priority. Line 14 flips the ComputedBCP flag to be true for that pair. If the BCP distance has been

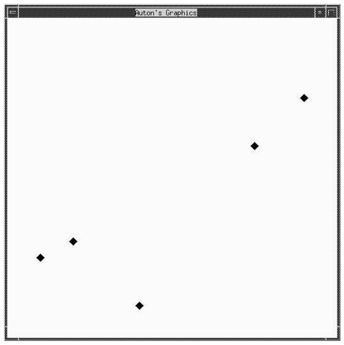


Figure 8.8: The original data set

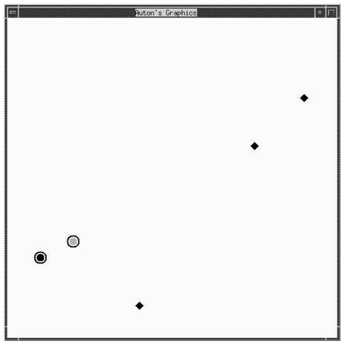


Figure 8.9: WSPD Pair 1

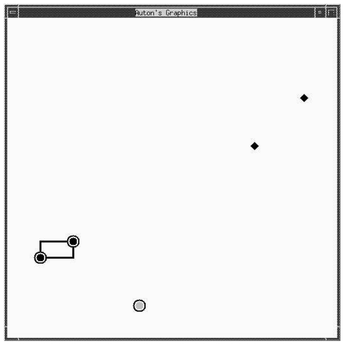


Figure 8.10: WSPD Pair 2

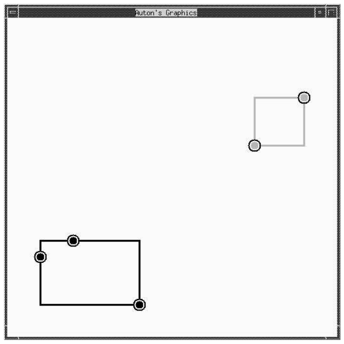


Figure 8.11: WSPD Pair 3

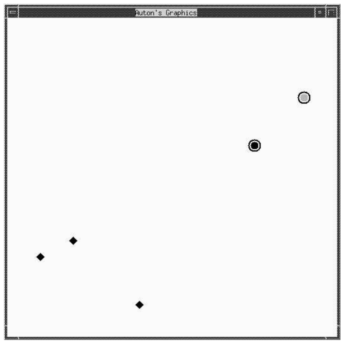


Figure 8.12: WSPD Pair 4

```
Line #
0......GeoMST2(S)
1......  –(A1,B1),...,(Am,Bm)˝ = WSPD(S)
2......  E = 0, PQ = 0
3......  for i = 1 to m do
4......    PQ.Insert(i,MargDistance(Ai,Bi))
5......    ComputedBCP[i] = FALSE
6......  while(not PQ.Empty()) do
7......    i = PQ.ExtractMin()
8......    if( Ai and Bi are not connected )
9......      if( ComputedBCP[i] )
10.....        E = E Union –(ai,bi)˝
11.....      else
12.....        –ai,bi,D˝ = BCP(Ai,Bi,Infinity)
13.....        PQ.Insert(i,D)
14.....        ComputedBCP[i] = true
15.....   T = (S,E)
16.....   return T
```

Figure 8.13: The GeoMST2 algorithm

computed, as shown in line 10, an edge is formed between the element $a \epsilon A_i$ and the element $b \epsilon B_i$ that yields the BCP distance.

There are two crucial points for the correctness of GeoMST2. First of all, Line 13 reinserts a pair into the priority queue when the BCP distance has been calculated; this BCP distance will be at least as large as the MargDistance. Secondly, an edge is only added to the MST if the actual BCP distance has been computed. These two points guarantee that an edge is added to the EMST only when all shorter edges have already been handled.

8.3.1 CFF-specific optimizations

A naive implementation would use GeoMST2 to calculate the minimum spanning tree until the distance obtained from the priority queue was greater than ϵ. This is wasteful for us because all we need are ϵ-clusters, i.e. clusters formed by joining all points that are within ϵ distance or less from each other. A point may be joined to a cluster through any link of length less than ϵ—not necessarily the link that would appear in the EMST. This observation can be exploited by modifying the step that finds the WSPD while leaving the rest of the GeoMST2 algorithm intact. If we can reduce the number of pairs that are in the WSPD, then GeoMST2 will be much faster due to less elements in the priority queue, as will be shown in Figures 8.18 and 8.19.

Optimization 1: Ignoring links that are $> \epsilon$

The main purpose of the WSPD is to reduce the number of edges under consideration when forming the MST. Every pair (A_i, B_i) in the set of pairs of the WSPD becomes an edge in the EMST unless it joins two already connected components. If the minimum distance separating the bounding hyper-rectangles of the sets A_i and B_i is greater than ϵ, as illustrated in Figure 8.15, then an edge of length ϵ or less cannot possibly exist between a point in A_i to a point in B_i. While forming the WSPD, we need not include any (A_i, B_i) IPs with separation distance greater than ϵ.

Optimization 2: Joining all elements that are within ϵ distance of each other

If the maximum distance separating the bounding hyper-rectangles of A_i and B_i is less than ϵ, then the points in A_i and B_i must belong to the same ϵ-cluster. We can simply join all the points in A_i and B_i if they are not already connected. Since we need not form a EMST, we can connect the points in any order. Furthermore, we need not add such pairs that have this property to the WSPD.

Summary

These two optimizations speed up two crucial areas of the GeoMST algorithm. First of all, they reduce the amount of time required to produce the WSPD. Secondly, the number of well-separated pairs is reduced, thereby speeding up the later half of the GeoMST2 algorithm by reducing the size of the priority queue.

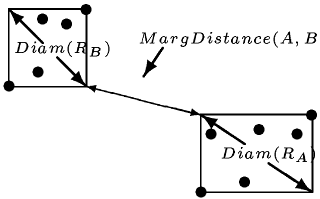


Figure 8.14: A Well-Separated Pair

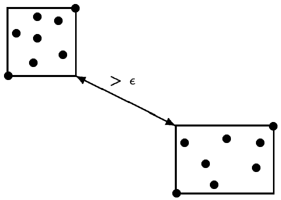


Figure 8.15: Optimization 1

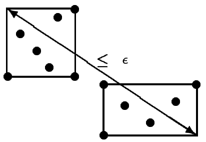


Figure 8.16: Optimization 2

8.4 Results

Experiments were conducted on a subset of data taken from the Sloan Digital Sky Survey (The Sloan Digital Sky Survey, 2000), which is a digital map of the sky. The data set used had 7 columns, consisting of four colors, two sky coordinates, and a redshift value. We performed most of our experiments on the four color values.

Figure 8.17 compares the time taken to perform the clustering step of the CFF algorithm by using Kruskal's MST algorithm, GeoMST2, and our new implementation called ϵ-clustering. Modifications were made to both Kruskal's algorithm and GeoMST2 to terminate the algorithm the moment an edge that was greater than ϵ was encountered. The experiments were run on 4 dimensional data sets with 200 to 1000 records, in increments of 200. Since Kruskal's algorithm is non-competitive as the number of datapoints are above 10000, we compare the speeds of GeoMST2 and ϵ-clustering on data sets of size 20000 to 100000, in increments of 20000 records. These experiments were run on 3 and 4 dimensional data. Figures 8.18 and 8.19 show the results. ϵ-clustering outperforms GeoMST2 in all the cases.

Figure 8.20 shows the effects of increasing the number of datapoints on the time taken by ϵ-clustering. We ran experiments on data sets of size 200000 to 1000000, in increments of 200000. The experiments were performed on data in 2 to 5 dimensions. For certain settings of the parameters, the time increases linearly, while for others, time appears to increase on the order of $O(nlogn)$.

These experiments indicate that the bandwidth h, threshold c, and ϵ have an impact on the running time. h and c affect the number of high density datapoints. The ϵ parameter affects the time spent in the WSPD step. If ϵ is very large, the WSPD step ends early as all the points in the fair split tree node can be connected. Similarly, a small value of ϵ results in more pairs that are separated by a distance greater than ϵ. Figures 8.21 and 8.22 illustrate the effect of varying ϵ on time on a 3-dimensional and a 4-dimensional data set with 100000 records each. The time seems to peak at about a value of 0.5 for ϵ for both plots.

8.5 Related Work

Banfield and Raftery (Banfield & Raftery, 1993) use a mixture-model approach to perform clustering on d-dimensional data with clutter. The data is assumed to be

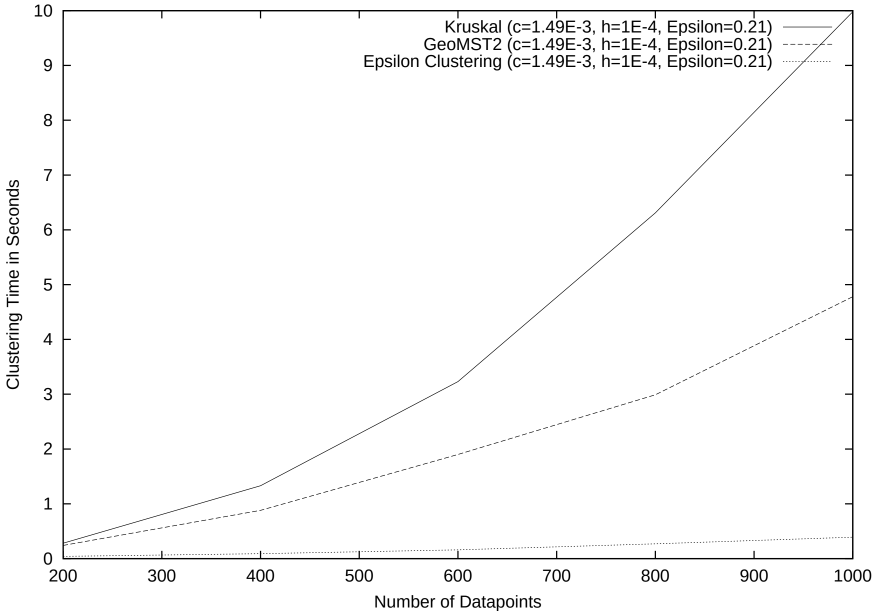


Figure 8.17: GeoMST2 vs. ϵ-Clustering vs Kruskal in 4D

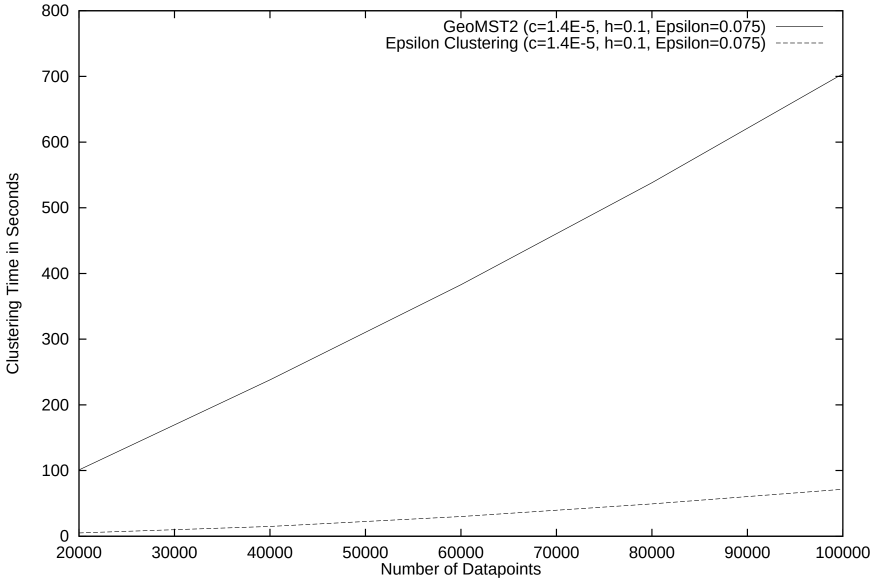


Figure 8.18: GeoMST2 vs. ϵ-Clustering in 3D

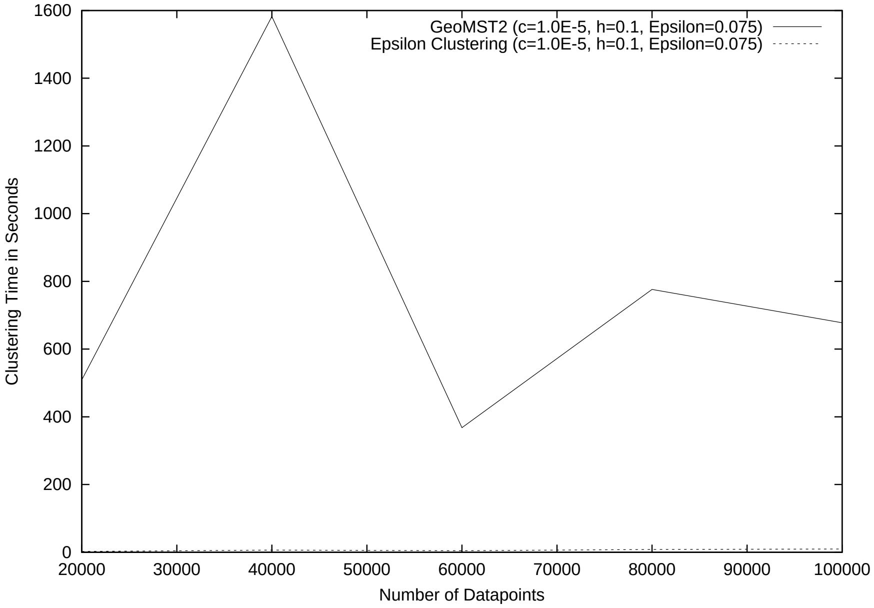


Figure 8.19: GeoMST2 vs. ϵ-Clustering in 4D

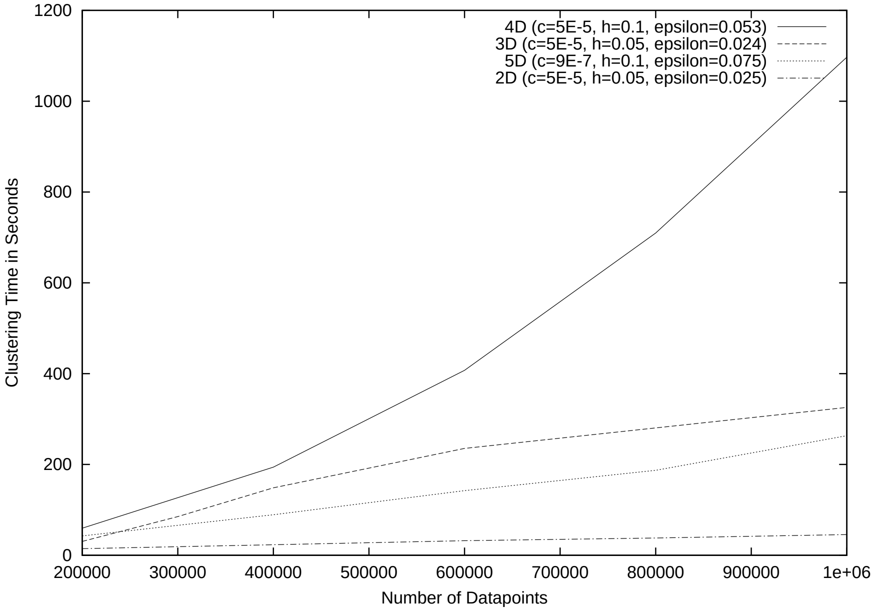


Figure 8.20: Change in time as Number of Datapoints increases

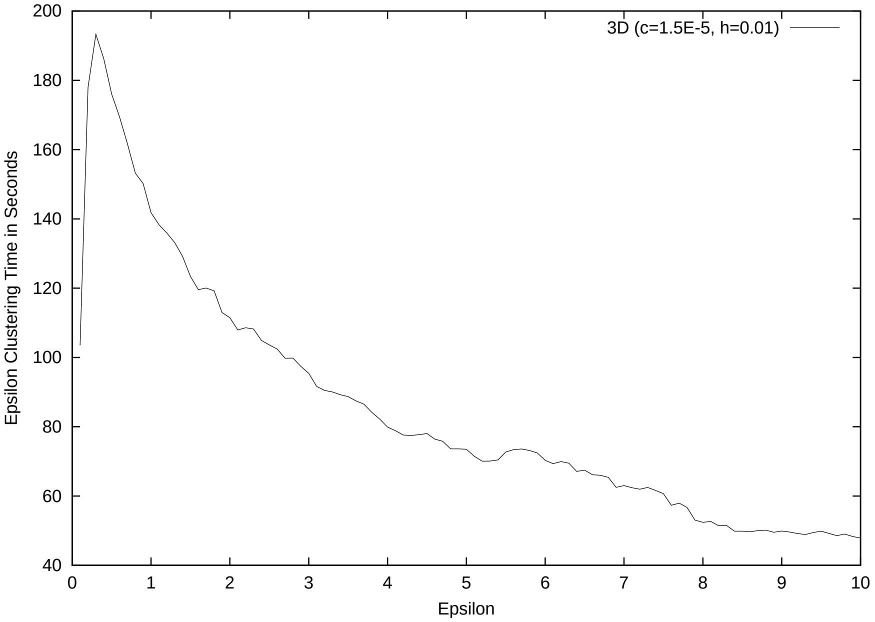


Figure 8.21: Change in Time as a Function of ϵ for 3D data

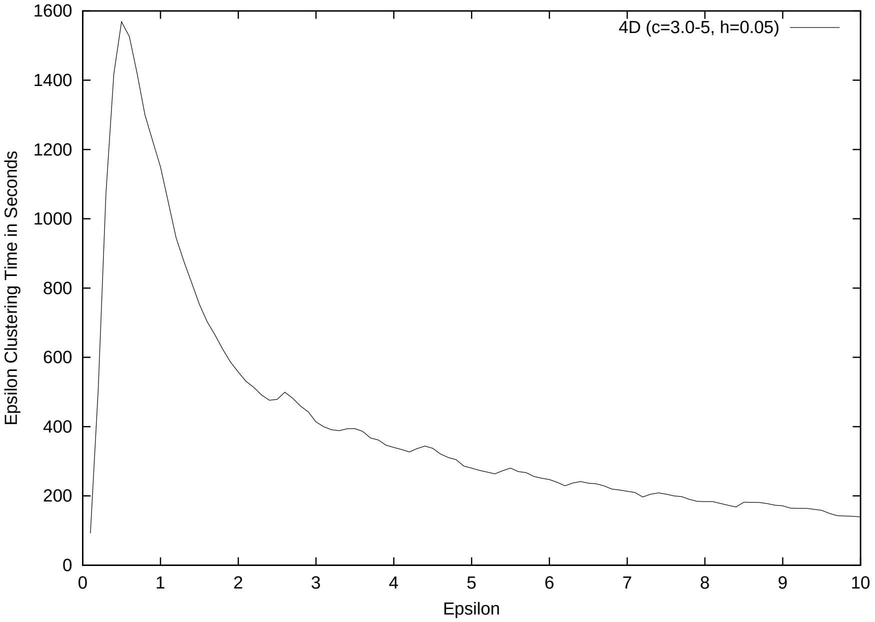


Figure 8.22: Change in Time as a Function of ϵ for 4D data

generated from a mixture of Gaussians together with a homogeneous spatial Poisson process which produces the clutter. The clusters are assumed to be highly linear or piecewise linear, where linear means that they are concentrated along their first principal component. Clustering is performed by maximizing the parameters and the partitioning of the data over the classification likelihood. Their approach, however, requires the user to provide information about the shape of the clusters and the algorithm's approximation of the number of clusters in the data performs poorly when the clutter makes up a majority of the data. Dasgupta and Raftery (Dasgupta & Raftery, 1998) extend on the work by Banfield and Raftery. The technique proposed by (Dasgupta & Raftery, 1998) iterates over a range of possible number of clusters for the data. In each iteration, an initial clustering is performed using the method proposed by Banfield and Raftery. Then, an EM step follows in which the maximized mixture likelihood is obtained upon convergence and this value is used to compute the Bayesian Information Criterion score for the model. The EM step is also modified to estimate the shape of the clusters from the data. The best model, along with its clustering, is selected.

Byers and Raftery (Byers & Raftery, 1998) propose a further algorithm that is not restricted by the shape or the number of the clusters. Once again, the clutter is assumed to be a homogeneous Poisson point process, but the features are also considered as a Poisson process on top of the clutter. This approach computes, for every data point, the kth nearest neighbor distance, where k is a parameter supplied by the user. This distance is modeled as a mixture of two gamma distributions, one for the clutter and the other for the features. The next step of the algorithm applies EM to estimate the parameters of the mixture. From the results of EM, a data point can be classified as clutter if its probability of belonging to the clutter component is higher than that of belonging to the feature component and vice versa.

8.6 Conclusion

ϵ-clustering is a computationally efficient algorithm for determining the connected components in Step 2 of the CFF algorithm. Our results show that it outperforms GeoMST2, one of the fastest EMST algorithms, by nearly an order of magnitude in higher dimensions. Combining ϵ-clustering with the nonparametric density estimation algorithms in (Gray & Moore, 2002) will yield an effective algorithm for identifying clusters against a noisy background with massive amounts of data.

8.7 Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 0121671

Chapter 9

Future Work and Conclusions

9.1 Future Work

9.1.1 Accounting for Uncertainty in the Bayesian network

When a Bayesian network structure is learned from data, there may be a small number of events for a particular parent and node value combination that results in a large variance for the estimate of the corresponding parameter. The Optimal Reinsertion algorithm currently fills the conditional probability table in each node with just the mean of the posterior distribution for that parameter while ignoring its variance. WSARE 3.0 then uses the network and its parameters to produce a p-value for each rule. One of the main concerns of WSARE 3.0 is that the variances of the network parameters are not taken into consideration when calculating the p-value. We are less confident about a p-value obtained from infrequent counts than for one calculated from a large sample, yet the current algorithm treats all p-values identically.

In general, Bayesian network structure learning algorithms following the framework of (Heckerman et al., 1995) implicitly store counts from which the parameter variance can be calculated but this information is typically ignored. We can exploit this dormant information to determine the posterior distribution of a query response, which we will define as the probability of a rule given the value of its environmental attributes eg. $Pr(X_0, \ldots, X_n | \text{Season} = \text{Winter}, \text{Day of Week} = \text{Friday})$. One proposed solution requires determining how sensitive the query response is to the Bayesian network parameters. A quantitative measure can be obtained by calculating the partial derivatives of the query response with respect to the Bayesian network parameters (van Allen, 2000). The parameter variances can then be propagated through the system of partial derivatives to obtain an approximation of the query response variance. At this point, the mean and the variance of the query response are known and the posterior distribution of the query response can be modelled as a normal (van Allen et al., 2001; van Allen, 2000) or as a Beta distribution (Singh, 2004). Finally, a p-value that incorporates the variances of the parameters can be obtained from a possibly modified version of this posterior distribution.

9.1.2 Incorporating the Spatial Scan Statistic

The Spatial Scan Statistic (Kulldorff, 1997) can be considered as the spatial analog to WSARE. Both algorithms can be summarized in two steps. The first step requires searching and scoring of anomalous events. After the most unusual event is found, the second step involves a Monte Carlo simulation to associate a p-value with the most unusual event. For the search phase, WSARE searches over one and two component rules while the Spatial Scan Statistic searches over scanning windows, which are typically circles. In order to score its regions, the Spatial Scan Statistic assumes a background Poisson or Bernoulli process. With this assumption, a likelihood ratio can be used as a scoring function. WSARE, on the other hand, makes no assumptions on the distribution of the null hypothesis. Instead, it uses Fisher's Exact test on a two-by-two contingency table. These two scoring functions are making different comparisons. The Spatial Scan Statistic compares events from different geographic regions while WSARE compares events from a recent time period against a baseline time period. In the second step, both algorithms rely on Monte Carlo simulations to handle the multiple hypothesis testing problem in the first step. However, the algorithms differ in the specifics of the simulation, with the randomization test for WSARE being somewhat slower since the distribution of the null hypothesis is done through randomization while the Spatial Scan Statistic simply generates cases from a known model.

With these similarities, a hybrid algorithm between the two approaches seems like an obvious next step. As a simple first step, WSARE can be extended to consider the spatial components of the data somewhat differently from the other attributes. During the rule search, WSARE can consider rules such as "Gender = Male" AND "All cases within a radius r of point (x_i, y_i)". A more advanced modification of

WSARE would allow it to handle any number of real valued attributes, perhaps by incorporating an approach similar to applying a one dimensional scan statistic for each real valued attribute. Clearly, computational issues are at the forefront for these extensions. Other concerns involve designing a scoring function and a Monte Carlo simulation that can be compatible for both algorithms.

9.1.3 Multiple time windows

WSARE compares events from the past 24 hours to events from a baseline period. A scenario that WSARE might not detect is when events in the past 24 hours do not seem unusual but events in the most recent 4 hours contain striking irregularities. By looking at data from the past 24 hours, WSARE smooths out the anomalous patterns that it might have noticed if it only considered data from the past 4 hours. As a result, one extension we would like to add is to allow WSARE to search for anomalous events over multiple time windows eg. events from the past 4 hours, 6 hours, 12 hours, etc. Naturally, many counting statistics can be reused in order to alleviate the computational burden of this modification.

9.1.4 WSARE in higher dimensions

WSARE presently operates on low dimensional, categorical and dense data. On the other end of the spectrum is high dimensional and sparse data such as text mining data and drug design data. We would like to extend WSARE so that it is able to find anomalous patterns in high dimensional data. A frequently suggested approach for high dimensional data is to find a lower dimensional subspace of the problem through Principal Component Analysis and then search for low probability data points according to some model (Shyu et al., 2003). However, as was mentioned previously, this anomaly detection method finds isolated outliers and is inappropriate for detecting shifts in the proportions of groups between one time period and another.

9.1.5 Non-biosurveillance data

Since WSARE makes few assumptions about the data, it is an extremely general algorithm that can be applied to data other than biosurveillance streams. Examples of potential applications for WSARE include fraud detection (Fawcett & Provost, 1997), monitoring maintenance of trains and airplanes, topic detection, monitoring of robot logs, and supernova detection. We would certainly like to evaluate the performance of WSARE on such tasks in the near future.

9.2 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 a priori, 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 a 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 Zipcode = 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. In addition, we assume that by monitoring rules with a given limit on the number of components, which is typically two for WSARE 2.0, we are able to characterize this cluster in attribute space. 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 features.

We evaluated the effectiveness of the rule search in the gridworld simulator, where a disease with a target demographic group of males in their 50s was released into the simulator. With this target group and the locality of interaction in the simulator, we were able to produce data that was ideal for detection by WSARE 2.0. As a result, when compared against three univariate algorithms that relied on aggregate daily counts, WSARE 2.0 outperformed them in terms of sensitivity, specificity, and timeliness.

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 features 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 was 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.

The CityBN simulator was built in order to generate surveillance data which contain temporal fluctuations due to day of week effects and seasonal variations of background illnesses such as flu, food poisoning and hayfever. In addition, many of the attributes from the two Bayesian networks used to generate data for the simulator were not explicitly included as attributes in the surveillance data. Some of these hidden variables were responsible for temporal fluctuations in the surveillance data.

WSARE 3.0 used the environmental attributes of season, day of week, flu level and weather in order to detect the anthrax release in the simulated data. Despite the fact that these environmental attributes 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 compared WSARE 3.0 against three univariate algorithms – a control chart, a moving average algorithm, and an ANOVA regression algorithm. The ANOVA regression is an effective detector in the face of temporal fluctuations, as was shown in the DARPA Challenge (Buckeridge et al., 2004). These three algorithms operated on two univariate versions of the data – one version consisted of total daily counts while the other was the daily counts of cases involving respiratory problems. In both cases, WSARE 3.0 outperformed all the univariate algorithms in terms of sensitivity, specificity and timeliness. WSARE 3.0 also produced a better AMOC curve than WSARE 2.0 because WSARE 2.0 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.

Although WSARE performed well on our simulated data, when it was applied to the DARPA challenge data, its limitations could be seen. First of all, WSARE cannot detect a spike in total daily counts because it relies on changes in ratios as the signal of an outbreak. Secondly, the DARPA challenge data was extremely nonstationary. The environmental attributes of season and day of week could not account for the variation due to new sources of data contributing data and previously existing sources not consistently reporting data. Another problem for WSARE was the fact that the "seasonal" trends in the DARPA challenge did not follow seasonal or monthly time boundaries and we approximated it using seasons. In general, knowing the exact behavior of temporal trends is difficult and sometimes these trends are not even consistent from year to year. Finally, in data sets with more than one outbreak, WSARE incorporated data during epidemic periods into its baseline since it had no knowledge that an outbreak had occurred.

Multivariate surveillance data with known outbreak periods is extremely difficult to obtain. As a result, we resorted to evaluating WSARE on simulated data. We concede that although the simulated data sets posed challenging problems to detection algorithms, the simulators do not reflect real life. Furthermore, the simulators that were built generated data which satisfied many of the assumptions inherent to the rule search and the Bayesian network modelling approaches of the algorithm. We suggest two further evaluation techniques to obtain a better understanding of the effectiveness of WSARE. First of all, a very common simulation strategy is to use existing multivariate surveillance data as a background and inject outbreaks into this data. Since actual data is used, the generated data sets will contain real temporal trends in the background. However, the process of adding outbreaks must be carefully done so that the outbreaks cannot be easily detected using an obvious univariate signal. In general, creating a challenging multivariate detection problem is extremely difficult and subject to many iterations of fine tuning. Our second suggestion is to run WSARE on actual surveillance data, such as the ED data used in Sections 6.4 and 6.5 and then validate the results with public health officials. Of course, this process is time consuming but some ground truth can be established.

In short, WSARE has been demonstrated to outperform traditional univariate methods over simulated data in terms of timeliness, sensitivity and specificity. Its performance over real world data requires further evaluation. Furthermore, although WSARE is a powerful detection algorithm, a naive implementation of it results in a computationally expensive algorithm. We have included a variety of optimizations which substantially speed up the algorithm, including techniques such as greedy rule search, racing, Optimal Reinsertion, and differential counting. 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. Recently, WSARE 3.0 has been deployed to monitor ED cases in Pennsylvania and other states.

Appendix A

A sample Bayesian network learned from simulator data

Figure A.1 illustrates an example Bayesian network learned from the CityBN simulator data. The contingency tables for each node have also been shown in this section. Each row in the contingency table corresponds to a combination of parent values and each entry in the row corresponds to the probability of taking on that node value given the parent values.

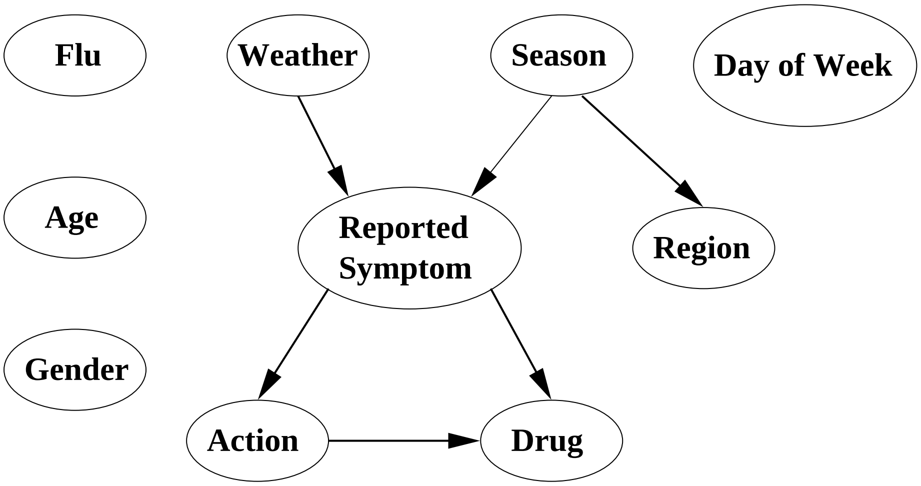


Figure A.1: A learned Bayesian network from the CityBN data

Region

Season	C	E	N	NE	NW	S	SE	SW	W
fall	0.075	0.116	0.157	0.200	0.034	0.074	0.234	0.065	0.043
spring	0.103	0.068	0.098	0.281	0.024	0.119	0.131	0.117	0.059
summer	0.084	0.076	0.099	0.204	0.023	0.086	0.274	0.113	0.041
winter	0.069	0.105	0.158	0.202	0.040	0.076	0.233	0.077	0.040

Age

	child	senior	working
	0.212	0.326	0.462

Gender

	female	male
	0.500	0.500

Flu

	decline	high	low	none
	0.302	0.130	0.057	0.511

Day of Week

	sat	sun	weekday
	0.146	0.149	0.704

Action

Reported Symptom	absent	evisit	purchase
nausea	0.559	0.441	0.001
none	0.215	0.035	0.750
rash	0.324	0.674	0.001
respiratory	0.300	0.703	0.000

Reported Symptom

Weather	Season Value	nausea	none	rash	respiratory
cold	fall	0.150	0.539	0.017	0.294
cold	spring	0.126	0.530	0.082	0.262
cold	winter	0.159	0.537	0.011	0.293
hot	fall	0.139	0.454	0.125	0.282
hot	spring	0.141	0.481	0.127	0.251
hot	summer	0.145	0.504	0.137	0.214

Drug

Reported Symptom	Action	aspirin	none	nyquil	vomit-b-gone
absent	nausea	0.142	0.580	0.172	0.106
absent	none	0.150	0.585	0.154	0.111
absent	rash	0.004	0.989	0.004	0.004
absent	respiratory	0.165	0.598	0.138	0.099
evisit	nausea	0.001	0.996	0.001	0.001
evisit	none	0.004	0.987	0.004	0.004
evisit	rash	0.002	0.995	0.002	0.002
evisit	respiratory	0.000	0.999	0.000	0.000
purchase	none	0.321	0.000	0.412	0.267

Bibliography

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

Anderson, N. H., & Titterington, D. M. (1997). Some methods for investigating spatial clustering, with epidemiological applications. *Journal of the Royal Statistical Society*, *160*, 87–105.

Assuncao, R., & Reis, E. (1999). A new proposal to adjust moran's i for population density. *Statistics in Medicine*, *18*, 2147–2162.

Aussem, A., & Murtagh, F. (1997). Combining neural network forecasts on wavelet-transformed time series. *Connection Science*, *9*, 113–121.

Bailey, T. C., & Gatrell, A. C. (1995). *Interactive spatial data analysis.* New York, New York: John Wiley and Sons, Inc.

Banfield, J., & Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, *49*, 803–821.

Banks, J. (1989). *Principles of quality control.* John Wiley and Sons, Inc.

Baron, M. I. (2002). Bayes and asymptotically pointwise optimal stopping rules for the detection of influenza epidemics. In C. Gastonis, R. E. Kass, A. Carriquiry, A. Gelman, D. Higdon, D. Pauler and I. Verdinell (Eds.), *Case studies in bayesian statistics*, vol. 6, 153–163. New York: Springer-Verlag.

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.

Besag, J., & Newell, J. (1991). The detection of clusters in rare diseases. *Journal of the Royal Statistical Society. Series A (Statistics in Society*, *154*, 143–155.

Bickel, P., & Yahav, J. (1967). Asymptotically pointwise optimal procedures in sequential analysis. *Proc. 5th Berkeley Symp. Math. Statist. Prob.* (pp. 401–413). Berkeley, California: Univ. California Press.

Bickel, P., & Yahav, J. A. (1968). Asymptotically optimal bayes and minimax procedures in sequential estimation. *Ann. Math. Stat*, 442–456.

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

Bithell, J. F. (1990). An application of density estimation to geographical epidemiology. *Statistics in Medicine*, *9*, 691–701.

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.

Box, G. E. P., & Jenkins, G. M. (1976). *Time series analysis: Forecasting and control.* San Francisco: Holden-Day.

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.

Buckeridge, D. L., Burkom, H., Campbell, M., & Moore, A. W. (2004). Outbreak detection algorithms: a synthesis of research from the darpa bioalirt project. In preparation.

Buntine, W. (1991). Theory Refinement on Bayesian Networks. *Proceedings of the Seventh Conference on UAI* (pp. 52–60).

Burkom, H. S. (2003). Biosurveillance applying scan statistics with multiple, disparate data sources. *Journal of Urban Health: Bulletin of the New York Academy of Medicine*, *80*, i57–i65.

Byers, S., & Raftery, A. E. (1998). Nearest-neighbour clutter removal for estimating features in spatial point processes. *Journal of the American Statistical Association*, *93*, 577–584.

Callahan, P. (1995). *Dealing with higher dimensions: the well-separated pair decomposition and its applications*. Doctoral dissertation, Johns Hopkins University, Baltimore, Maryland.

Callahan, P., & Kosaraju, S. (1995). A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields. *Journal of the ACM*, *62*, 67–90.

Callahan, P., & Kosaraju, S. R. (1993). Faster algorithms for some geometric graph problems in higher dimensions. *ACM-SIAM Symposium on Discrete Algorithms* (pp. 291–300).

Chakrabarti, S., Sarawagi, S., & Dom, B. (1998). Mining surprising patterns using temporal description length. *Proceedings of the 24th International Conference of Very Large Databases*.

Chatfield, C. (1989). *The analysis of time series: An introduction*. London: Chapman and Hall. 4th edition.

Chickering, D. M. (1996a). Learning bayesian networks is np-complete. In D. Fisher and H. Lenz (Eds.), *Learning from data: Artificial inteligence and statistics v*, 121–130. Springer-Verlag.

Chickering, D. M. (1996b). Learning equivalence classes of bayesian network structures. *Proceedings of the Twelfth Conference on UAI* (pp. 150–157). Portland, Oregon: Morgan Kaufmann.

Chickering, D. M., Meek, C., & Heckerman, D. (2003). Large-sample learning of bayesian networks is np-hard. *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence* (pp. 124–133). Morgan Kaufmann.

Choi, K., & Thacker, S. B. (1981). An evaluation of influenza mortality surveillance, 1962-1979 i. time series forecasts of expected pneumonia and influenza deaths. *American Journal of Epidemiology*, *113*, 215–226.

Cooper, G. F., & Herskovits, E. (1992). A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*, 309–347.

Cuevas, A., Febrero, M., & Fraiman, R. (2000). Estimating the number of clusters. *The Canadian Journal of Statistics*, *28*, 367–382.

Cuzick, J., & Edwards, R. (1990). Spatial clustering for inhomogeneous populations. *Journal of the Royal Statistical Society, Series B*, *52*, 73–104.

Cuzick, J., & Edwards, R. (1996). Cuzick-edwards one-sample and inverse two-sampling statistics. In *Methods for investigating localized clustering of disease*, 200–202. International Agency for Research on Cancer, Lyon: IARC Scientific Publicastions 135.

Das, D., Weiss, D., Mostashari, F., Treadwell, T., McQuiston, J., Hutwagner, L., Karpati, A., Bornschlegel, K., Seeman, M., Turcios, R., Terebuh, P., Curtis, R., Heffernan, R., & Balter, S. (2003). Enhanced drop-in syndromic surveillance in new york city following september 11, 2001. *Journal of Urban Health, Supplement 1*, *80*, i76–i88.

Dasgupta, A., & Raftery, A. E. (1998). Detecting features in spatial point processes with clutter via model-based clustering. *Journal of the American Statistical Association*, *93*, 294–302.

Diggle, P., Chetwynd, A., Haggvist, R., & Morris, S. (1995). Second-order analysis of space-time clustering. *Statistical Methods in Medical Research*, *4*, 124–136.

Diggle, P. J. (2000). Overview of statistical methods for disease mapping and its relationship to cluster detection. In P. Elliott, J. C. Wakefield, N. G. Best and D. J. Briggs (Eds.), *Spatial epidemiology methods and applications*, 87–103. New York: Oxford University Press.

Diggle, P. J., & Chetwynd, A. G. (1991). Second-order analysis of spatial clustering for inhomogeneous populations. *Biometrics*, *47*, 1155–1163.

Dong, G., & Li, J. (1999). Efficient mining of emerging patterns: Discovering trends and differences. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Elidan, G., Ninio, M., Friedman, N., & Schuurmans, D. (2002). Data perturbation for escaping local maxima in learning. *Proceedings of AAAI-02* (pp. 132–139).

Eppstein, D. (1999). Spanning trees and spanners. In J.-R. Sack and J. Urrutia (Eds.), *Handbook of computational geometry*, 425–461. North-Holland,Amsterdam: Elsevier Science Publishers B.V.

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

Farrington, C. P., Andrews, N. J., Beale, A. D., & Catchpole, M. A. (1996). A statistical algorithm for the early detection of outbreaks of infectious disease. *Journal of the Royal Statistics Society*, *159*, 547–563.

Farrington, C. P., & Beale, A. D. (1998). The detection of outbreaks of infectious disease. *Proceedings of the International Workshop on Geomedical Systems (GEOM ED '97)* (pp. 97–117). Leipzig: B. G. Teubner.

Farrington, P., & Andrews, N. (2004). Outbreak detection: Application to infectious disease surveillance. In R. Brookmeyer and D. F. Stroup (Eds.), *Monitoring the health of populations*, 203–231. New York, New York: Oxford University Press.

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery* (pp. 1–28). Boston: Kluwer Academic Publishers.

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. *Proceedings on the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 53–62). San Diego, CA.

Feagin, O. T. (1971). Maine's sentinel physician system. *J Maine Med Assoc*, *62*, 187.

Fitzner, K. A., McGhee, S. M., Headley, A. J., & Shortridge, K. F. (1999). Influenza surveillance in hong kong: results of a trial physician sentinel programme. *Hong Kong Med Journal*, *5*, 87–94.

Friedman, N., & Goldszmidt, M. (1997). Sequential update of Bayesian network structure. *Proceedings of the Thirteenth Conference on UAI* (pp. 165–174).

Friedman, N., Nachman, I., & Peér, D. (1999). Learning Bayesian network structure from massive datasets: The "sparse candidate" algorithm. *Proceedings of the Fifteenth Conference on UAI* (pp. 206–215).

Ganti, V., Gehrke, J. E., Ramakrishnan, R., & Loh, W. (1999). A framework for measuring changes in data characteristics. *Proceedings of the Eighteenth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems.*

Ge, Y., Dudoit, S., & Speed, T. P. (2003). Resampling-based multiple testing for microarray data analysis. *Sociedad de Estadistica e Investigacion Operativa Test*, *12*, 1–77.

Geary, R. C. (1954). The contiguity ratio and statistical mapping. *Incorporated Statistician*, *5*, 115–145.

Glaz, J., & Balakrishnan, N. (1999). Introduction to scan statistics. In J. Glaz and N. Balakrishnan (Eds.), *Scan statistics and applications*, 4–18. Boston, MA: Birkhauser.

Goldenberg, A. (2001). Framework for using grocery data for early detection of bioterrorism attacks. Master's thesis, Carnegie Mellon University.

Goldenberg, A., Shmueli, G., & Caruana, R. (2003). Using grocery sales data for the detection of bio-terrorist attacks. Submitted to Statistics in Medicine.

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

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

Gray, A., & Moore, A. W. (2001). N-body problems in statistical learning. *Advances in Neural Information Processing Systems 13 (December 2000).* MIT Press.

Gray, A., & Moore, A. W. (2002). Efficient kernel density algorithms. In Preparation.

Hamerly, G., & Elkan, C. (2001). Bayesian approaches to failure prediction for disk drives. *Proceedings of the eighteenth international conference on machine learning* (pp. 202–209). Morgan Kaufmann, San Francisco, CA.

Hamilton, J. D. (1994). *Time series analysis.* Princeton, New Jersey: Princeton University Press.

Hartigan, J. (1975). *Clustering algorithms.* New York: John Wiley.

Hastie, T. J., & Tibshirani, R. J. (1990). *Generalized additive models.* London: Chapman and Hall.

Heckerman, D., Geiger, D., & Chickering, D. M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, *20*, 197–243.

Hulten, G., & Domingos, P. (2002). Mining complex models from arbitrarily large databases in constant time. *Proc. 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.*

Hutwagner, L., Thompson, W., Seeman, G. M., & Treadwell, T. (2003). The bioterrorism preparedness and response early aberration reporting system (EARS). *Journal of Urban Health*, *80*, i89–i96.

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

Janes, G. R., Hutwagner, L., JR., W. C., up, D. F. S., & Williamson, G. D. (2000). Descriptive epidemiology: Analyzing and interpreting surveillance data. In S. M. Teutsch and R. E. Churchill (Eds.), *Principles and practice of public health surveillance*, 112–167. New York: Oxford.

Kelsall, J. E., & Diggle, P. J. (1995). Non-parametric estimation of spatial variation in relative risk. *Statistics in Medicine*, *14*, 2335–2342.

Kelsall, J. E., & Diggle, P. J. (1998). Spatial variation in risk of disease: a nonparametric binary regression approach. *Applied Statistics*, *47*, 559–573.

Knox, E. G. (1964). The detection of space-time interactions. *Applied Statistics*, *13*, 25–29.

Koch, M. W., & McKenna, S. A. (2001). *Near-real time surveillance against bioterror attack using space-time clustering* (Technical Report SAND2001-0695C). Sandia National Laboratories.

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

Kulldorff, M. (1999a). Spatial scan statistics: models, calculations, and applications. In J. Glaz and N. Balakrishnan (Eds.), *Scan statistics and applications*, 303–322. Boston, MA: Birkhauser.

Kulldorff, M. (1999b). Statistical evaluation of disease cluster alarms. In *Disease mapping and risk assessment for public health*, 143–149. New York, New York: John Wiley and Sons, Ltd.

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

Kulldorff, M., Huang, L., & Pickle, L. (2003). An elliptic spatial scan statistic and its application to breast cancer mortality data in northestern united states.

Kulldorff, M., & Information Management Systems Inc. (2003). Satscan v 3.1: Software for the spatial and space-time scan statistics. http://www.satscan.org/.

Kulldorff, M., Rand, K., & Williams, G. (1997). Satscan v 1.0: Software for the spatial and space-time scan statistics.

Lane, T., & Brodley, C. E. (1999). Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security*, *2*, 295–331.

Lawson, A. B. (2001). *Statistical methods in spatial epidemiology.* New York, New York: John Wiley and Sons, Ltd.

Lawson, A. B., & Kulldorff, M. (1999). A review of cluster detection methods. In A. Lawson, A. Biggeri, D. Bohning, E. Lesaffre, J.-F. Viel and R. Bertollini (Eds.), *Disease mapping and risk assessment for public health*, 99–110. New York: John Wiley and Sons, Ltd.

Lawson, A. B., & Williams, F. L. R. (1993). Applications of extraction mapping in environmental epidemiology. *Statistics in Medicine*, *12*, 1249–1258.

Lazarus, R., Kleinman, K., Dashevsky, I., Adams, C., Kludt, P., Alfred DeMaria, J., & Platt, R. (2002). Use of automated ambulatory-care encounter records for detection of acute illness clusters, including potential bioterrorism events. *Emerging Infectious Diseases*, *8*.

Lee, J., & Wong, D. W. S. (2001). *Statistical analysis with arcview gis*. New York: John Wiley and Sons, Inc.

Lucas, J. M., & Saccucci, M. S. (1990). Exponentially weighted moving average control schemes: Properties and enhancements. *Technometrics*, *32*, 1–29.

Mantel, N. (1967). The detection of disease clustering and a generalised regression approach. *Cancer Research*, *27*, 209–220.

Maron, O., & Moore, A. W. (1997). The racing algorithm: Model selection for lazy learners. *Artificial Intelligence Review*, *11*, 193–225.

Marshall, R. J. (1991). A review of methods for the statistical analysis of spatial patterns of disease. *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, *154*, 421–44.

Maxion, R. A., & Tan, K. M. C. (2001). *Anomaly detection in embedded systems* (Technical Report CMU-CS-01-157). Carnegie Mellon University.

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.

Montgomery, D. C. (2001). *Introduction to statistical quality control*. John Wiley and Sons, Inc. 4th edition.

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*.

Moore, A. W. (2000). The Anchors Hierarchy: Using the triangle inequality to survive high dimensional data. *Twelfth Conference on Uncertainty in Artificial Intelligence*. AAAI Press.

Moore, A. W., & Schneider, J. (2002). Real-valued All-Dimensions search: Low-overhead rapid searching over subsets of attributes. *Conference on UAI* (pp. 360–369).

Moran, P. A. P. (1950). Notes on continuous stochastic phenomena. *Biometrika*, *37*, 17–23.

Mostashari, F., & Hartman, J. (2003). Syndromic surveillance: a local perspective. *Journal of Urban Health*, *80*, i1–i7.

Narasimhan, G., Zhu, J., & Zachariasen, M. (2000). Experiments with computing geometric minimum spanning trees. *Proceedings of ALENEX'00* (pp. 183–196). Springer-Verlag.

Naus, J. I. (1965). The distribution of the size of the maximum cluster of points on a line. *Journal of the American Statistical Association*, *60*, 532–538.

Neill, D. B., & Moore, A. W. (2003). *A fast multi-resolution method for detection of significant spatial overdensities* (Technical Report CMU-CS-03-154). Carnegie Mellon University.

Nobre, F. F., & Stroup, D. F. (1994). A monitoring system to detect changes in public health surveillance data. *International Journal of Epidemiology*, *23*, 408–418.

Oden, N. (1995). Adjusting moran's i for population density. *Statistics in Medicine*, *14*, 17–26.

Ogden, R. T. (1997). *Essential wavelets for statistical applications and data analysis.* Boston: Birkhauser.

Openshaw, S., Charlton, M., Craft, A. W., & Birch, J. M. (1988). Investigation of leukaemia clusters by use of a geographical analysis mac hine. *The Lancet*, *1*, 272–273.

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

Percival, D. B., & Walden, A. T. (2000). *Wavelet methods for time series analysis.* Cambridge, United Kingdom: Cambridge University Press.

Pereira, F., Just, M., & Mitchell, T. (2001). Distinguishing natural language processes on the basis of fmri-measured brain activation. *Principles of Data Mining and Knowledge Discovery* (pp. 374–385). Springer-Verlag.

Polikar, R. (2001). http://engineering.rowan.edu/ polikar/WAVELETS/WTtutorial.html.

Preparata, F., & Shamos, M. (1985). *Computational geometry: an introduction.* New York: Springer-Verlag.

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

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

Redondo, S. M., Gil, M. M., Hernandez, C. B., & Simon, M. V. (2002). Requests for hiv tests and their performance in primary health care. study of the sentinel physician network of castilla y leon in spain, 1990-1996. *Gac Sanit*, *16*, 114–120.

Reichart, D., Nichol, R., Castander, F., Burke, D., Romer, A., Holden, B., Collins, C., & Ulmer, M. (1999). A deficiency of high-redshift, high-luminosity x-ray clusters: Evidence for a high value of omega matter? *The Astrophysical Journal*, *518*, 521–532.

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

Reis, B. Y., Pagano, M., & Mandl, K. D. (2003). Using temporal context to improve biosurveillance. *Proceedings of the National Academy of Sciences of the United States of America*, *100*, 1961–1965. http://www.pnas.org/cgi/doi/10.1073/pnas.0335026100.

Research and Practice Fundamentals (2003). Available at http://rpf.health.pitt.edu/rpf/modules/edMod.cfm?main=edIndex.

Ripley, B. D. (1976). The second-order analysis of stationary point processes. *Journal of Applied Probability*, *13*, 255–266.

Roberts, S. W. (1959). Control-charts-tests based on geometric moving averages. *Technometrics*, *1*, 239–250.

Rogerson, P. A. (1997). Surveillance systems for monitoring the development of spatial patterns. *Statistics in Medicine*, *16*, 2081–2093.

Rogerson, P. A. (2001). Monitoring point patterns for the development of space-time clusters. *Journal of the Royal Statistics Society, Series A*, *164*, 87–96.

Schwartz, G. (1979). Estimating the dimensions of a model. *Annals of Statistics*, *6*, 461–464.

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

Shyu, M.-L., Chen, S.-C., Sarinnapaorn, K., & Chang, L. (2003). A novel anomaly detection scheme based on principal component classifier. *Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03)* (pp. 172–179). Melbourne, Florida, USA.

Simonsen, L., Clarke, M. J., Williamson, D., Stroup, D. F., Arden, N. H., & Schonberger, L. B. (1997). The impact of influenza epidemics on mortality: Introducing a severity index. *American Journal of Public Health*, *87*, 1944–1950.

Singh, A. P. (2004). 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.

Sonesson, C., & Bock, D. (2003). A review and discussion of prospective statistical surveillance in public health. *Journal of the Royal Statistical Society*, *166*, 5–22.

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

Stern, L., & Lightfoot, D. (1999). Automated outbreak detection: a quantitative retrospective analysis. *Epidemiol. Infect.*, *122*, 103–110.

Storey, J. D. (2002). A direct approach to false discovery rates. *Journal of the Royal Statistical Society: Series B*, *64*, 479–498.

Storey, J. D. (2003). The positive false discovery rate: A bayesian interpretation and the q-value. *The Annals of Statistics*, *0*, 1–23.

Storey, J. D., Taylor, J. E., & Siegmund, D. (2002). Strong control, conservative point estimation, and simultaneous conservative consistency of false discovery rates: A unified approach. *Journal of the Royal Statistical Society, Series B*. In press.

Storey, J. D., & Tibshirani, R. (2003). Statistical significance for genome-wide studies. *Proceedings of the National Academy of Sciences*, *100*, 9440–9445.

Stroup, D. F., Williamson, G. D., Herndon, J. L., & Karon, J. M. (1989). Detection of aberrations in the occurrence of notifiable diseases survei llance data. *Statistics in Medicine*, *8*, 323–329.

Tango, T. (1984). The detection of disease clustering in time. *Biometrics*, 15–26.

Tango, T. (1995). A class of tests for detecting 'general' and 'focused' clustering of rare diseases. *Statistics in Medicine*, *14*, 2323–2334.

Tango, T. (1999). Comparison of general tests for spatial clustering. In A. Lawson, A. Biggeri, D. Bohning, E. Lesaffre, J.-F. Viel and R. Bertollini (Eds.), *Disease mapping and risk assessment for public health*, 111–116. New York: John Wiley and Sons, Ltd.

The Health Insurance Portability and Accountability Act (1996). Available at http://aspe.hhs.gov/admnsimp/pl104191.htm.

The Sloan Digital Sky Survey (2000). http://www.sdss.org.

Tillett, H. E., & Spencer, I.-L. (1982). Influenza surveillance in england and wales using routine statistics. *Journal of Hygiene*, *88*, 83–94.

Tsui, F.-C. (1996). *Time series prediction using a multi-resolution dynamic predictor.* Doctoral dissertation, University of Pittsburgh, Pittsburgh.

Tsui, F.-C., Wagner, M. M., Dato, V., & Chang, C.-C. H. (2001). Value of icd-9-coded chief complaints for detection of epidemics. *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* (pp. 711–715). Hanley and Belfus, Inc.

Turnbull, B. W., Iwano, E. J., Burnett, W. S., Howe, H. L., & Clark, L. C. (1990). Monitoring for clusters of disease: application to leukemia incidence in upstate new york. *American Journal of Epidemiology*, *132*, S136–S143.

van Allen, T. (2000). 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.

van Allen, T., Greiner, R., & Hooper, P. (2001). Bayesian error-bars for belief net inference. *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence.*

van Casteren, V., & Leurquin, P. (1992). Eurosentinel: development of an international sentinel network of general practitioners. *Methods Inf Med*, *31*, 147–52.

Velleman, P. F., & Hoaglin, D. C. (1981). *Applications, basics, and computing of exploratory data analysis.* Boston: Duxbury Press.

Wagner, M., Carley, K., Cooper, G., Fridsma, D., Moore, A., Schneider, J., & Tsui, R. (2001a). Scalable biosurveillance systems proposal.

Wagner, M. M., Tsui, F. C., Espino, J. U., Dato, V. M., Sittig, D. F., Caruana, R. A., McGinnis, L. F., Deerfield, D. W., Druzdzel, M. J., & Fridsma, D. B. (2001b). The emerging science of very early detection of disease outbreaks. *Journal of Public Health Management Practice*, *7*, 51–59.

Wagner, M. M., Tsui, F. C., Espino, J. U., Dato, V. M., Sittig, D. F., Caruana, R. A., McGinnis, L. F., Deerfield, D. W., Druzdzel, M. J., & Fridsma, D. B. (2001c). The emerging science of very early detection of disease outbreaks. *Journal of Public Health Management Practice*, *7*, 51–59.

Wakefield, J. C., Kelsall, J. E., & Morris, S. E. (2000). Clustering, cluster detection, and spatial variation in risk. In P. Elliott, J. C. Wakefield, N. G. Best and D. J. Briggs (Eds.), *Spatial epidemiology methods and applications*, 128–152. New York: Oxford University Press.

Watier, L., Richardson, S., & Hubert, B. (1991). A time series construction of an alert threshold with application to s. bovismorbificans in france. *Statistics in Medicine*, *10*, 1493–1509.

Weatherall, J. A. C., & Haskey, J. C. (1976). Surveillance of malformations. *British Medical Bulletin*, *32*, 39–44.

Whittemore, A. S., Friend, N., Byron W. Brown, J., & Holly, E. A. (1987). A test to detect clusters of disease. *Biometrika*, *74*, 631–635.

Williamson, G. D., & Hudson, G. W. (1999). A monitoring system for detecting aberrations in public health surveillance reports. *Statistics in Medicine*, *18*, 3283–3298.

Xiang, Y., Wong, S., & Cercone, N. (1997). A microscopic study of minimum entropy search in learning decomposable markov networks. *Machine Learning*, *26*, 65–92.

Zhang, J., Tsui, F.-C., Wagner, M. M., & Hogan, W. R. (2003). Detection of outbreaks from time series data using wavelet transform. *Proc AMIA Fall Symp* (pp. 748–752). Omni Press CS.