RANK-BASED TEMPO-SPATIAL CLUSTERING: A FRAMEWORK FOR RAPID OUTBREAK DETECTION USING SINGLE OR MULTIPLE DATA STREAMS

by

Jialan Que

Bachelor of Science, Nankai University, China, 2001

Master of Science, Nankai University, China, 2004

Master of Science, University of Pittsburgh, 2008

Submitted to the Graduate Faculty of

Intelligent Systems Program in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

University of Pittsburgh

2012

UNIVERSITY OF PITTSBURGH

INTELLIGENT SYSTEMS PROGRAM

This dissertation was presented

by

Jialan Que

It was defended on

April 13, 2012

and approved by

Gregory F. Cooper, Associate Professor, Biomedical Informatics and Intelligent Systems

Roger S. Day, Associate Professor, Biomedical Informatics and Biostatistics

Milos Hauskrecht, Associate Professor, Computer Science

Dissertation Advisor: Fu-Chiang Tsui, Research Assistant Professor, Biomedical Informatics and Intelligent Systems

RANK-BASED TEMPO-SPATIAL CLUSTERING: A FRAMEWORK FOR RAPID OUTBREAK DETECTION USING SINGLE OR MULTIPLE DATA STREAMS

Jialan Que, PhD

University of Pittsburgh, 2012

In recent decades, algorithms for disease outbreak detection have become one of the main interests of public health practitioners as a way to identify and localize an outbreak as early as possible in order to inform further public health response to prevent a pandemic from developing. Today's increased threat of biological warfare and terrorism provide an even stronger impetus to develop methods for outbreak detection based on symptoms as well as definitive laboratory diagnoses.

In this dissertation work, I explore the problems inherent to rapid disease outbreak detection using both spatial and temporal information. I develop a framework of non-parameterized algorithms which search for patterns of disease outbreak in spatial sub-regions of the monitored region within a certain period. Compared to the current existing spatial or tempo-spatial algorithm, the algorithms in this framework provide a methodology for fast searching of either a univariate data set or multivariate data set. It first measures how likely a study area has an outbreak occur given the baseline data and currently observed data. Then it applies a greedy searching mechanism to look for clusters with high posterior probabilities given the risk measurement for each unit area as a heuristic. The performance of the proposed algorithms is then evaluated.

From the perspective of predictive modeling, I adopted a Gamma-Poisson (GP) model to compute the probability of having an outbreak in each cluster when analyzing univariate data. I built a multinomial generalized Dirichlet (MGD) model to identify outbreak clusters from multivariate data that include the OTC data streams collected by the national retail data monitor (NRDM) [1] and the ED data streams collected by the RODS system [2].

Key contributions of this dissertation include 1) the introduction of a rank-based tempo-spatial clustering algorithm, RSC, which utilizes greedy searching and a Bayesian GP model for disease outbreak detection with comparable detection timeliness, cluster positive prediction value (PPV) and improved running time; 2) the proposing of a multivariate extension of RSC (MRSC) which applies an MGD model. The evaluation demonstrates the advantage of the MGD model in effectively suppressing the false alarms caused by baseline shifts.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.0 Introduction

Until the 20th century, infectious disease outbreaks, whether naturally occurring or caused by bioterrorist attacks, routinely devastated the world's urban population. Between 1348 and 1351 the Black Plague killed 25% to 50% of Europe's population. In 1518, smallpox wiped out a large portion of the native population of Hispaniola (now Haiti and the Dominican Republic) and spread from there to Mesoamerica (present-day Mexico), contributing to the demise of the Aztecs. And more recently, the SARS outbreak in 2003 took away 774 people's lives. This last outbreak started from an apartment building in Hong Kong and then quickly spread to most of the southern cities in China and several countries nearby in Asia. Recent H1N1 swine flu outbreaks starting from Mexico are also an example.

To our knowledge, these disease outbreaks normally start from a relatively small number of geo-locations and then expand to larger, often contiguous or non-contiguous geographical areas when morbidity and mortality become significant and economic loss becomes large. Thus in the recent decades, it has become one of the main interests of public health practitioners to identify and localize an outbreak as early as possible in order to warrant further public health response before a pandemic develops. Today's increased threat of biological warfare and terrorism provide an even stronger impetus to develop methods for outbreak detection based on symptoms as well as definitive laboratory diagnoses.

1.1 Research domain

From the perspective of data dimensions analyzed for outbreak detection, currently there are three principal approaches: temporal analysis, spatial analysis and tempo-spatial analysis.

Temporal analysis using time series algorithms is the most popular approach due to its relatively simple handling of time series data as opposed to other types of information such as demographic data or geographic data. Time series algorithms such as moving average (*e.g.,* Exponentially Weighted Moving Average [3]), control charts (e.g., cumulative sum (CuSUM) [4]), adaptive linear regressions (e.g., Recursive Least Squares [5]), the Bayesian change-point detector [6], and the Wavelet Anomaly Detector (WAD) [7], are commonly used in biosurveillance systems.

The other two principal approaches, spatial and tempo-spatial algorithms, allow for better detection and localization of the outbreaks caused by infectious but non-contagious disease agents (*e.g.,* aerosol release of *B. Anthraces*, water borne diseases caused by *pathogenic microorganisms*, etc.), which typically spread in an aggregated group of geographic areas. Moreover, spatial approaches are also desired to analyze (either retrospectively or prospectively) geographical patterns of non-infectious syndromes such as infant death [8], prostate cancer survival data [9] and other data types. Essentially, the common use of a spatial algorithm is not limited to outbreak detection, but to test whether there are significant aberrancies correlated with geographical distributions. Current state-of-the-art spatial algorithms include Kulldorff's spatial scan statistic (KSS) [8] and the Bayesian spatial scan statistic (BSS) [10]. Takahashi, et. al. recently developed a flexible spatial scan statistic (FSS), which is an improvement over KSS in that it relaxes the constraint on cluster shape [11]. This method, however, results in higher complexity, which is impractical for processing large data sets. Other algorithms, such as the risk-adjusted nearest neighbor hierarchical clustering algorithm (RNNH) [12] and support vector machines (SVMs) [13], utilize traditional clustering approaches proven to be computationally efficient. However, it is a challenge for these approaches to automatically determine the required control parameters (*e.g.* parameters that can be set to influence the number and the shape of the clusters) [13].

In addition, researchers have started using the multivariate analysis on multiple data types to rapid detect and monitor unusual disease outbreak. Multiple data types are routinely collected by public health surveillance systems, such as chief complaints from emergency departments (ED), school or work absenteeism data, sales of over-the-counter (OTC) health care products and daily measurements of water quality. Some of these data are believed to have similar responses to some disease outbreaks. For example, if a flu-like disease outbreak occurs in a region, we often expect its effects to be seen in both OTC medications sales and emergency department chief complaint records. To be more specific, consider the characteristics of OTC data and ED data. The signal of an outbreak is often expected to appear first in OTC medication sales then in ED data since individuals with the initial symptoms of the disease will typically attempt to treat themselves before seeking medical care [14]. While the early signal in OTC data is appealing as an elevation, this signal will probably be weak. However, if the signal of ED chief complaint data appears following the signal of OTC data, the indication of an outbreak is stronger.

1.2 Overview of the proposed methodology

Despite the success of existing disease outbreak detection algorithms, most of them face some common limitations. First, they are computationally intensive due to extensive searching and/or randomization testing. This is important as in time-sensitive applications, an algorithm taking too long to complete can render its results outdated or delayed for decision makers. For instance, directly applying these algorithms to large data sets will probably result in computational infeasibility. Second, certain artificial cluster shapes (e.g., circle, rectangle) used by some algorithms may not conform to true outbreak clusters which may provide inaccurate information for decision makers. Furthermore, in addition to the univariate algorithms, there is still limited research on the multivariate analysis for rapid outbreak detection given multiple data streams.

In this dissertation work, I explore the problems of rapid disease outbreak detection using both spatial and temporal information. I develop a non-parameterized framework which searches for patterns of disease outbreak in spatial sub-regions of the monitored region within a certain period. Compared to the current existing spatial or tempo-spatial algorithm, the algorithms in this framework provide a methodology for fast searching. It applies a measurement to decide which study area is more likely to have an outbreak occurring given the baseline data and currently observed data. Then it will apply a greedy searching mechanism to look for clusters with high posterior probabilities given the risk measurement for each unit area as heuristic. I will also explore the performance of the proposed framework.

In this framework, I adopt a Gamma-Poisson model to compute the probability of having an outbreak in each cluster when analyzing univariate data. I build a multinomial generalized Dirichlet model to identify outbreak clusters from multivariate data which include the OTC data streams collected by the national retail data monitor (NRDM) [1] and the ED data streams collected by the RODS system [2]. The detection power of this multivariate model is evaluated by comparing it with the univariate methods and two other existing multivariate methods. We can also adjust the parameters of the model to detect the outbreaks with either same effects on different data streams or different effects.

1.3 Dissertation hypothesis

In this dissertation work, I propose a rank-based spatial clustering framework which includes the algorithm analyzing a single data stream at a time (RSC) and the algorithm analyzing multiple data streams simultaneously (MRSC). I plan to compare both RSC and MRSC algorithms to the currently existing state-of-the-art algorithms, BSS, KSS and their multivariate versions, MBSS and MKSS, respectively. My hypotheses include that 1) RSC is more computationally efficient than BSS and KSS while still being able to achieve comparable detection power and detection timeliness; 2) MRSC has better detection power than the univariate detectors when an outbreak is present in multiple data streams; and 3) the MRSC algorithm adjusted for varied outbreak effects improves the performance of detecting outbreaks having different effects on data streams.

1.4 Guide for the reader

The dissertation is organized as follows. Chapter 2 contains the background of the dissertation research. Since the proposed framework uses Bayes' Theorem to compute the posterior probability of a cluster having an outbreak, I first provide an introduction of Bayes' Theorem and priors. Then I provide an overview of the Poisson-Gamma distribution I applied in univariate model and multinomial and generalized Dirichlet distributions used in the multivariate model.

In Chapter 3, I provide a brief overview of some commonly used disease outbreak detection algorithms, which include both temporal and spatial or tempo-spatial approaches. In addition, each detection approach is categorized as either a frequentist method or a Bayesian method. I focus more carefully on reviewing spatial methods since the disease outbreak detection algorithm I propose searches for spatial clusters having outbreaks. I review both currently existing univariate detection methods and multivariate detection methods. Furthermore, the methods to calculate predicted/expected values are also described since they provide baseline values used in outbreak detection. Finally, I discuss some issues existing in current approaches and the hypothetical advantages of the proposed framework.

Chapter 4 describes the experimental domain of this dissertation work. The data sets for this study include the real syndromic data collected by the RODS system and the superimposed outbreak data. The background data for the experiments are from two data sources. One is the ED data set, which contains the counts of patient's visit to emergency rooms categorized by the chief complaints; the other is the over-the-counter (OTC) pharmaceutical sales data collected by the National Retail Data Monitor *(*NRDM). The outbreak data are simulated by outbreak simulation models. For univariate analysis, I used the linear shaped simulation model. For multivariate analysis, the outbreak data were simulated by the multivariate spatial-temporal event simulator.

Chapter 5 first introduces the proposed rank-based spatial clustering algorithm (RSC). The rank of each cluster is determined by the cluster's disease risk. Two approaches to estimate disease risks are described. Then I propose the greedy searching algorithm based on the rank of each cluster and the adjacency relationship between clusters. I adopt the Poisson-Gamma model to compute the posterior probability of having an outbreak inside each cluster. In the evaluation, several sets of experiments are done to test the hypothesis. The second part of Chapter 5 described RSC algorithm which searches for outbreak cluster from grid-based structure. The performance of the algorithm is also showed.

Chapter 6 describes a multivariate extension of RSC (MRSC). In particular, the Multinomial generalized Dirichlet model used for analyzing multiple data streams is proposed. I also described the inference of the model to compute the posterior probability. It is followed by the estimation of hyper-parameters of the model. Then I also describe how to adjust the model parameters to detect outbreaks with either same effects on multiple data streams or different effects. I illustrate the results of five sets of experiments to demonstrate the algorithm performance in the evaluation sub-chapter.

Finally, Chapter 7 contains conclusions and suggestions for future research.

Table 1.1 lists the notations which are used in the dissertation.

Table 1.1 List of notations

Notation	**Notation represents**
H	A general hypothesis
H_1	Alternative hypothesis
H_0	None hypothesis
$Poisson(k;\lambda)$	A Poisson distribution with parameter k and mean λ
$Gamma(\alpha,\beta)$	A Gamma distribution with shape parameter α and rate parameter β
$MultiNom(p_k;n)$	A Multinomial distribution with parameters $p_k > 0$ for $k = 1,2,\dots,K$ and $\sum_{k=1}^{K} p_k = 1$; $n > 0$ is number of trials
$GD(\boldsymbol{\alpha};\boldsymbol{\beta})$	A generalized Dirichlet distribution with parameterse $\boldsymbol{\alpha} = (\alpha_1,\dots,\alpha_K)$ and $\boldsymbol{\beta} = (\beta_1,\dots,\beta_K)$
D	A single time series data
$\boldsymbol{D}$	Data with $T \times K$ dimensions
T	The length of an available time series
K	The number of data streams in analysis
Z	The sub-region (cluster) in test; or the three dimensional cylinder (two in space and on in time) in spatial temporal scan statistics.
G	The entire study region
z	The variable used to represent any study area in G
$c_{s,t}$	The observed value in area s, time t; $c_{z,t}^{k}$ is used in multivariate analysis where k represents data stream k
$c_{z,t}^{k}$	The observed value in area z, time t and data stream k; used in multivariate analaysis
$b_{z,t}$	The expected value in area z, time t; $b_{z,t}^{k}$ is used in multivariate analysis where k represents data stream k
$b_{z,t}^{k}$	The expected value in area z, time t and data stream k; used in multivariate analysis
δ	Used in different outbreak simulation models to represent the outbreak strength

2.0 Background

Disease-outbreak detection is an important application domain in anomaly detection. The term “biosurveillance” denotes disease surveillance practiced by public health organizations and many other organizations that monitor for disease, such as hospitals, agribusinesses, and zoos [5]. “Electronic biosurveillance” refers to the systematic collection and automated analysis of electronically available data with the intent of detecting outbreaks of disease rapidly [15]. These electronic data include information related to emergency department (ED) visits, over-the-counter (OTC) medication sales, school or work absentees, water quality records. The goal of surveillance of these data feeds for disease outbreaks is to identify or/and characterize outbreaks rapidly with few false alarms.

The proposed spatial outbreak-detection framework is a Bayesian approach that assumes the data follow the Gamma-Poisson distribution and the Multinomial-generalized-Dirichlet distribution for univariate and multivariate analysis, respectively. In this chapter I first provide background knowledge about Bayesian theorem and choice of data distributions in Bayesian inference. In addition, since the procedure before applying most spatial/tempo-spatial algorithms is to compute the baselines of background time series data, I will then briefly describe the methods for baseline calculation.

2.1 Bayesian framework

2.1.1 Bayes' Theorem

Let H be a hypothesis and D denote some available evidence in data. We are often interested in knowing the posterior probability of H in light of D, that is $P(H|D)$. Assume we can estimate the likelihood $P(D|H)$. Frequently such likelihood are derived from a model that represents the probability that H generates D. A Bayesian approach requires the specification of a prior probability of H, which is our belief in H before seeing data D. Equation 2.1 is the well-known application of the Bayes' rule to derive $P(H|D)$.

$$P(H|D) = \frac{P(D|H)P(H)}{\sum_{H\prime \in \mathrm{H}} P(D|H')P(H')} \qquad (2.1)$$

where the sum is taken over all hypotheses H' in a set H that are modeled as having a non-zero prior probability.

2.2 Priors

In Bayesian statistical inference, a prior probability distribution, often called simply the prior, of an uncertain quantity p (for example, suppose p is the proportion of voters who will vote for the politician named Smith in a future election) is the probability distribution that would express one's uncertainty about p before the "data" (for example, an opinion poll) is considered [16]. The unknown quantity may be a parameter or latent variable.

As shown in the above chapter, Bayes' theorem multiplies the prior by the likelihood and then normalizes, to get the posterior probability, which is the conditional distribution of the uncertain quantity given the data. A prior is either the purely subjective assessment of an experienced expert or a non-informative distribution. People will usually choose a conjugate if they can, to make calculation of the posterior distribution easier.

To distinguish the parameters of prior distributions from the parameters of data models, the former are often called hyper-parameters [17]. For instance, if one is using a Gamma distribution to model the parameter λ of a Poisson distribution, then λ is a parameter of the Poisson distribution and α and β are parameters of the prior distribution, Gamma, hence hyper-parameters.

Informative priors express specific, definite information about a variable. One of the methods to assess the prior information is called the empirical Bayes method which utilizes the data to inform the prior distribution. It assumes a prior distribution for an unknown parameter θ, the distribution of θ, which we write as $p(\theta)$, has its own parameters, referred to as hyper-parameters. The hyper-parameters can either be assumed to be known, for example, by assessing expert opinions or be estimated by using methods such as maximum likelihood or method of moment matching.

Non-informative priors or "uninformative priors" express vague or general information about a variable. They are actually objective priors, i.e. ones not subjectively elicited. Non-informative priors can express "objective" information such as "the variable is positive" or "the variable is less than some limit". The simplest and oldest rule for determining a non-informative prior is the principle of indifference, which assigns equal probabilities to all possibilities.

2.3 Several statistical distributions

2.3.1 Gamma and Poisson distribution

In probability theory and statistics, the Poisson distribution is a discrete probability distribution that expresses the probability of a number of events occurring in a fixed period of time if these events occur with a known average rate and independently of the time since the last event [18].

With the discrete Poisson model, the number of cases within a time period (e.g., one day) is Poisson-distributed. The probability that there are exactly k cases ($k \geq 0$) is equal to $f(k;\lambda) = \frac{\lambda^k e^{-\lambda}}{k!}$, where λ is a positive real number, equal to the expected number of occurrences that occur during the given interval.

In Bayesian inference, the conjugate prior for the rate parameter λ of the Poisson distribution is the Gamma distribution which is $\lambda \sim Gamma(\alpha, \beta)$. If the goal of the detection is to respond to the elevated number of cases, then we presume the prior follows $\lambda \sim Gamma(m\alpha, \beta)$ in the alternative hypothesis where $m > 1$ against the non-hypothesis where $\lambda \sim Gamma(\alpha, \beta)$. This distribution model is used in the Bayesian spatial scan statistic algorithm proposed by Neill et. al. [10].

2.3.2 Multinomial distribution

In probability theory, multinomial distribution is a generalization of binomial distribution. It can be considered a categorical distribution, where each trial results in exactly one of some fixed finite number k of possible outcomes, with probabilities

$p_1, p_2, \ldots, p_K$ (so that $p_k > 0$ for $k = 1,2, \ldots, K$ and $\sum_{k=1}^{K} p_k = 1$), and there are n independent trials[19].

The probability mass function of the multinomial distribution is:

$$f(x_1, \ldots, x_K; n, p_1, \ldots, p_K) = \begin{cases} \frac{n!}{x_1! \cdots x_K!} p_1^{x_1} \cdots p_K^{x_K} & when \sum_{k=1}^{K} x_k = n\,, \\ 0 & otherwise, \end{cases} \tag{2.2}$$

for non-negative integers $x_1, \cdots, x_K$.

The expected number of times for the outcome k to be observed over n trials is $E(X_k) = np_k$. The covariance matrix is as follows. Each diagonal entry is the variance of a binomially distributed random variable, and is therefore $Var(X_k) = np_k(1 - p_k)$. The off-diagonal entries are the covariances: $Cov(X_k, X_l) = -np_k p_l$ for i, j distinct.

One of the most common of multinomial distribution involves drawing cards. Suppose a card is drawn randomly from an ordinary deck of playing cards, and then put back in the deck. This exercise is repeated five times. What is the probability of drawing 1 spade, 1 heart, 1 diamond, and 2 clubs? To solve this problem, we apply the multinomial formula. We know the following: 1) the experiment consists of 5 trials, so $n = 5$; 2) the 5 trials produce 1 spade, 1 heart, 1 diamond, and 2 clubs; so $n_1 = 1$, $n_2 = 1$, $n_3 = 1$, and $n_4 = 2$; 3) on any particular trial, the probability of drawing a spade, heart, diamond, or club is 0.25, 0.25, 0.25, and 0.25, respectively. Thus, $p_1 = 0.25$, $p_2 = 0.25$, $p_3 = 0.25$, and $p_4 = 0.25$. We plug these inputs into the multinomial formula, $P = \frac{n!}{n_1! n_2! n_3! n_4!} p_1^{n_1} p_2^{n_2} p_3^{n_3} p_4^{n_4} = 0.05859$, and get our answer.

Multinomial distribution can also be applied in the field of public health surveillance. For example, pneumonia patients with respiratory symptoms will seek treatment through different ways. Some will probably visit the emergency department, some will purchase an OTC medication and others may visit their physicians, call the nurse or wait the sickness out. To model the infected population by using multinomial distribution, one can compute p_1 as the probability of people visiting the ED, p_2 as the probability of people purchasing OTC medication and p_3 as the probability of other behaviors, assuming the people who visit ED will not purchase OTC medications within the same time interval and vice versa. The categorical data collected by a syndromic surveillance system can be another example of where multinomial distribution can be used. If the surveillance system collects the data of patient visits to Emergency Department and categorizes each visit in terms of the syndrome a patient may have, such as respiratory, gastrointestinal, constitutional or others, these categorical data can be considered following a multinomial distribution. To model this data set, one can compute p_1 as the probability of an ED patient having reparatory syndrome and p_2 as the probability of the patient having constitutional syndrome and so on. If the syndrome of a patient have cannot be recognized, one can use p_K to model a category of others or unknown. Generally, p_k is the probability of an ED patient having syndrome k and $\sum_{k=1}^{K} p_k = 1$.

2.3.3 Dirichlet distribution

In probability and statistics, the Dirichlet distribution, often denoted $Dir(\alpha)$ is a family of continuous multivariate probability distributions parameterized by a vector α of positive real numbers. It is the multivariate generalization of the beta distribution. Dirichlet distributions are very often used as prior distributions in Bayesian statistics, and in fact the Dirichlet distribution is the conjugate prior of the categorical and multinomial distribution. That is, its probability density function returns the belief that the probabilities of *K* rival events are x_i given that each event has been observed $\alpha_i - 1$ times.

The support of the Dirichlet distribution (i.e. the set of values for which the density is non-zero) is a *K*-dimensional vector of real numbers in the range (0, 1), all of which sum to 1. These can be viewed as the probabilities of a *K*-way categorical event. Another way to express this is that the domain of the Dirichlet distribution is itself a probability distribution, specifically a *K*-dimensional discrete distribution. Note that the technical term for the set of points in the support of a *K*-dimensional Dirichlet distribution is the open standard *(K-1)-simplex*, which is a generalization of a triangle, embedded in the next-higher dimension. For example, with *K* = 3, the support looks like an equilateral triangle embedded in a downward-angle fashion in three-dimensional space, with vertices at (1,0,0),(0,1,0) and (0,0,1), i.e. touching each of the coordinate axes at a point 1 unit away from the origin.

The Dirichlet distribution of order $K \geq 2$ with parameters $\alpha_1, \cdots, \alpha_K > 0$ has a probability density function is given by

$$f(x_1, \cdots, x_K; \alpha_1, \cdots, \alpha_K) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} x_k^{\alpha_k - 1} \tag{2.3}$$

for all $x_1, \cdots, x_{K-1} > 0$ satisfying $x_1 + \cdots + x_{K-1} < 1$, where $x_K = 1 - x_1 - \cdots - x_{K-1}$. The density is zero outside this open (K-1)-simplex.

The normalizing constant is the multinomial beta function, which can be expressed in terms of the gamma function where $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_K)$:

$$B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma(\sum_{k=1}^{K} \alpha_k)} \tag{2.4}$$

Let $\boldsymbol{X} = (X_1, \cdots, X_K) \sim Dir(\boldsymbol{\alpha})$, meaning that the first $K - 1$ components have the above density and $X_K = 1 - X_1 - \cdots - X_{K-1}$. Define $\alpha_0 = \sum_{k=1}^{K} \alpha_k$. Then $E(X_k) = \frac{\alpha_k}{\alpha_0}$ and $Var(X_k) = \frac{\alpha_k(\alpha_0 - \alpha_k)}{\alpha_0^2(\alpha_0 + 1)} = \frac{E(X_k)(1 - E(X_k))}{\alpha_0 + 1}$. The covariance between X_j and X_m is always negative in that $Cov(X_j, X_m) = \frac{-\alpha_j \alpha_m}{\alpha_0^2(\alpha_0 + 1)}$.

When $\alpha_k \to 0$, the distribution becomes non-informative. The means of all the p_k stay the same if all α_k are scaled with the same multiplicative constant. The variances will, however, get smaller as the parameters α_k grow. The pdfs of the Dirichlet distribution with certain parameter values are shown in Figure 2.1 [20].

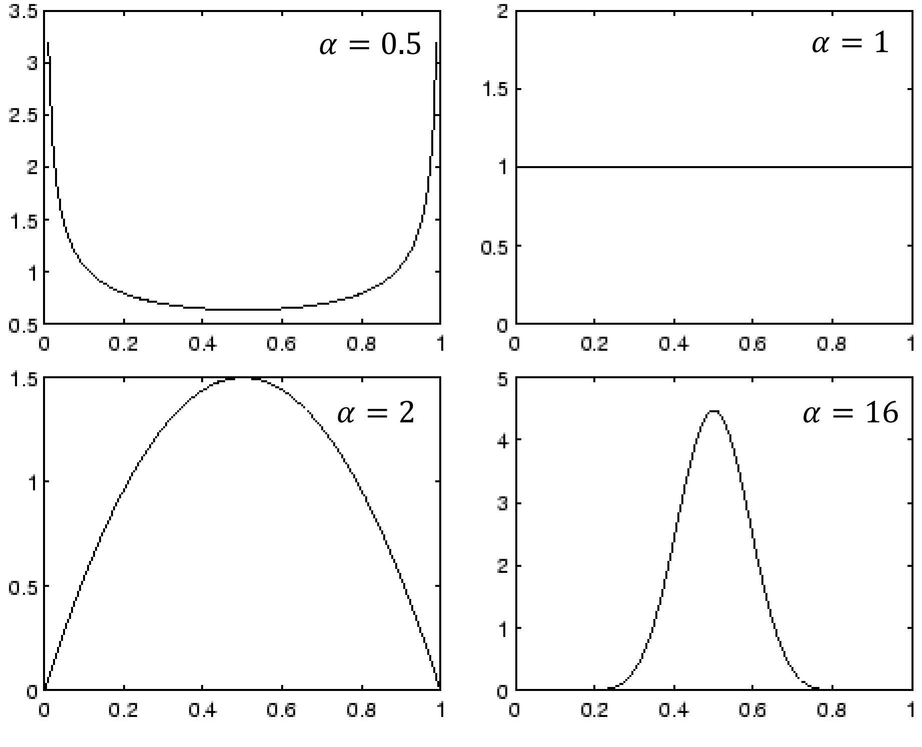


Figure 2.1 Plots of one component of a two dimensional Dirichlet distribution. The parameters are chosen such that $\alpha = \alpha_1 = \alpha_2$ with the values for α shown on each individual image. Because both the parameters of the distribution are equal, the distribution of the other component will be exactly the same.

The Dirichlet distribution is conjugate to the multinomial distribution in the following sense: if $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_K)$ and β_k is the number of occurrences of n points from the discrete distribution on $(1, \ldots, K)$ defined by $\boldsymbol{X}$ and $\boldsymbol{X} \sim Multinomial(\boldsymbol{X})$, then $\boldsymbol{X}|\boldsymbol{\beta} \sim Dir(\boldsymbol{\alpha} + \boldsymbol{\beta})$. This relationship is used in Bayesian statistics to estimate the hidden parameters, *X*, of a categorical distribution (discrete probability distribution) given a collection of n samples. Intuitively, if the prior is represented as $Dir(\boldsymbol{\alpha})$, then $Dir(\boldsymbol{\alpha} + \boldsymbol{\beta})$ is the posterior following a sequence of observations with histogram $\boldsymbol{\beta}$.

Several researchers have assigned a Dirichlet distribution as a prior to the parameter vector of a multinomial distribution. For example, Novick and Grizzle in [21] applied Bayesian analysis by using the Dirichlet density as the prior on the categorized data collected from an on-going experiment to compare the relative efficacy of four operative treatments for ulcer. In addition, Lochner and Basu [22] have used a Dirichlet prior density for life-testing situations with either complete or censored data. Ferguson in [23] has studied a “Dirichlet process” in which for any arbitrary partitioning $P(A_1, \cdots, A_m)$ of the sample space of the parent population, $P(A_1, \cdots, A_m)$ has a Dirichlet distribution.

2.3.4 Generalized Dirichlet distribution

Although some studies in Bayesian inference using Dirichlet distribution to model priors of multinomial distributions have produced useful results, some researchers have objected to the use of a Dirichlet prior density in some situations [24][25][26]. For example, when $\boldsymbol{X} = (X_1, X_2, \cdots, X_K)$ has a Dirichlet distribution, any two random variables in $\boldsymbol{X}$ will be negatively correlated. However, in some practical cases, two random variables may be positively correlated, and hence the Dirichlet distribution will not be a reasonable choice to be a prior distribution in Bayesian analysis. Generalized Dirichlet distribution has a more general covariance structure than Dirichlet distribution [27]. Connor and Mosimann [28] used the concept of complete neutrality to generalize the Dirichlet distribution.

A random vector $\boldsymbol{X}$ is said to be completely neutral if $(X_1, X_2, \cdots, X_j)$ is independent of $(X_{j+1}, X_{j+2}, \cdots, X_K)/V_j$ for all $j < k$, where $V_j = 1 - X_1 - X_2 - \cdots - X_j$. Let $Z_1 = X_1$, and let $Z_j = X_j/V_{j-1}$ for $j = 2,3, \cdots, K$. When the Z_j are independent, then $\boldsymbol{X}$ is also completely neutral. Connor and Mosimann supposed that each Z_j has a beta distribution with parameters α_j and β_j, and derived the density function for the generalized Dirichlet distribution as follows:

$$f(\boldsymbol{x}) = c \prod_{k=1}^{K-1} x_k^{\alpha_k - 1}(1 - x_1 - \cdots - x_k)^{\beta_k - \alpha_{k+1} - \beta_{k+1}} \tag{2.5}$$

for $x_1 + x_2 + \cdots + x_{K-1} < 1$ and $x_j > 0$ for $j = 1,2, \cdots, K$, where

$c = [\prod_{k=1}^{K-1} B(\alpha_k, \beta_k)]^{-1} x_K^{\beta_{K-1} - 1}$ and $B(\alpha_j, \beta_j) = \Gamma(\alpha_j)\Gamma(\beta_j)/\Gamma(\alpha_j + \beta_j)$ is the Beta function for $j = 1,2, \cdots, K - 1$, and $\alpha_K = 1, \beta_K = 0$.

In a Dirichlet distribution, X_j and X_m are always negatively related (recall $Cov(X_j, X_m) = \frac{-\alpha_j \alpha_m}{\alpha_0^2(\alpha_0 + 1)}$). However, while in a generalized Dirichlet distribution, X_1 is always negatively correlated with the other random variables but X_j and X_m can be positively correlated for $j, m > 1$[26]. If there exists some $j < m$ such that X_j and X_m are positively (negatively) correlated, than X_j will be positively (negative) correlated with X_n for all $n > j$[27]. Since the generalized Dirichlet distribution has a more general covariance structure than the Dirichlet distribution, this makes the generalized Dirichlet distribution more practical and useful.

3.0 Related work

In this chapter, I provide a brief overview of some commonly used disease outbreak detection algorithms, which include both temporal and spatial approaches. In addition, each detection approach is categorized as either a frequentist method or a Bayesian method. I focus more carefully on reviewing spatial methods since the disease outbreak detection algorithm I propose searches for spatial clusters having outbreaks.

From perspective of the data dimensions an algorithm analyzes, existing disease outbreak-detection algorithms can be categorized as temporal detection algorithms, spatial detection algorithms and spatial-temporal detection algorithms. Temporal methods operate on aggregate data that are measured only with respect to the time to find unusual spikes. Spatial methods involve accumulating data over some time interval, removing the time information, and then searching for areas of unusually high incidences of events. Spatial-temporal methods use spatial and temporal data to look for areas of unusually high incidences of events.

3.1 Temporal detection methods

Temporal analysis using time series algorithms is the most popular approach due to its relatively simple handling of a sequence of data points (i.e. counts) aggregated at a certain amount of time interval (e.g. one day) and a certain census tract (e.g. zip code or street group). Time series algorithms such as moving average (*e.g.,* Exponentially Weighted Moving Average [3]), control charts (e.g., cumulative sum (CuSUM) [4]), adaptive linear regressions [5] (e.g., Recursive Least Squares), the Bayesian change-point detector [6], and the Wavelet Anomaly Detector (WAD) [7], are commonly used in biosurveillance systems.

3.2 Spatial and tempo-spatial detection methods

Researchers have recently developed spatial or tempo-spatial algorithms to take spatial distribution into account, in the belief that the additional spatial distribution information allows detection algorithms to better detect and localize the outbreaks caused by infectious but non-contagious disease agents (*e.g.,* aerosol release of *B. anthracis*, water borne diseases caused by pathogenic microorganisms, etc.), which typically spread in an aggregated group of geographic areas. Moreover, spatial approaches are also desired to analyze (either retrospectively or prospectively) geographical patterns of non-infectious syndromes such as infant death [8], and of prostate cancer survival data [9] as well as data of other types. Essentially, the common use of a spatial algorithm is not limited to outbreak detection, but to test whether there are significant aberrancies correlated with geographical distributions.

3.2.1 Frequentist approaches

Spatial scan statistic (KSS) proposed by Kulldorff et al. is one of the state-of-the-art algorithms used to search for geographical distributions of clusters possibly having outbreaks. KSS scans the region of interest for clusters using circular or elliptic windows in different sizes and locations. The areas within a scanning window are considered a potential cluster. This algorithm finds the cluster with the highest likelihood ratio of having an outbreak in the cluster (H_1) vs. no outbreaks (H_0). At the end, KSS applies Monte Carlo Simulation to test the significance of an identified cluster.

Flexible spatial scan statistic (FleXScan) is an improvement over KSS in that it relaxes the constraint on cluster shape [29]. As I described in the above paragraph, the purely spatial scan statistic (KSS) imposes a circular window Z on each centroid of regions for each of the time intervals. For each of centroids, the radius of the circle is varied from zero up to a pre-set maximum radius, for example, the window never includes more than 50% of the study region (or total population at risk). In another way, we can use a pre-set maximum number of regions K to be included in the cluster as an upper-bound of the radius. If the base contains the centroid of a region, then that whole region is included in the base. In total, a very large number of different but overlapping circular bases are created, each with a different set of neighboring regions and each being a possible candidate area containing a disease outbreak. On the other hand, a flexible space-time scan statistic imposes a prismatic window with an arbitrarily shaped base. For any given region s, it creates the set of arbitrarily shaped bases consisting of k connected regions ($1 \leq k \leq K$) including s. To avoid detecting a cluster of unlikely peculiar shape, the connected regions are restricted as the subset of the K-nearest neighbors to the region s, where $K = 1$ implies the region z itself. By this way, FSS searches for flexible shaped clusters. However, this algorithm has a higher time complexity than KSS, which makes it less practical for processing large data sets.

Upper level set scan statistic (ULS) is also a scan statistic algorithm, but searches from a reduced parameter space [30]. By estimating the elevated response, $G_z = \frac{Y_z}{A_z}$, for each study area z, it searches clusters from a collection of subsets, each of which is called an upper level set (ULS), $U_g = \{z: G_z \geq g\}$. g is a threshold value to determine a ULS. All the connected areas of all possible upper level sets make up the search space of the ULS algorithm. Like KSS, ULS computes the likelihood ratio of having an outbreak in each cluster. This algorithm is faster than KSS because it reduces the searching space, but it still needs the randomization test to decide the significance of each cluster.

Machine learning clustering methods Other algorithms, such as *Risk-adjusted Nearest Neighbor Hierarchical Clustering* (RNNH) [12] and *support vector machines* (SVMs) [13], utilize traditional clustering approaches proven to be computationally efficient and provide alternative methodologies other than scan statistic algorithms. RNNH was first developed for crime hotspot analysis [31]. It is based on the well-known nearest neighbor hierarchical clustering method, combining the hierarchical clustering capabilities with kernel density interpolation techniques. In other words, it dynamically adjusts the threshold distance inversely proportional to some density measure of the baseline factor (e.g., the population). SVM is a systematic approach with well-defined optimization formulations and can be solved using well-established computational methods. One of SVM-based approaches, data description and novelty detection (DDND) is particularly relevant to anomaly detection and was used in tempo-spatial data analysis in infectious disease outbreak detection [13]. The authors of [13] developed a risk-adjusted variation based on the ideas similar to those in RNNH. They compute the kernel density estimations using the baseline points and they adjust with parameter q in the Gaussian kernel function based on such density estimations. The basic intuition is as follows: When the baseline density is high, they use higher q, which makes it harder for points to cluster together. However, each method evaluated in the study of [13] has control parameters that can be set to influence the number and the shape of the hotspots. In their study, the authors tried the experiment with various settings first and then choose the best settings for each method examined, so the comparison is far from complete. Thus, it is still a challenge for these approaches to automatically determine the required control parameters [13]. Other concerns include that these approaches are based on analyzing the data points in terms of case time and exact geo-location which may not be available because of confidential issue.

3.2.2 Bayesian approaches

Bayesian spatial scan statistic (BSS) employs a rectangular window (aligning with *x,y*-axes) to search for clusters. The rectangular scanning window is composed of one or more grid cells in an $m \times m$ grid covering the whole area of interest. The algorithm identifies the cluster with the highest posterior probability of having an outbreak, which is computed using a Poisson-Gamma model. Unlike KSS, BSS does not require a randomization test to determine cluster significance.

BARD (Bayesian aerosol release detector) analyzes both medical surveillance data and meteorological data for early detection and characterization of outdoor releases of *B.anthracis* [32]. The approach is general and could be applied to outbreaks due to other biological agents that can be disseminated by outdoor aerosol release. BARD is the first algorithm to integrate meteorological data and a model of atmospheric dispersion into the analysis of medical surveillance data. BARD computes the posterior probability of a release given these data and a posterior distribution over the release location, quantity, and time. BARD is also used for simulating outbreaks due to inhalational anthrax. Simulated anthrax cases generated by BARD were used in [33].

PANDA (the population-wide anomaly detection and assessment algorithm) uses a causal Bayesian network to model spatio-temporal patterns of a non-contagious disease in an entire population of people [14] [34][35]. In [34], each person in the population being monitored for an outbreak is explicitly modeled using a sub-network. In particular, each person in the population is represented with a six-node network structure that includes disease status, patient symptoms and other personal information while avoiding any information that could personally identify the individual (e.g., name, social security number, and home street address). The primary clinical information about each person is whether he or she presented to the ED with a chief complaint of interest (e.g., a cough). The sub-networks are connected through a common set of nodes that represent the disease outbreak conditions, such as the hypothesized location and time of release of anthrax spores. Given current data about individuals in the population, techniques for inference and modeling are applied on a Bayesian network to derive the posterior probabilities of outbreak diseases in the population.

In the original PANDA model, the algorithm was designed to monitor only ED chief complaint data. It was extended by Wong and his colleagues to simultaneously monitor data sources of different granularity – specifically aggregated regional counts for OTC sales and multivariate ED records for individual patients [14].

Jiang and Cooper in [35] further extended the original PANDA-CDCA (PC) algorithm to spatial-temporal system, PANDA-CDCA-Temporal-Spatial (PCTS) algorithm. They proposed the Bayesian Network Spatio-temporal (BNST) model to incorporate spatial information and temporal information by using two extra nodes SUB and Y representing the outbreak sub-region and the number of days into the outbreak if there is an outbreak. Their study support that PCTS provides improved disease outbreak detection, relative to PC from which it was derived. Besides often detecting outbreaks earlier (at a given false alert rate), PCTS was better at maintaining a stable detection signal over time.

3.2.3 Issues of algorithms using scan statistics

Regardless of whether they use a frequentist or Bayesian approach, scan-statistic-like algorithms face some common limitations. First, they are computationally intensive due to extensive searching and/or randomization testing. This is important as in time-sensitive applications, an algorithm taking too long to complete can render its results outdated or delayed for decision makers. Since directly applying these algorithms to large data sets will probably result in computational infeasibility, Neill and Moore et al have developed a fast multi-resolution algorithm which relies on an overlap-kd tree data structure. It greatly reduced the time complexity of BSS from $O(m^4)$ to $O((mlogm)^2)$ by using approximation [36][37].

Neill also proposed an algorithm called fast subset scan (FSS) [38]. The algorithm treats event detection as a search over subsets of data records, finding the subset which maximizes some score function. Neill theoretically proved that many commonly used functions (e.g.Kulldorff's spatial scan statistic and extensions) satisfy the 'linear time subset scanning' property, enabling exact and efficient optimization over subsets. In [38], he demonstrated that proximity-constrained subset scans substantially improved the timeliness and accuracy of event detection, detecting emerging outbreaks faster than existing methods. He also found in certain cases, the unconstrained fast subset scan approach reduced to a variant of the upper level set (ULS) scan statistic. Moreover, both FSS and ULS are closely related to the rank-based clustering algorithm (RSC) originally proposed in [33]. The fast localized multiscan algorithm, one of the mostly related algorithms using FSS proposed in [38], has a time complexity of $O(n^3)$.

Secondly, certain artificial cluster shapes (e.g., circle, rectangle) used by those algorithms may not conform to true outbreak clusters which may provide inaccurate information for decision makers. As an effort to overcome this issue, some work has been done to identify flexible shapes of outbreak clusters, such as the flexible spatial scan statistic (FLeXScan) [29] and a recursive algorithm for Bayesian cluster detection using PANDA [39]. As I explained in the previous chapter, FSS increases the order of the time complexity of KSS since it searches for all the k connected areas starting from any area s within the study region with $1 \leq k \leq K$ and K is an upper-bound of cluster size. The recursive Bayesian cluster detection algorithm searches for clusters based on all the rectangular clusters as created in BSS. It recursively searches for finer clusters inside each rectangular cluster found by BSS and exhaustively joins any two rectangles and updates the score if the joint cluster has a higher score. The worst case time complexity of this algorithm is $O(m^8)$ where m is the grid size [39].

3.3 Multivariate methods

In disease surveillance, there are often more than one data sources for which we wish to do surveillance. If each data set is analyzed separately rather than collectively, the statistical power to detect an outbreak that is present in all data sets may suffer due to the data noise alone. Another major reason for taking a multivariate approach to disease surveillance is that no single data source captures all the individuals in the outbreak. Depending on the disease, some will go to their pharmacy and buy an over-the-counter medication; some will call their physician or a nurse hot-line, while others may visit their regular physician, go to a hospital emergency room, or call an ambulance [40]. On the other hand, if the data sets are simply added together by taking the sum of the values in each data set, then a present signal in one data set with relatively low counts may be overwhelmed by the random noise in another data set with relatively high counts.

Various multivariate approaches exist.

Multivariate scan statistics (MKSS), proposed by Kulldorff et al., is an extension of the spatial and space-time scan statistic that simultaneously incorporates multiple data sets into a single likelihood function. This is done by defining the combined log likelihood as the sum of the individual log likelihoods for those data sets for which the observed case count is higher (or lower) than the expected.

We can write down this model in mathematical notation. Let $b_Z = \sum_{z,t \in Z} b_{zt}$ be either the population or the expected number of cases in cylinder Z including location s during time period t, $B = \sum_{z,t} b_{zt}$ is the total population/expected cases, $c_Z = \sum_{z,t \in Z} c_{zt}$ is the number of observed cases in cylinder Z, $C = \sum_{z,t} c_{zt}$ is the total number of observed cases, and $k = 1,..,K$ represents the index of the data types. For the Poisson model, let

$$LR_k(high, Z) = \left(\frac{c_Z}{b_Z}\right)^{c_Z} \left(\frac{C - c_Z}{C - b_Z}\right)^{C - c_Z} I(c_Z > b_Z) \tag{3.1}$$

and

$$LR_k(low, Z) = \left(\frac{c_Z}{b_Z}\right)^{c_Z} \left(\frac{C - c_Z}{C - b_Z}\right)^{C - c_Z} I(c_Z < b_Z) \tag{3.2}$$

be the likelihood ratio for high and low clusters, respectively, for cylinder Z in data set k. The test statistic can now be written as

$$\mathcal{Z} = \max_Z \max(\sum_k LLR_k(high, Z), \sum_k LLR_k(low, Z)) \tag{3.3}$$

If there is only interest in one of these two, only one of the sums in the second step above is used. That is, when searching only for high clusters which is often the scenario for outbreak detection, $\mathcal{Z} = \max_Z \sum_k LLR_k(high, Z)$.

In order to adjust for the multiple testing inherent in the many cylinder location and sizes evaluated in the same way, a randomization test is used. This method is computer intensive because of the nature of the scanning window and the need to evaluate the test statistic for 999 or more random replicas of the data set [40]. Another negative aspect of this method is that a data source with a large count may mask data sources with smaller counts. As an alternative, Burkom proposes calculating the log likelihood ratio for each data source and summing these ratios to form the scan statistic [41]. The technique of Edgington's consensus method that Burkom suggest we use assumes independence among the data sources [42].

Multivariate Bayesian scan statistic (MBSS) is an extension of the Bayesian spatial scan statistic approach [43]. MBSS integrates prior information and observations from multiple data streams in a principled Bayesian framework, computing the posterior probability of the event (e.g., disease outbreaks).

Mathematically, we can use $\boldsymbol{D}$ to represent a data set $\boldsymbol{D}$ consisting of multiple data streams D_k, for $k = 1, \dots, K$. Each data stream consists of spatial time series data collected at a set of spatial locations s, for $z \in G$. For each stream D_k and location z, we have a time series of counts $c_{z,t}^k$, where $t = 0$ represents the current time step and $t = 1, \dots, T$ represent the counts form 1 to T time steps ago respectively. By assuming a Gamma-Poisson model the posterior probability of having an event E_k happening can be written as the following if we drop the sub- and superscripts and simply write $c = c_{z,t}^k$, $b = b_{z,t}^k$, $q = q_{z,t}^k$, $x = x_{z,t}^k$, $\alpha = \alpha_z^k$ and $\beta = \beta_z^k$, where $b_{z,t}^k$ is the expected value for area s at time t for data type k, and $c_{z,t}^k = q_{z,t}^k b_{z,t}^k$. α_z^k and β_z^k are the shape and rate parameters of Gamma distribution, respectively.

$$P(c|b,x,\alpha,\beta) = \int P\big(q \sim Gamma(x\alpha,\beta)\big)P\big(c \sim Poisson(qb)\big)dq \tag{3.4}$$

$$= \frac{\beta^{x\alpha} b^c}{\Gamma(x\alpha)c!}\int q^{x\alpha+c-1} e^{-(\beta+b)q} dq = \frac{\beta^{x\alpha} b^c \Gamma(x\alpha+c)}{(\beta+b)^{x\alpha+c}\Gamma(x\alpha)c!}$$

This approach assumes that the counts $c_{z,t}^k$ are conditionally independent given the values of $b_{z,t}^k$, $x_{z,t}^k$, $\alpha_z^k$and β_z^k, the likelihood of the entire data set $\boldsymbol{D} = \{c_{z,t}^k\}$ for a given set of effects $\boldsymbol{X} = \{x_{z,t}^k\}$ is the product of these conditional probabilities:

$$P(\boldsymbol{D}|\boldsymbol{X}) = \prod_{z,t,k} P(c_{z,t}^k|b_{z,t}^k, x_{z,t}^k, \alpha_z^k, \beta_z^k) \propto \prod_{z,t,k}\left(\frac{\beta_z^k}{\beta_z^k + b_{z,t}^k}\right)^{x_{z,t}^k \alpha_s^k} \frac{\Gamma(x_{z,t}^k \alpha_z^k + c_{z,t}^k)}{\Gamma(x_{z,t}^k \alpha_z^k)} \tag{3.5}$$

In this expression, terms not dependent on the $x_{z,t}^k$ have been removed since these are constant for all hypotheses under consideration. For the null hypothesis H_0, it sets $x_{z,t}^k = 1$ everywhere, $P(\boldsymbol{D}|H_0) \propto \prod_{z,t,k}\left(\frac{\beta_z^k}{\beta_z^k + b_{z,t}^k}\right)^{\alpha_z^k} \frac{\Gamma(\alpha_z^k + c_{z,t}^k)}{\Gamma(\alpha_z^k)}$; for the alternative hypothesis $H_1(Z, E_k)$, the marginal probability is $P\big(D\big|H_1(Z, E_k)\big) = \sum_X P(D|X)P(X|H_1(Z, E_k))$, where the effects $x_{z,t}^k$ are dependent on the event type E_k and its magnitude.

PANDA also provides an extended version which combines information from multiple data streams [14]. The authors extend the causal Bayesian network model used in the Population-wide Anomaly detection and Assessment (PANDA) [34] to incorporate evidence from daily OTC sales data. They model, at the level of individual person, the actions that result in the purchase of OTC products, as well as admission to an ED. A prototype model for detecting an Anthrax outbreak was created by expert judgment. In addition to the nodes, *Home Zip*, *Age Decile*, *Gender*, and *outbreak strength*, three other evidence nodes representing two different data streams are incorporated as well and are called *OTC sales for Zip Code*, *Respiratory Chief Complaint When Admitted* and *ED Admission*. The parameter of some nodes were estimated from a training set consisting of one year's worth of ED patient data from the year 2000 or from one year's worth of OTC data from 2004. The parameters of other variables were obtained from U.S. Census data about the region. The probabilities of the rest nodes (whether prior or conditional) were derived as a logical function of their parents or assessed subjectively as informed by the literature or by general knowledge about the infectious diseases. Let o be the set of population-wide evidence, namely the OTC sales volume for each zip code in the county-wide region. Similarly, let e be the collective set of evidence from individual people consisting of case information from those that were recently seen in EDs in the region. The posterior probability of a disease outbreak given the OTC data ($D(o)$) and the ED data ($D(e)$) can be written mathematically as $(H_1|D(o), D(e)) = kP(D(o), D(e)|H_1)P(H_1)$ where k is the proportionality constant. After using Bayesian network inference, they derived the term as $P(D(o), D(e)|H_1) = \sum_i P(D(o)|D(e), I)P(e|I)P(I = i|H_1)$ given the evidence set I. The evaluation of the work was not provided in [12].

WSARE, an abbreviation for "what's strange about recent events", is a rule-based anomaly detection algorithm which is used to tackle the problem of early disease outbreak detection [44]. It searches over all possible one or two component rules in the data set. It determines whether the count of cases that match the rule in the test data set is significantly different from the expected count determined by the training data set. The statistical significance of each rule is determined using a Fisher's exact test on the two-by-two contingency tables. To account for multiple hypothesis testing, the p-values are adjusted using a randomization test. In a later version of the algorithm [45], the authors considered determining the baseline using a Bayesian network rather than directly using the counts from the training data set. A similar rule-based study was conducted as well in [46].

Other multivariate surveillance methods have also been proposed and applied to the disease surveillance domain. One proposed by Burkom is a derivation of the spatial scan statistic. It combines multiple and disparate data sources as the covariance variables in the scan statistic [47]. However, because the scan statistic numerators are formed by adding counts of disparate sources, a data source with highly variable or relatively large counts may mask signals in sources with smaller or more stable counts. Another method is a network-based method proposed by Reis et al. In [48], he describes an epidemiological network model that monitors the relationships between health-care utilization data streams for the purpose of detecting disease outbreaks. Instead of monitoring the observed counts directly, he models the significance of the ratios between each target data stream (as numerator) and each context data stream (as denominator). However, although this method integrates information from multiple data streams, it is still a purely temporal detection method and so does not take spatial information into account; therefore, while it may be used to detect anomalous increases in the aggregate time series of the entire area being monitored, it cannot detect and pinpoint a spatial cluster of affected locations.

Table 3.1 provides an innovation timeline for all the spatial or tempo-spatial algorithms and also lists the main properties of these algorithms. Most of them have been discussed with more details in the previous sub-sections.

Table 3.1 An innovation timeline of the spatial/tempo-spatial algorithms

Time	**Algorithm (Authors)**	**Properties**						
		Frequentist	Bayesian	General	Univariate	Multivariate	Search space	Cluster shape
1997	A spatial scan statistic [8] (Kulldorf et. al.)	√			√		All circular windows	Circular
2004	Upper level set scan statistic [30] (Patil et. al.)	√			√		Stratified subsets	Flexible
2004	PANDA [34] (Cooper et. al.)		√			√	N/A	Flexible
2005	A Bayesian spatial scan statistic [10] (Neill et. al.)		√		√		All rectangular windows	Rectangular
2005	WSARE [14] (Wong et. al.)			√		√	N/A	N/A
2006	An elliptic spatial scan statistic [49] (Kulldorf et. al.)	√			√		All elliptical windows	Elliptic
2007	BARD [32] (Hogan et. al.)		√		√		N/A	Plume-shaped
2007	Mutivariate spatial scan statistic [40] (Kulldorff et. al.)	√				√	All circular/ elliptic windows	Circular/ Elliptic
2008	A flexibly shaped spatial-time scan statistic [29] (Takahashi et. al.)	√			√		All k-nearest-neighbors in the circular windows	Flexible
2008	A multi-level rank-based spatial clustering [33] (Que et. al.)		√		√		Subsets	Flexible
2010	PCTS [35] (Jiang et. al.)		√		√		Recursive	Flexible

2010	A multivariate Bayesian scan statistic (Neill et. al.)		√			√	All windows	Rectan-gular/Circular
2011	Fast subset scan [50] (Neill et. al.)			√	√	√	Subsets	Flexible
2011	Rank-bases spatial clustering [51] (Que et. al.)		√		√		Subsets	Flexible
2012	Fast subset scan [38] (Neill et. al.)			√	√	√	Subsets	Flexible

3.4 Calculation of baselines

In the above introduction of disease outbreak detection algorithms, we have paid relatively little attention to the question of how the underlying populations or baselines are obtained. However, determination of baseline is a critical issue in the performance of anomaly detection. Better identification of the real underlying pattern within data can improve the performance of detection methods by reducing false alarm rates.

In the population-based methods (such as in the spatial scan statistic [8]), we often use census data as baseline data, which gives an unadjusted population corresponding to each census tract. This population can then be adjusted for covariates using demographic data such as the distribution of patient gender or age group, giving an estimated “at-risk” population for each census tract.

The expectation-based methods make use of historical data to compute the number of cases we expect to see in each area. When the data are not complete or not rational to the population/adjusted population, it would be inappropriate using population as the baseline. For example, if we only collect the over-the-counter (OTC) sales data from partial vendors and we lack the market share information of these vendors, then using the entire population to compute the estimated OTC sales for the data we have would decrease the sensitivity of detection. Thus, we must predict the expected number of cases for each area based on its history of past counts at that location. This becomes a univariate time series analysis problem and any of the temporal detection algorithms we mentioned in Chapter 3.1 can be used compute the expected number of cases. For example, simple mean or exponentially weighted moving average methods can compute the estimated number of cases as the mean of the counts 7, 14, 21 and 28 days ago, as in Neill et al. [10]. For data sets which include strong day-of-week effects, we can stratify the data sets into subsets including data for different days of the week and apply time series algorithms on each subset.

Among the aforementioned time series algorithms, the WAD [5] algorithm is well established as one of the most effect methods of capturing seasonal effects compared to other time series algorithms [6][52]. WAD is a non-parametric algorithm using wavelet transform, suitable for non-stationary time series. It makes use of frequency decomposition in wavelet transform to predict the number of cases. By setting a proper scale of resolution (e.g., setting the scale of 6 to get 2^6=64 day frequency or setting the scale of 7 to get 2^7=128 day frequency), it is able to capture underlying seasonal trends. Another method, proposed by Kulldorff et al [53], is to compute the expected count in a given region as the total count of the entire area under surveillance, multiplied by the historical proportion of counts in that region. In addition to the methods we discussed above, there are approaches using regressions, Bayesian theories and others. Until now, accurate inference of expected counts from historical data is still an open problem.

The testing and evaluation of the algorithm detection methods either use population/adjusted population or use expected cases estimated by historical data was further discussed in the work by Siegrist et al [52] and Buckeridge et al [6].

3.5 Computational considerations

Before applying any spatial or tempo-spatial algorithm for early outbreak detection, it is necessary to consider the computational resources need to perform an algorithm. If the algorithm is to be used to analyze a very large area (e.g. multiple states) at a detailed level (e.g., zip code or street group), we need to consider whether an algorithm can achieve an efficient performance; whether it has a high computational complexity or require a randomization test. In the following paragraph, we discuss the computational complexity of two spatial scan statistics, the spatial scan statistic by Kulldorff et al (KSS) [8] and the Bayesian spatial scan statistic by Neill et al (BSS) [10].

The KSS algorithm considers a set of N distinct spatial locations in two dimensions. The number of circular regions it searches is proportional to N^2 and the number of elliptic regions (assuming the length of both semi-major and semi-minor can vary) is proportional to N^3. KSS searches for clusters either over N^2 circular scan windows [8] or searches over N^3 elliptic scan windows [49]. In addition, the algorithm runs on R replications to decide the significance of the top clusters, where R is often greater than N. This makes the algorithm have computational complexity of $O(RN^2)$ for circular clusters and $O(RN^3)$ for elliptic clusters. BSS utilizes a two dimensional $M \times M$ grid, and the number of axis-aligned rectangular regions with varied lengths, widths and centers is proportional to M^4. The searching operations within this algorithm has a computational complexity of $O(M^4)$.

3.6 The hypothetical advantages of proposed algorithms

As I mentioned in Chapter 1, my research problem is focused on rapid disease outbreak detection. I will develop a non-parameterized framework using tempo-spatial clustering algorithms. The framework includes both a univariate algorithm, the rank-based spatial clustering algorithm (RSC), and a multivariate algorithm, the multivariate rank-based spatial clustering algorithm (MRSC).

The data RSC and MRSC analyze are aggregated to some extent (e.g., a ZIP code level, a city level, etc.). They are more similar to the spatial scan statistic algorithms such as BSS/MBSS, KSS/MKSS, FSS and ULS. They are all population based algorithms rather than individual based algorithms such as PANDA-CDCA and PANDA-CDCA-Spatial-Temporal.

Compared to the spatial or tempo-spatial algorithms discussed in this chapter, both RSC and MRSC algorithms in this framework provide a methodology with lower order of time complexity in terms of cluster searching. They apply a measurement to decide which study area is more likely to have an outbreak occurring given the baseline data and currently observed data. Using the estimated risk measurement as heuristic, they apply a greedy searching mechanism to look for the cluster with the highest posterior probabilities.

RSC and MRSC are able to find clusters with irregular shapes since they do not impose any artificial shape for cluster scanning as KSS, MKSS, BSS and MBSS do. However, they require the information of adjacency relationship between any two study areas. I will discuss more about the time complexity of computing the shortest distance between any two geographic areas. Fortunately, using a hash table or querying a spatial database system can solve this problem in linear time. FSS and ULS also search for flexible-shaped clusters. Nonetheless, FSS has a higher order of time complexity than KSS which makes it computational infeasible when analyzing a large data set. RSC is similar to ULS regarding they search for clusters from reduced search spaces. However, the ULS and RSC differ in three respects: the risk estimation models used, the cluster search space and cluster significance testing. More specifically, ULS stratifies the data set into several subsets using a set of pre-defined threshold rates and looking for tessellated areas from each subset, whereas RSC creates a new cluster each time when taking the next-ranked area into consideration, which makes its search space a super set of ULS. A special case is when ULS sets a set of thresholds and each value in the set is equal to the risk rate of each geographic unit in the study. In this way, ULS finds sets of clusters identical to those RSC would find if both apply the same risk estimation model. In terms of significance testing, ULS uses frequentist randomization test whereas RSC applies Bayesian inference.

Like MBSS and MKSS, MRSC searches for clusters from multiple data streams. Particularly, it is more similar to MBSS since it also applies a Bayesian model to derive the posterior probability of having an outbreak inside each cluster. However, the data model of MRSC is different from MBSS. MRSC applies a hierarchical Multinomial-generalized-Dirichlet (MGD) distribution to model the multiple data streams simultaneously, whereas MBSS applies a hierarchical Poisson-Gamma distribution to model a single data stream and compute the joint probability. As in the preliminary study of my dissertation work, MRSC is designed to be more sensitive to the outbreak which occurs simultaneously in multiple data streams. In addition, since the multinomial distribution used in MRSC actually models the weights of each single data stream, another hypothetical advantage of MRSC is that it is more robust to the data with underlying non-disease related shifts. I will also discuss more about it in Chapter 6.0

4.0 The experimental domain

The experimental domain for my proposed research is the semi-synthetic data sets created for this study, which include the real syndromic data collected by the RODS system and the superimposed outbreak data. For univariate analysis, I used the linear shaped simulation model which was used in [10][43][51][54]. I will describe this model in details in Chapter 4.2. For multivariate analysis, the outbreak data were simulated by the multivariate spatial-temporal outbreak simulator [55], which will be introduced in Chapter 4.3.

The background data for the experiments are from two data sources. One is the ED data set, which contains the counts of patient's visit to emergency rooms categorized by the chief complaints; the other is the over-the-counter (OTC) pharmaceutical sales data collected by the National Retail Data Monitor *(*NRDM) [1]. The outbreak simulation model used for this research is called the multivariate spatial-temporal outbreak simulator. This model generates multiple data streams of outbreak data which can be used for evaluating detection algorithms used in disease surveillance systems.

4.1 Background data

Disease surveillance refers to methods relying on detection of individual and population health indicators that are discernible before confirmed diagnoses are made. During an outbreak of an infectious disease, prior to the laboratory confirmation of the disease, ill persons may exhibit behavioral patterns, symptoms, signs, or laboratory findings that can be tracked through a variety of data sources [15]. Disease surveillance is important at such times because it could detect a surge by analyzing these data sources, thus providing an early warning at the start of an outbreak as well as acting as a tool for monitoring an ongoing crisis.

4.1.1 ED data

Should a disease outbreak or an unannounced biological attack occur, the first sign could be an increase in healthcare utilization, probably by patients with relatively common symptoms, for example, anthrax-infected persons with respiratory complaints. If the first wave of patients is spread out over a large geographic area, it may present similar aberrant patterns across different public health organizations in the area, meaning real-time surveillance based on syndromes can provide one of the quickest ways to recognize and respond to many natural or unnatural disease outbreak scenarios.

Over the past decade, physicians and researchers at the RODS lab have been collecting syndromic data for surveillance. The first data type they started collecting was the chief complaints of patients during their visit to emergency rooms in Allegheny County in Pennsylvania. After a patient was registered in an emergency department, the chief complaint was automatically sent to the RODS system and classified into one of the nine syndrome groups including gastrointestinal, constitutional, respiratory, rash, hemorrhagic, botulinic, neurological and other.

The RODS system then provides a data set that contains the daily counts of patients' ED visits grouped by different syndromes and the patient home ZIP code. From this data set, I selected a 3 year period as my study period which is from Jan. 1, 2006 to Dec. 31, 2008 for this study.

4.1.2 OTC data

Over-the-counter (OTC) medication sales can serve as an early indicator of communitywide disease outbreaks as when people first get sick, they often first try OTC medicine to get well before seeking professional medical treatment [56,57,58]. Therefore, to enhance detection of natural and intentional infectious disease outbreaks, since 2003 the RODS lab has tracked OTC medication sales as well as ED data. Each OTC medication purchase is automatically classified into one of 23 categories based on the symptoms and the age group it treats; examples include anti-fever pediatric, anti-fever adult, bronchial remedy, diarrhea remedy, thermometer pediatric and thermometer adult.

4.1.3 OTC data simulation model using NRDM data

Because the existing data set collected by NRDM does not include information about patient home ZIP code for each product sale, I created a data simulation model to allocate the counts of OTC sales into patients' residential ZIP code areas. The possible application of this model is to simulate background counts for the areas with missing data and thereby allow the outbreak simulation algorithms to generate outbreak data for these areas.

To demonstrate the model, I chose to look at OTC medication purchases made by the patients living in six ZIP code areas with or without pharmacy stores (Figure 4.1). The model is illustrated in Figure 4.2. The nodes are connected by three types of arrows representing the different types of commuting. We presume: 1) people living in a ZIP code area with pharmacy stores will purchase OTC medications from those stores; and 2) people living in a ZIP code area without stores will purchase OTC medications from a) an adjacent ZIP code area that has stores (solid arrows), b) the nearest with-store ZIP code area if neither their living ZIP codes nor the adjacent has stores (dashed arrows), or c) their ZIP code areas where they work, which has stores (doubled arrows).

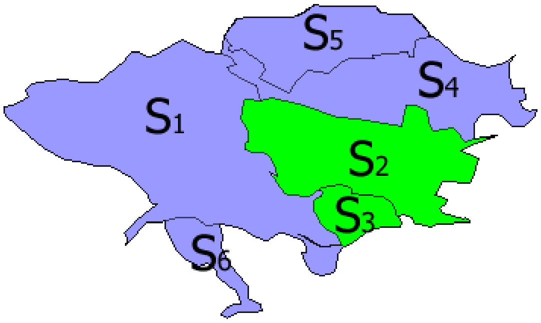


Figure 4.1 Illustration of six adjacent ZIP code areas. The green areas represent the ZIP codes with pharmacy stores and the blue ones represent the ones without.

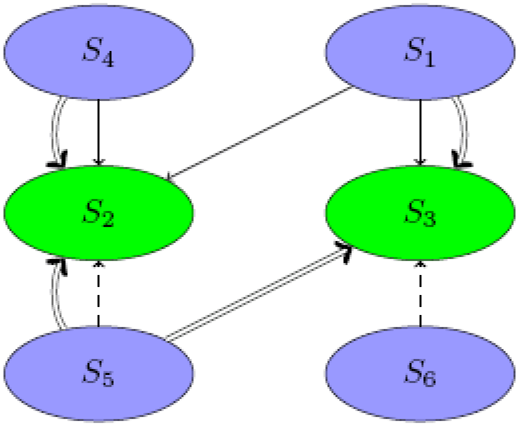


Figure 4.2 Modeling OTC medication purchases made by the patients living in the six ZIP code areas in Figure 4.1. The arrows illustrate three types of commuting in between: 1) doubled arrows represent work flows from none-store areas to with-store areas, respectively; 2) solid arrows represent the remaining population who travel to the connected with-store areas; they start from none-store areas and end at connected with-store areas; 3) dashed arrows represent those who start from a none-store area which has no adjacent areas with pharmacy stores and end at the nearest with-store area.

The simulation consists of three steps. I use $S_i \in \Phi$ to denote each of with-store area and $S_j \in \Delta$ to represent each none-store area.

First, I split each none-store node into sub-nodes so that each sub-node only has one arrow going out. In the rightmost graph in Figure 4.3, s_{ij}^{w} represents the population work flow between S_i and S_j which was collected during the 2000 census, and s_{ij} represents the remaining population in area S_i who purchased OTC medication in area S_j which is computed as proportional to the population of its target node. Mathematically, I use $W(S_i)$ to denote the set of all the working areas for the people who live in area S_i and I use the set $O(S_i)$ to represent the set of areas neighboring of area S_i other than the working areas. I represent the set of sub-nodes as $\{s_{ij}, s_{ik}^{w}: S_j \in O(S_i), S_k \in W(S_i)\}$. I compute the corresponding population for each sub-node, $\mu(s_{ij})$ and $\mu(s_{ij}^{w})$, as shown in Equation (4.1) and (4.2).

$$\mu(s_{ij}^{w}) = \theta_{ij} \qquad (4.1)$$

$$\mu\left(s_{ij}\right) = \frac{\mu\left(S_j\right)}{\sum_{S_j \in O(S_i)} \mu\left(S_j\right)} \left(\mu(S_i) - \sum_{S_j \in W(S_i)} \mu(s_{ij}^{w})\right) \tag{4.2}$$

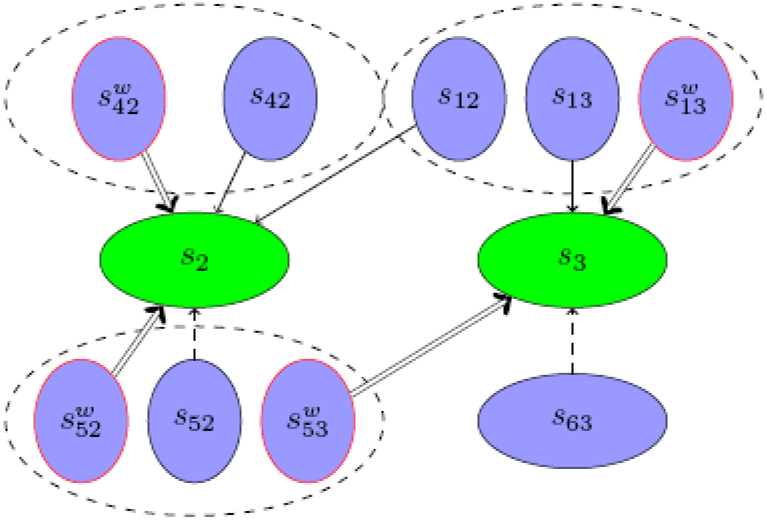


Figure 4.3 Splitting nodes

Second, for each with-store node which has multiple arrows coming in, I adjust its counts and re-allocate them to all the other nodes which have arrows coming in. I introduce the randomness by assuming the counts for all the incoming nodes and the node itself follow a Multinomial distribution. The parameters of $MultiNom(p_j, \{p_{ij}^{w}\}, \{p_{kj}\})$ are estimated as ratios of populations (Equation (4.3)-(4.5)) where $S_i \in \Delta$, $S_j \in \Phi$. I use $V(S_j)$ to represent the set of areas which have people work in S_j and $I(S_j)$ to represent the set of the rest of the incoming none-store nodes.

$$p_j = \frac{\mu(S_j)}{\mu\left(S_j\right) + \sum_{S_i \in V(S_j)} \mu(s_{ij}^{w}) + \sum_{S_i \in I(S_j)} \mu(s_{ij})} \tag{4.3}$$

$$p_{ij}^{w} = \frac{\mu(s_{ij}^{w})}{\mu\left(S_j\right) + \sum_{S_i \in V(S_j)} \mu(s_{ij}^{w}) + \sum_{S_i \in I(S_j)} \mu(s_{ij})} \tag{4.4}$$

$$p_{ij} = \frac{\mu(s_{ij})}{\mu\left(S_j\right) + \sum_{S_i \in V(S_j)} \mu(s_{ij}^{w}) + \sum_{S_i \in I(S_j)} \mu(s_{ij})} \tag{4.5}$$

If we use $C_j(t)$ to represent the total counts for a with-store node S_j on day $t, t = 1,2,\cdots,T$, then $C_j(t) = x_j(t) + \sum_{S_i\in V(S_j)} x_{ij}^w(t) + \sum_{S_i\in I(S_j)} x_{ij}(t)$ where $[x_j(t), \{x_{ij}^w(t)\}, \{x_{ij}(t)\}] \sim MultiNom(p_j, \{p_{ij}^w\}, \{p_{kj}\})$.

In the third step, I combine the sub-nodes back into the original none-store node by summing the allocated counts together, as in Equation (4.6), where $\widetilde{C_\iota}(t)$ represents the simulated counts in area S_i. The simulated counts for the with-store node are then adjusted in Equation (4.7).

$$\widetilde{C_\iota}(t) = \sum_{S_j\in W(S_i)} x_{ij}^w(t) + \sum_{S_j\in O(S_i)} x_{ij}(t), \quad S_i \in \Delta \tag{4.6}$$

$$\widetilde{C_J}(t) = x_j(t), \quad S_j \in \Phi \tag{4.7}$$

Figure 4.4 is an example of the simulated counts in Allegheny County of Pennsylvania. I this model, I re-allocated the counts from 54 ZIP code areas with stores (in green) to the remaining 43 ZIP codes without stores.

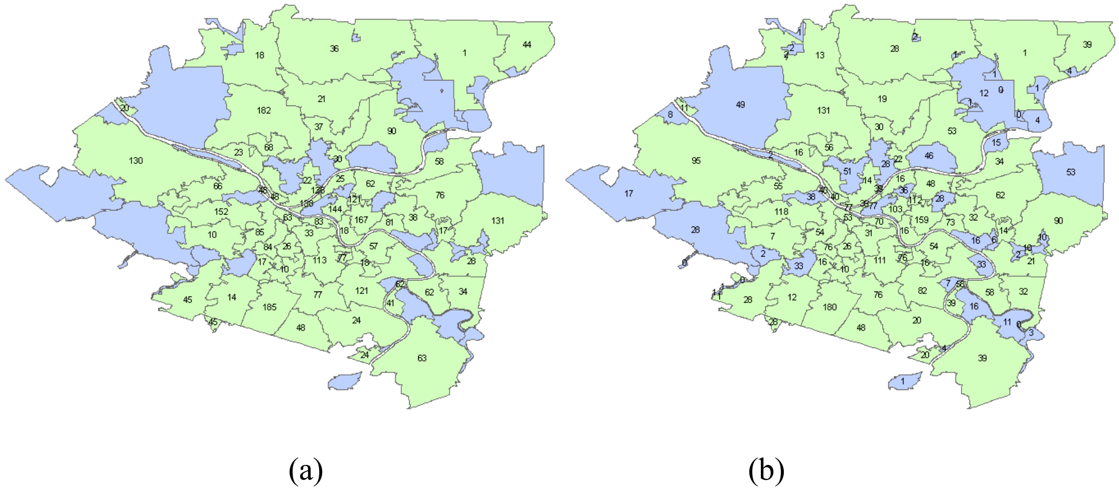


Figure 4.4 The simulated OTC counts in Allegheny County. (a) is the distribution of the counts which are available in the 54 ZIP code areas with stores and (b) is the simulated counts after applying the model.

The data set used in this research is the OTC sales of anti-cough/cold products. The simulation model provides a simulated OTC data set which can be used later in the multivariate analysis for early outbreak detection in addition to other data types which are available with regard to patient's residential ZIP code (for example, Ed visits, as mentioned above).

The data describing the commuting patterns was collected during the 2000 Census; it has national coverage and is provided at the census tract level: each commuting flow denotes the average daily number of commuters between a residence census tract and a work census tract. Since the surveillance data to analyze is available at a ZIP code level, we were required to adjust the commuting flows from a tract-to-tract level to a ZIP-to-ZIP level. Several approaches have been studied. The first, attempted by Buckeridge, was to convert the County-to-County commuting flows to the ZIP code level [59]. In [59] the flow conversion was realized by leveraging additional employment data at the ZIP code level. The second approach was to convert the Tract-to-Tract flows to a block group level .

In the following, I describe the approach we used to convert the Tract-to-Tract commuting flows to the ZIP code level. Note that ZIP code area can overlap with one or more census tracts, and vice versa. To perform the conversion we split each commuting flow between a pair of census tracts T_1, and T_2 into several smaller-sized flows according to the following to rules: (i) the number of workers coming from each constituent partial ZIP code area of T_1 was assumed to be proportional to the allocated area of the ZIP code within T_1; (ii) the number of workers going to T_2 was assumed to be divided among its constituent partial ZIP code areas, and they are proportional to the allocated area of those partial ZIP codes within T_2. The final commuting graph at the ZIP code level was then created by aggregating the portions of workflows between each ZIP code pair.

4.2 Linear outbreak simulator

Since the precise occurrence of outbreaks in historical public health surveillance data is often not well-defined, and historical surveillance data generally contain few well-documented outbreaks, outbreak simulation is often necessary to test detection algorithms [52][55][57][60][61]. Here I describe a simplified simulation model with linear increasing cases and it was used in the evaluation of univariate algorithms which I will discuss in the next chapter. The generated cases will then be injected into the background real data set to generate semi-synthetic experimental data. The injected signal is defined by a controlled feature set, including outbreak size (*K*), slope (δ) and duration (D). Eq. (4.8) defines the simulated outbreak counts; these counts will be added on the top of the background data in the pre-selected outbreak areas.

$$O(t,\delta,z) = \begin{cases} \delta \cdot t \cdot \mu_z , & 1 \le t \le \left\lfloor \frac{D}{2} \right\rfloor \\ \delta \cdot (D-t) \cdot \mu_z , & \left\lfloor \frac{D}{2} \right\rfloor + 1 \le t \le D \end{cases} \tag{4.8}$$

where t is the number of days after outbreak release and μ_z is the mean value of the daily counts in area z. Figure 4.5 shows a simulated outbreak curve, which has a simple triangular shape with an upward phase (Phase 1) simulating the spreading period. We define a true positive as an algorithm being able to find any of the outbreak areas within Phase 1. Although this simplified simulation is clearly a not very realistic outbreak, it does have several advantages: it allows us to precisely control the slope of the outbreak curve and examine how this affects our method's detection ability; in addition, the slowly elevated curve (slower than the log normal curves often observed in real outbreaks) extends the outbreak onset period to some extent, which allows us to distinguish the performance in terms of timeliness of each of the detection algorithms [10][62].

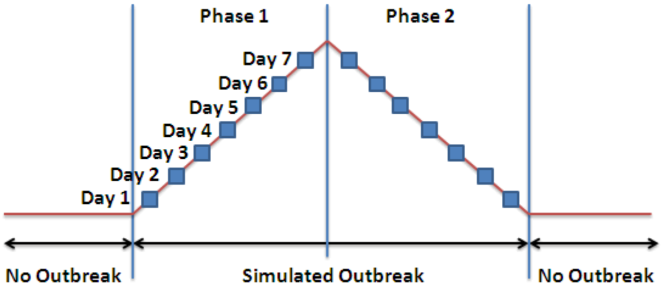


Figure 4.5 Illustration of the temporal shape of an artificial outbreak from Day 1 to Day *D* (*D*=14).

The geographical shapes of simulated outbreaks created in this study are designed to be flexible and independent of any detection algorithm, except that the simulator assumes an outbreak cluster has contiguous areas. First, I randomly selected a unit area as an outbreak area. Second, I randomly chose the rest of the outbreak areas; each one had to be adjacent to at least one of the previously selected areas. In this way, the outbreak simulator was mostly simplified and generalised. It conformed to the exhibiting characteristics of most outbreaks generated by many well-known simulation models which distribute infected cases into clustered (oftentimes connected) regions [63,64,65,66].

4.3 Multivariate spatial-temporal event simulator

I apply a multivariate spatial-temporal event simulator [55] to generate artificial outbreak cases and then superimpose them on the background data described in Chapter 4.1 to evaluate the multivariate algorithm I will propose in Chapter 5.

4.3.1 Parameters

Outbreak magnitude, denoted by O, is the total number of outbreak cases, including those not captured by the biosurveillance systems.

Behavior probability vector is a vector of length 2^M, consisting of the joint probabilities for the M behaviors for each outbreak case. For example, consider the simulation of a pneumonia outbreak with two data available streams: ED respiratory visit data and OTC anti-cough/cold medication sales. A joint probability vector (0.1, 0.35, 0.45, 0.1) means that for any outbreak case with probability 0.1, the case both went to an ED with a respiratory chief complaint and bought an anti-cough/cold OTC product; with probability 0.35 the case went to the emergency room but didn't buy an OTC product; with probability 0.45 the case didn't go to an ED but bought an OTC product; and with probability 0.1 the case neither went to an ED nor bought an OTC product.

Data coverage vector is a vector of length M, consisting of the coverage of each data stream. For example, RODS system is collecting ED data from 91% of the healthcare givers in Allegheny County, Pennsylvania. It means the ED data streams have 91% of coverage in this area. In other word, if an outbreak occurs in this area and we will expect that approximately 91% of outbreak cases can be captured by the system.

Spatial-temporal template is a function f of time and space that describes how the rate of new cases changes across time and space. Specifically, it is defined as $f(z,t) = f_Z(z)f_T(t|z)$, and it is a bounded joint probability mass function and probability density function over the spatial location and event times for each case. $f_Z(z)$ is the probability that a case is assigned to region z. This probability is a function of the elevated risk in region z. Specifically, for region z, let r_z denote the elevated spatial disease risk, and n_z denote the population. Then

$$f_Z(z) = \frac{n_z r_z}{\sum_{z' \in Z} n_{z'} r_{z'}} \tag{4.9}$$

One can define the spatial disease risk r_z as 1) flat, meaning each region s has the same risk, no matter how far it is to the center of the outbreak; 2) linear, meaning the region z is negatively proportional to the distance to itself and the center; and 3) other relationship.

One can define the temporal simulation function $f_T(t|z)$ given tract z as 1) flat, meaning each time unit has the same expected number of outbreak cases; 2) linear, meaning the mean value of the simulated cases at time t is linearly increased from the onset of an outbreak; and 3) other relationship.

One can also define $f_T(t|z) = f^*(t - l_z)$ when a time lag applies in one or multiple data streams, where l_z is the lag in the data stream in tract z.

4.3.2 The model

The outbreak simulation includes four steps:

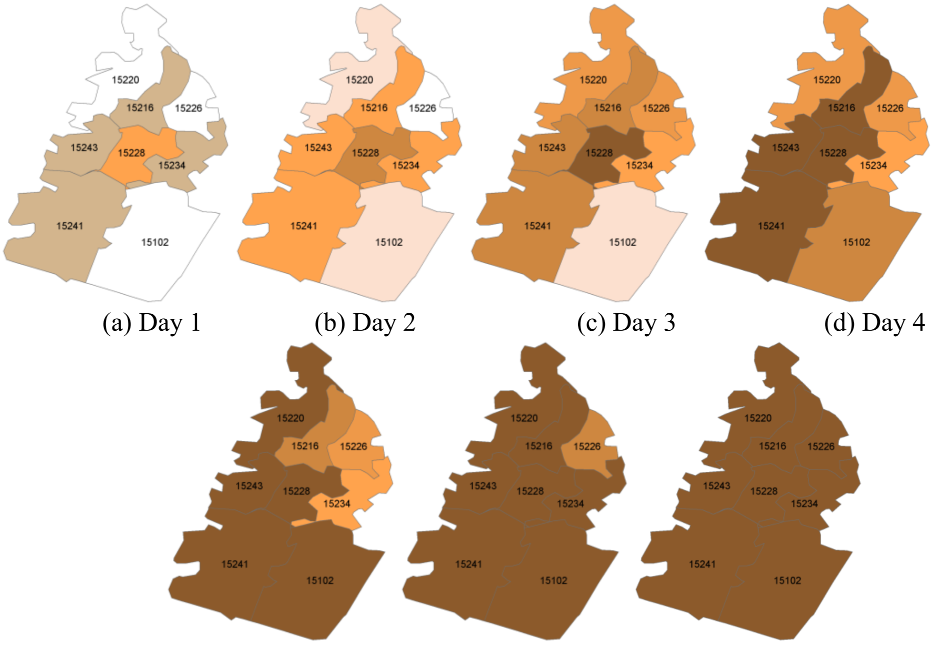
Step 1: Determine the total number of outbreak cases C and the release region z_0.

Step 2: Distribute the cases geographically into regions. In other words, randomly assign each case to a region according to the spatial template f_Z. For example, r_z can be defined as a function which is in inverse ratio to the distance between z and z_0, such that f_Z becomes a decreasing function of distance from the release region.

Step 3: Use the behavior probability vector, which may depend on the case's region, to determine the behaviors that the case will engage. Then, for each such behavior, use the associated coverage probability, which also may depend on the case's region, to determine whether that behavior is captured by the biosurveillance system.

Step 4: After determining the collection of captured behaviors for each case, simulate the vector of behavior event times by drawing an observation from the joint marginal distribution to event times for those behaviors in that case's region.

Figure 4.6 shows a geographical illustration of a simulated 7-day outbreak. The darker areas represent the areas with more outbreak cases. This outbreak is centered at the ZIP code area, 15228, and it spreads to the areas surrounding to the center when it develops along the duration (linear function, the counts in region s is negatively proportional to the distance to itself and the center). Figure 4.7 shows a temporal illustration of a 7-day outbreak using a linear temporal template where the numbers of the cases follow the Poisson process with linearly increased mean values.



(a) Day 1 (b) Day 2 (c) Day 3 (d) Day 4

(e) Day 5 (f) Day 6 (g) Day 7

Figure 4.6 A simulated 7-day outbreak showing the increased strength (darker colored) along when the outbreak develops. The outbreak is centered at 15228 and covers 8 ZIP code areas.

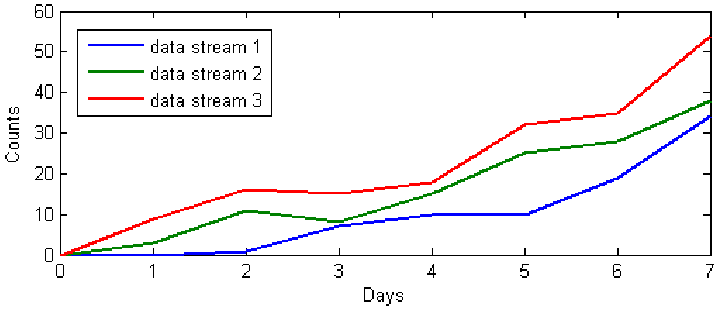


Figure 4.7 The simulated 7-day outbreak which infects 3 data streams. The y axis represents the summed counts over 8 ZIP code areas for each day within the outbreak duration (x axis

5.0 Rank-based tempo-spatial clustering (RSC)

This chapter describes the first part of the primary work I have done for the dissertation research. It includes a rank-based spatial clustering (RSC) framework I am proposing to meet the need for a biosurveillance system to identify disease outbreaks rapidly. In this chapter, I focus on the algorithms analyzing univariate data. A multivariate extension will be described in Chapter 6.0.

5.1 An algorithm for early outbreak detection—rank-based tempo-spatial clustering (RSC)

The input data for RSC include the observed values for each unit and the corresponding expected values. First, the study unit is chosen at the desired resolution level: a ZIP code area, a county, or even a rectangular grid cell. Observed data are then aggregated into each unit. To compute expected values, one can choose from the large variety of existing time series algorithms or statistical regression models.

The key steps in RSC include: 1) measuring the risk of each geographic unit having an ongoing outbreak; 2) ranking each unit by estimating its risk rate; 3) searching for clusters based on geographic adjacency given the order of the rankings; 4) computing the posterior probabilities of the clusters and identifying those with the highest.

5.1.1 Risk rate assessment

To assess risk rate, field epidemiologists normally prioritize all the areas of interest and then investigate the most abnormal ones first. Likewise, to prioritize, RSC first assesses the risk of a disease outbreak occurring in each unit area.

In the following, I propose two measures to estimate the risks. One is called standard score, which is computed as the number of standard deviations of the observed count varying from the expected count. This value is predicted from a time series of previous data [67]; the other is posterior probability using Bayesian inference.

Standard score (z-score). Generally, a risk R can be estimated as a ratio, $R(z,T) = q \equiv \frac{c_{z,T}}{b_{z,T}}$, where $c_{z,T}$ is the number of observed cases in area z on the most current day $t = T$, and $b_{z,T}$ usually denotes the population or the expected value computed from the baseline data for z_i on day T [30]. This ratio represents how far away the number of observed cases in area s is from the expected value. However, it does not clearly represent the normalized extent of the deviation.

The model computes a standard score (also known as z-score), $R(z,T) = SR(z,T) = \frac{c_{z,T}-b_{z,T}}{\sigma_z}$, to measure the risk for each area z, where σ_z represents the estimated standard deviation of the residuals. Residuals are computed by subtracting expected values from observed ones in the time series for each area z. Historical data are required to compute expected values using this method.

Bayesian posterior probability using a Gamma-Poisson model. Standard score may not apply when most of the counts are close to mean values because in such cases, it yields a close-to-zero standard deviation and results in an unreasonably large value for the z-score. With $R(z,T)$ estimated as the posterior probability, however, one can use a Bayesian approach, $P(H_1(z,T)|D)$, where T is the most current day [33]. I write it as $R(z,T) = BR(z,T)$= $P(H_1(z,T)|D)$. I assume that the counts for each area z within each period $t, 1 \leq t \leq T$ follow a Poisson distribution, which is commonly used to model a certain variable that counts a number of discrete occurrences during a time-interval of a given length [18]. Gamma distribution is used in Bayesian inference to model the prior variable q (the ratio between observed counts and expected counts) since it is the conjugate prior of a Poisson distribution. Expert knowledge can also be introduced by setting different prior probabilities, $P(H_1(z,T))$, to different unit areas s at different times t. For simplicity purposes, we applied uniform priors in this study. The posterior probability of H_1 (having an outbreak in area z_i) on day T is computed using Bayes theorem (Eq. (5.1)), where the likelihood of H_0 (not having an outbreak) and H_1 are integrals over the ratio q (Eq. (5.2) and (5.3)) with different shape parameters α and α', respectively. The marginal probability is computed as the sum over the two hypotheses as denoted in Equation (5.4).

$$P\left(H_1(z,T)\middle|D_{z,T}\right) = \frac{P\left(D_{z,T}\middle|H_1(z,T)\right)P(H_1(z,T))}{P(D_{z,T})} \tag{5.1}$$

$$P\left(D_{z,T}\middle|H_0(z,T)\right) = \int P\big(q{\sim}Gamma(\alpha_z,\beta_z)\big)P(c_{z,T}{\sim}Poisson(qb_{z,T}))dq \tag{5.2}$$

$$= \frac{b_{z,T}^{c_{z,T}}}{c_{z,T}!} \times \frac{\beta_z^{\alpha_z}\Gamma(\alpha_z + c_{z,T})}{\left(\beta_z + b_{z,T}\right)^{\alpha_z + c_{z,T}}\Gamma(\alpha_z)}$$

$$P\left(D_{z,T}\middle|H_1(z,T)\right) = \int P\big(q{\sim}Gamma(\alpha'_z,\beta_z)\big)P(c_{z,T}{\sim}Poisson(qb_{z,T}))dq \tag{5.3}$$

$$= \frac{b_{z,T}^{c_{z,T}}}{c_{z,T}!} \times \frac{\beta_z^{\alpha_z}\Gamma(\alpha'_z + c_{z,T})}{\left(\beta_z + b_{z,T}\right)^{\alpha'_z + c_{z,T}}\Gamma(\alpha'_z)}$$

$$P\big(D_{z,T}\big) = P\left(D_{z,T}\middle|H_0(z,T)\right)P\big(H_0(z,T)\big) + P\left(D_{z,T}\middle|H_1(z,T)\right)P\big(H_1(z,T)\big) \tag{5.4}$$

In the above equations, $c_{z,T}$ and $b_{z,T}$ are the observed and the expected values for area z on day T, respectively, and $\Gamma()$ represents the gamma function. The shape parameter (α_z) and the rate parameter (β_z) of the Gamma distribution are learned from the historical data by matching the first and second moments to sample mean and variance (Eq. (5.5-5.8)), assuming no outbreaks. As with the alternative hypothesis, I assume that the outbreak will increase q by a multiplicative factor m; thus I multiply α_z by χ while leaving β_z unchanged. Since we typically do not know the exact value of χ, here we use a discretized uniform distribution for χ, $\alpha'_i = \chi\alpha_i$ where $\chi = 1, \ldots, 3$ at intervals of 0.2.

$$\alpha_z = \frac{(E(q_z))^2}{Var(q_z)} \tag{5.5}$$

$$\beta_z = \frac{E(q_z)}{Var(q_z)} \tag{5.6}$$

$$\hat{\alpha}_z = \frac{(E_{sample}(q_z))^2}{Var_{sample}(q_z)} \tag{5.7}$$

$$\hat{\beta}_z = \frac{E_{sample}(q_z)}{Var_{sample}(q_z)} \tag{5.8}$$

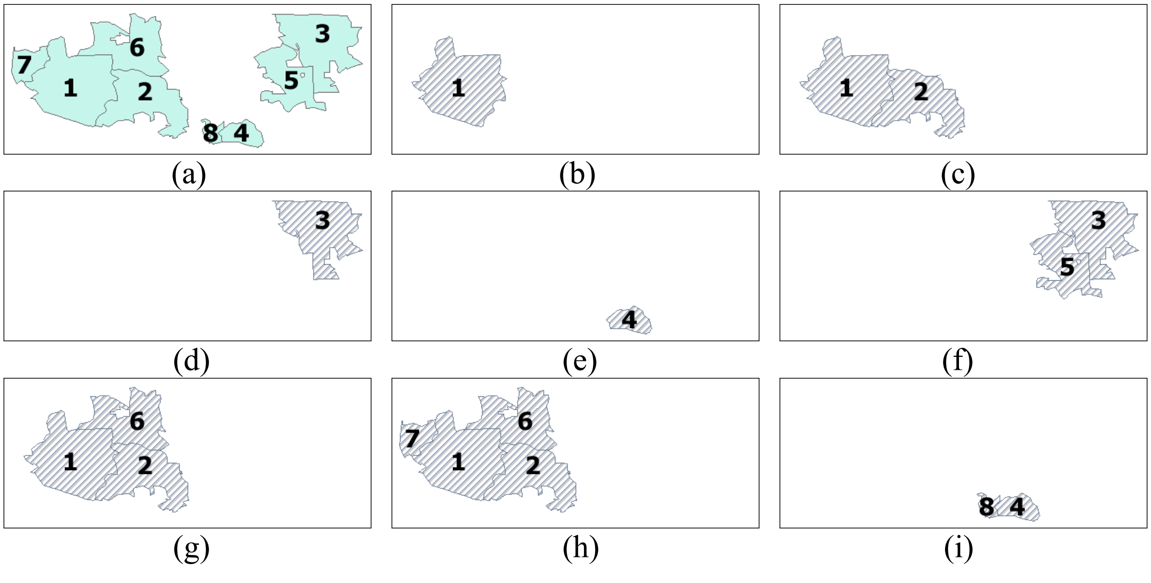
5.1.2 Adjacency criterion

RSC identifies potential clusters by determining the shortest Euclidean distance d_{ij} between every two areas z_i and z_j. In most cases, when a unit study area is a demographic area (e.g., a street group, a postal code or a county), the computation of the shortest distance needs to be done only once and the results can be stored for later use. I define an adjacency threshold, η. If $d_{ij} \leq \eta$, then areas z_i and z_j are considered to be adjacent. The areas are considered to fall into the same cluster when each is adjacent to at least one of the others in the cluster.

Computing the shortest distance between any two geographic areas is not trivial. However, most spatial database systems (e.g., postGIS) offer querying capabilities on topological relations. If regions of a plane are stored as vector polygons, the task starts by checking if the two polygons are intersecting; otherwise the shortest distance in between is computed. The first step takes $O(uv)$ operations, where each operation is to determine if two line segments intersect, with u and v representing the numbers of the vertices for the two polygons. If the two polygons do not intersect, the second step is then to compute their shortest distance, which can be completed in another $O(uv)$ operation [62]. For the purpose of simplicity, the adjacency threshold is set to 0 (*i.e.*, $\eta = 0$) in this study, and the task is reduced to intersection detection only.

5.1.3 Searching for clusters

RSC sorts all of the areas in descending order based on the risk rates estimated for all study areas. The search for emerging clusters is greedy — it starts from the highest ranked area, and this area becomes the first potential cluster itself. Then in the second iteration, the area with the second ranking is considered for clustering. If this area is adjacent to the first, they merge to form a bigger cluster. If not, they remain two separate clusters. Similarly, when the next area comes in, the algorithm checks if it is adjacent to one or more of the previously constructed clusters. If so, the algorithm unites the new area and its adjacent clusters into one cluster; otherwise it constructs a separate single-area cluster. Figure 5.1 illustrates all the clusters produced by the algorithm from a region where some areas are adjacent (connected in this example) and some are not.



(a) (b) (c) (d) (e) (f) (g) (h) (i)

Figure 5.1 Clusters created within a region of eight areas (Fig. 5.1(a)). Fig. 5.1(b)-(i) show the eight clusters in the order in which they were created. The number in each cell is the ranking of its risk rate, which represents the order in which it will be considered by the algorithm.

In order to constrain the growth of large clusters, we can determine an upper bound. For example, in this study, a cluster stops merging if it includes more than half of the study areas. The searching will cease after all the study areas have been analyzed. Each created cluster is scored using posterior probability, which will be described in the following chapter. A pseudo code is provided in Figure 5.2.

```
$CL = \{\}$
compute $R(z_i), i = 1, \ldots, n$
sort $(z_1, z_2, \cdots, z_n)$ into $(z_{r_1}, z_{r_2}, \cdots, z_{r_n})$ where $R\left(z_{r_j}\right) \geq R(z_{r_{j+1}}), j = 1, \cdots, n-1$
for $k = 1$ to $n$
    add $z_{r_k}$ into a new cluster $Z_{new}$
    for each cluster $Z$ in $CL$
        if $z_{r_k}$is adjacent to any area in $Z$
            add all the areas in $Z$ into $Z_{new}$
    if $|Z_{new}| < \frac{n}{2}$
        compute score $F(Z_{new})$ for cluster $Z_{new}$
        add $Z_{new}$ into CL
output $\max_S F(Z)$ in $CL$ if $F(Z)$ is greater than or equal to a predefined threshold
```

Figure 5.2 The pseudo code for cluster searching.

5.1.4 Priors

In practices, most Bayesian analyses are performed with so-called "noninformative" priors, that is, priors constructed by some formal rules. As in this dissertation, the prior probability of each cluster Z having an outbreak is assumed to follow a uniform distribution, $P\big(H_1(Z)\big) = \frac{P_1}{K}$, where P_1 represents the probability of there being an outbreak somewhere in the entire study region, and K is the total number of clusters created by the algorithm. However, subjectivism is the dominant philosophical foundation of Bayesian inference, so it is beneficial to construct an "informative" prior. One way to construct a meaningful prior for disease outbreak clustering is to adjust priors by using information about cluster sizes, populations, etc.. Since it is not a focus for this dissertation, this topic can certainly be explored in future work.

5.1.5 Computing posterior probability of a cluster

To compute the posterior probability of each cluster Z I use a Bayesian approach with the Gamma-Poisson model used in Bayesian spatial scan statistic (BSS) [10]. The hypothesis the model is testing is whether one cluster of areas has a disease outbreak occurring or whether there is no outbreak occurring in the entire study region. Eq. (5.9) and Eq. (5.10) compute the likelihood of not having an outbreak in the region (H_0) and the likelihood of having an outbreak in a cluster S (H_1), respectively.

$$P(D|H_0) = \frac{{\beta_{all}}^{\alpha_{all}}\Gamma(\alpha_{all} + C_{all})}{(\beta_{all} + B_{all})^{\alpha_{all}+C_{all}}\Gamma(\alpha_{all})} \times \prod_{all} \frac{{b_i}^{c_i}}{c_i!} \tag{5.9}$$

$$P(D|H_1(Z)) = \frac{{\beta_{in}}^{\alpha_{in}}\Gamma(\alpha_{in} + C_{in})}{(\beta_{in} + B_{in})^{\alpha_{in}+C_{in}}\Gamma(\alpha_{in})} \times \frac{{\beta_{out}}^{\alpha_{out}}\Gamma(\alpha_{out} + C_{out})}{(\beta_{out} + B_{out})^{\alpha_{out}+C_{out}}\Gamma(\alpha_{out})} \times \prod_{all} \frac{{b_i}^{c_i}}{c_i!} \tag{5.10}$$

In the null hypothesis, H_0, the number of observed cases summed over all the areas C_{all} follow a Poisson distribution, i.e., $C_{all}{\sim}Poisson(q_{all}B_{all})$, where B_{all} is the number of expected cases summed over all these areas, and the disease rate q_{all} follows a Gamma distribution, denoted as $q_{all}{\sim}Gamma(\alpha_{all}, \beta_{all})$. Similarly, in the alternative hypothesis $H_1(Z)$, the number of observed cases in the areas within cluster Z, C_{in}, follows a Poisson distribution $C_{in}{\sim}Poisson(q_{in}B_{in})$, where B_{in} is the number of expected cases summed over all the areas inside the cluster Z, and the disease rate has a Gamma distribution $q_{in}{\sim}Gamma(\alpha_{in}, \beta_{in})$. For the study areas outside cluster Z, C_{out}, B_{out}, α_{out} and β_{out} represent the corresponding parameters. The prior variables in

Gamma distributions can be estimated using the moment matching approach described in Eq. (5.5-5.8). The posterior probability can then be computed using the Bayes' Theorem in Eq. (5.11); the marginal probability is computed in Eq. (5.12).

$$P(H_1(Z)|D) = \frac{P(D|H_1(Z))P(H_1(Z))}{P(D)} \tag{5.11}$$

$$P(D) = P(D|H_0)P(H_0) + \sum_Z P(D|H_1(Z))P(H_1(Z)) \tag{5.12}$$

5.1.6 The temporal window

In addition to the spatial dimension, we can consider each searching region with different time durations $W(Z) = 1, \cdots, \omega$, for some constant integer ω. In other words, each cluster can be thought of as a prism with an n-sided polygonal base which covers the spatial areas of interest and a height which is the temporal duration. Since my study focuses on prospective analysis, which means that we are interested only in events that are current and recent, $W(Z) = 1$ represents the two dimensional cluster including the most current day's data while $W(Z) = \omega$ represents the cluster including the data from the most current day and the previous $\omega - 1$ days. In the following study, I use a simpler scenario setting $\omega = 1$, and thus only search over regions with 1-day duration; a larger value of the maximum temporal window size would be useful for identifying more slowly growing outbreaks. Choosing a different set of search regions would most likely affect the detection power of the methods, however, I expect that the relative performance of different methods will remain approximately the same.

5.1.7 Experiments

In this study, I compare the performance of RSC to KSS, the most widely known frequentist approach, and BSS, a Bayesian approach with competitive performance [10]. I also compare RSC to WAD, a well-established time series algorithm, to demonstrate the possible advantages of spatial algorithms [6][52]. I applied each algorithm to semi-synthetic data sets generated by superimposing outbreak cases into real over-the-counter (OTC) medicine sales data assumed to have no outbreaks.

Over-the-counter pharmaceutical sales data. As mentioned previously, some studies in the literature claim that the over-the-counter medication sales data can be one of the competent data sets for outbreak detection [56][57][58]. Thus I applied our algorithms to the data of OTC medicine sales in the cough/cold category in Pennsylvania between Jan. 1, 2006 and Dec. 31, 2008. This type of OTC medicine sales was chosen because it can indicate influenza activity.

I removed noisy data resulting from an imperfect data collection process. This was necessary as noisy data may bias an algorithm's detection power significantly since they do not correctly reflect the actual behaviors of medication purchases by patients. We defined an abnormal store reporting as a case when a store did not send any record in any of the 23 OTC categories for more than 27 days (allowing for an up to 5% data dropping rate plus 3 federal holidays each year). We then excluded those stores with abnormal reporting from this study. In the end, our dataset included 1,004 of the 1,502 stores located in the area being studied, and the data were aggregated into 471 ZIP code areas.

We chose the training period of between Jan. 1, 2007 and Dec. 31, 2007 to compute threshold values given fixed false alarm rates by assuming there were no known outbreak signals in this period. Each day within the training period was analyzed using an algorithm and the greatest score (i.e., the posterior probability) was recorded. A set of scores was then used as thresholds to control different false alarm rates. It is worth noting that the threshold values computed in this way are likely to be overestimated because of possibly existing but veiled outbreak signals. The overestimation problem can be corrected by excluding known outbreak periods from the training data set if historical outbreak information is available.

Semi-synthetic outbreaks. I used the linear outbreak simulation model introduced in Chapter 4.2 to generate artificial outbreak cases, which were then injected into the OTC data set to produce semi-synthetic experimental data. Recall that the injected signal is defined by a controlled feature set, including outbreak size (*K*, the number of study areas considered outbreak areas), slope (δ) and duration (D). I generated six groups of data sets, with K and δ chosen from {4, 8, 12} and {0.2, 0.3}, respectively. I arbitrarily chose 10 as the value for *D*. Each group included 100 outbreaks which were distinct from each other either geographically or temporally. More elevated outbreaks (*i.e.*, $\delta > 0.3$) are not discussed because of the indistinguishable performances among the different algorithms applied in the experiments.

During the evaluation period, from Jan. 1, 2008 to Dec. 31, 2008, I injected each simulated outbreak into a randomly selected interval of length *D* days during the evaluation period. I used a time series algorithm to compute each day’s expected value during the evaluation period, which was derived using the data from the 12 months prior to the study day.

Evaluation metrics. The evaluation metrics include 1) the receiver operating characteristic (ROC) curve [68], 2) the activity monitoring operating characteristic (AMOC) curve [69], 3) the areas under ROC and AMOC, 4) computation time, 5) cluster sensitivity (the proportion of the number of areas correctly detected) and 6) cluster positive predictive value (PPV, the proportion of the number of true outbreak areas in the detected cluster) [9][11][70].

Gold standard. For any spatial algorithm, we define a true positive output cluster as one that satisfies the three conditions: 1) having the highest score, which must be greater than a given threshold value, 2) having one or more outbreak area identified, and 3) having been identified within the upward phase of the outbreak (i.e., within the first 5 days).

For the purely temporal algorithm, WAD, a true positive output must be the unit area that 1) has the highest score, which must be greater than a given threshold value, and 2) is one of the outbreak areas.

Experimental results. I compared the performance of RSC to that of the wavelet anomaly detector (WAD) [7], the spatial scan statistic (KSS) [8][53][49] and the Bayesian spatial scan statistic (BSS) [10]. Note that the WAD was not only compared as an individual detection algorithm but also used to compute expected values required by the three spatial algorithms.

I applied the discrete Poisson model proposed in the KSS for a purely spatial analysis performed by *SaTScan* v8.0. Per the suggestion of the author of *SaTScan*, the size of a searched cluster was limited to be less than 3% of the population favoring relatively small and focused outbreaks. The parameter file *SaTScan* used for this study is in 7.2Appendix A. For BSS, we used a 24-by-24 grid structure to cover the entire study region (i.e., state of Pennsylvania), and the area of each cell was approximately 80 square miles.

The experiments were executed on a Linux server with a 2GHz Intel CPU and 4GB memory. All of the algorithms were implemented in Java 1.5 except for KSS.

The ROC curves of the RSC algorithms using the standard score model and the Bayesian model are shown in Figure 5.3, as well as the curves for WAD, BSS and KSS. Figure 5.4 provides the corresponding AMOC curves. Because the lowest false alarm rate KSS could achieve is 0.76[1], it is not shown in the figures, which show a false alarm rate ranging between 0 and 0.2 per day. I assume that any false alarm rates greater than 0.2 (1 false alarm per 5 days) have no practical advantage to public health practitioners.

[1] In order to compute false alarm rates, we applied the KSS (as well as other compared algorithms) on each day between 01/01/2007 and 12/31/2007 without injecting any outbreak cases. For each day, only the cluster with the lowest p-value was considered. The results of the KSS showed that 278 out of 365 clusters had the lowest p-value, 0.001, which means the lowest false alarm rate the algorithm was able to achieve was 278/365=0.76.

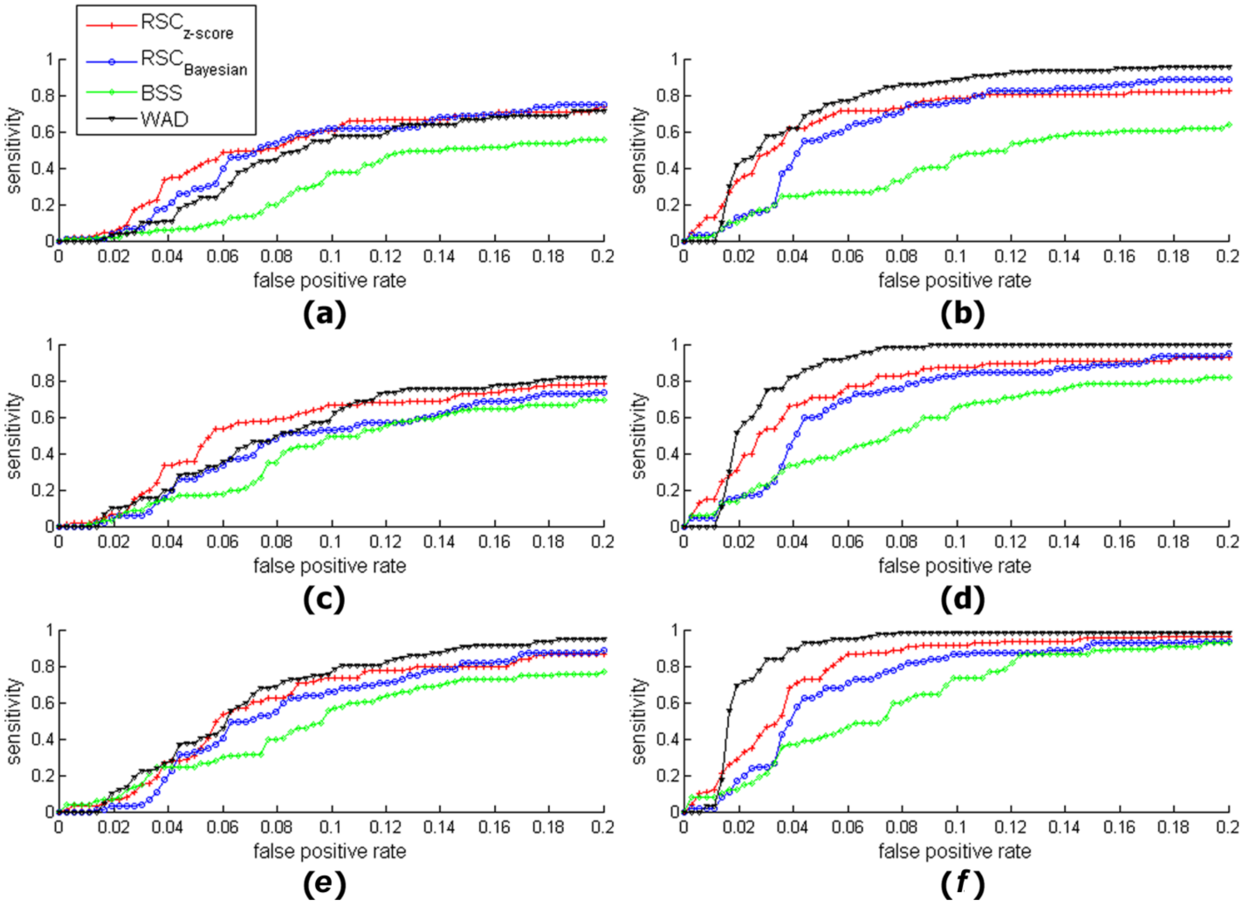


Figure 5.3 ROC curves of the four algorithms with different outbreak settings. (a) K=4, $\delta = 0.2$; (b) K=4, $\delta = 0.3$; (c) K=8, $\delta = 0.2$; (d) K=8, $\delta = 0.3$; (e) K=12, $\delta = 0.2$; (f) K=12, $\delta = 0.3$ where *K* represents outbreak size and δ represents outbreak intensity.

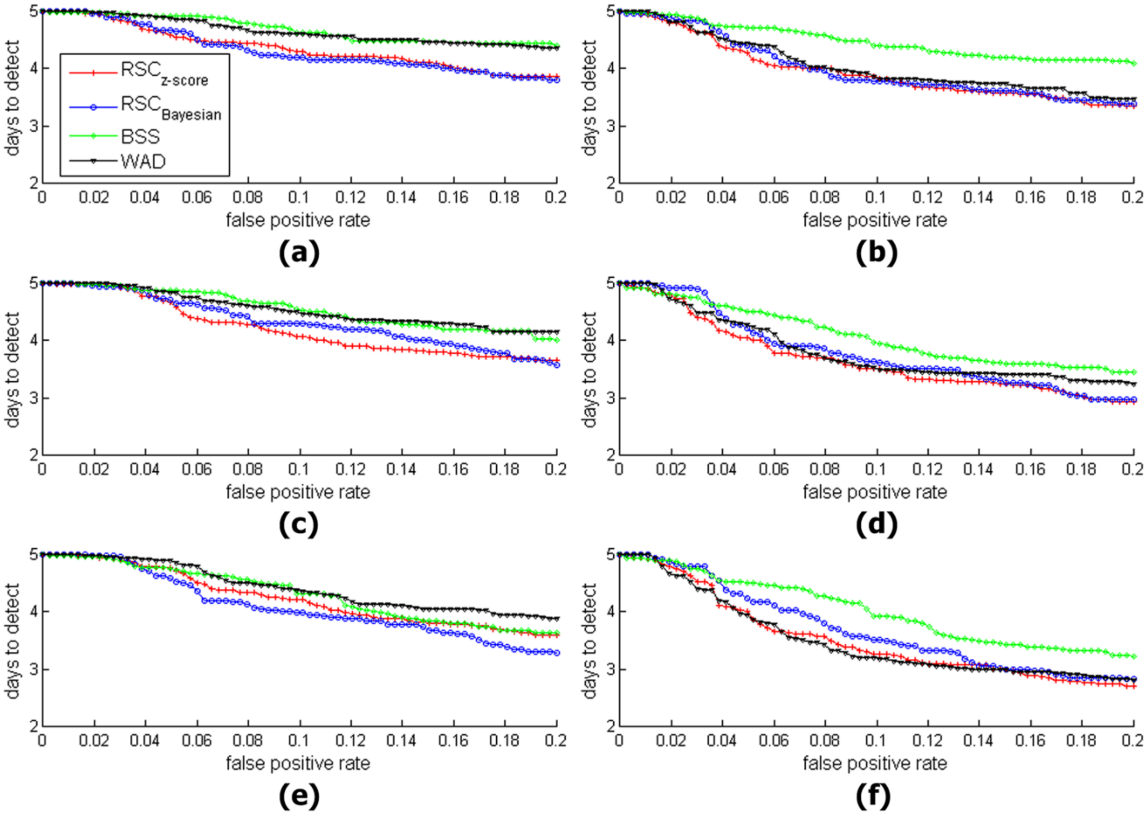


Figure 5.4 AMOC curves of the four algorithms with different outbreak settings. (a) K=4, $\delta = 0.2$; (b) K=4, $\delta = 0.3$; (c) K=8, $\delta = 0.2$; (d) K=8, $\delta = 0.3$; (e) K=12, $\delta = 0.2$; (f) K=12, $\delta = 0.3$, where K represents outbreak size and δ represents outbreak intensity.

Table 5.1 computes partial areas under ROC with a false positive rate within a range of [0, 0.2] using trapezoidal approximation. DeLong testing [71] showed no significant difference between any pair of the algorithms in this study regarding the areas under the curve, indicating that RSC, BSS and WAD showed similar detection powers.

Table 5.1 Algorithm comparison in terms of the areas under ROC between the range of false positive rates [0, 0.2]; the underscored numbers show the best performing methods and those in bold are not significantly different from the best.

K	δ	RSC$_{z\text{-score}}$	RSC$_{Bayesian}$	BSS	WAD
$K = 4$	$\delta = 0.2$	**0.1005**	**0.0957**	**0.0599**	**0.0875**
	$\delta = 0.3$	**0.1348**	**0.1280**	**0.0794**	**0.1525**
$K = 8$	$\delta = 0.2$	**0.1069**	**0.0889**	**0.0792**	**0.1028**
	$\delta = 0.3$	**0.1497**	**0.1364**	**0.1093**	**0.1706**

$K = 12$	$\delta = 0.2$	**0.1151**	**0.1082**	**0.0935**	**0.1256**
	$\delta = 0.3$	**0.1560**	**0.1402**	**0.1224**	**0.1740**

From the AMOC curves shown in Figure 5.4, the two RSC methods exhibited better timeliness than the other algorithms in 5 out of the 6 groups of experiments. The average days it took for the algorithms to detect the outbreaks are in Table 5.2. Paired student T-tests on the variable *days-to-detect* show that both the RSC methods were able to detect outbreaks significantly earlier than the BSS in all 6 groups of experiments, at a false alarm rate of 0.1 (i.e., 1 false alarm per 10 days). The RSC methods also outperformed the WAD when analyzing the data sets injected with low intensity outbreaks (i.e.,$\delta = 0.2$).

Table 5.2 Average days to detection at 1 false alarm per 10 days, for each of the 6 groups of simulations. The underscored results indicate the best performance and those in bold are not significantly different (at α=0.05) from the best.

K	δ	$\text{RSC}_{\text{z\_score}}$	$\text{RSC}_{\text{Bayesian}}$	BSS	WAD
$K = 4$	$\delta = 0.2$	**4.3**	**4.2**	4.63	4.62
	$\delta = 0.3$	**3.83**	**3.79**	4.4	**3.83**
$K = 8$	$\delta = 0.2$	**4.08**	**4.29**	4.53	4.47
	$\delta = 0.3$	**3.51**	**3.61**	3.95	**3.5**
$K = 12$	$\delta = 0.2$	**4.22**	**3.98**	4.32	4.37
	$\delta = 0.3$	**3.27**	3.52	3.92	**3.17**

The measures for average cluster sensitivities for the spatial algorithms at a false alarm rate of 0.1 are provided in Table 5.3. In all 6 groups of experiments, either $\text{RSC}_{\text{std\_score}}$ or $\text{RSC}_{\text{Bayesian}}$ had significantly higher values. This indicates that RSC is capable of identifying more outbreak areas than the other algorithms. In a similar fashion, Table 5.3 also shows the average measures for cluster PPVs (Positive Predictive Values).

Both RSC algorithms again performed the best, which suggests the RSC algorithms made fewer type II errors than the other algorithms did.

Table 5.3 Comparison of average cluster PPV's and cluster sensitivities at a false alarm rate of 0.1; the underscored results indicate the best performance and those in bold are not significantly different (at α=0.05) from the best.

Algorithm		RSC$_{\text{z-score}}$	RSC$_{\text{Bayesian}}$	BSS
Cluster PPV	$(K=4,\delta=0.2)$	**0.89**	**0.83**	0.54
	$(K=4,\delta=0.3)$	**0.89**	**0.85**	0.55
	$(K=8,\delta=0.2)$	**0.72**	**0.75**	**0.60**
	$(K=8,\delta=0.3)$	**0.84**	**0.84**	0.56
	$(K=12,\delta=0.2)$	**0.76**	**0.69**	0.49
	$(K=12,\delta=0.3)$	**0.71**	**0.70**	0.53
Cluster Sensitivity	$(K=4,\delta=0.2)$	**0.93**	**0.94**	0.81
	$(K=4,\delta=0.3)$	**0.95**	**0.95**	0.79
	$(K=8,\delta=0.2)$	**0.90**	**0.91**	0.76
	$(K=8,\delta=0.3)$	**0.91**	**0.89**	0.77
	$(K=12,\delta=0.2)$	**0.86**	**0.91**	**0.77**
	$(K=12,\delta=0.3)$	**0.86**	**0.86**	0.71

In the experiments, the average running times for the RSC$_{\text{z-score}}$, RSC$_{\text{Bayesian}}$, WAD, BSS and KSS are 26 seconds, 24 seconds, 0.22 seconds, 44 minutes and 2.58 seconds, respectively. WAD ran faster than any of the spatial algorithms. However, among the spatial algorithms, both the RSC$_{\text{std\_score}}$ and RSC$_{\text{Bayesian}}$ were 100+ times faster than the BSS. The running time for each algorithm can be explained theoretically looking at their computational complexity. The computational complexity of RSC is $O(n^2)$, where n is the number of total areas of interest. As I described in Chapter 3.5, BSS has an $O(m^4)$ computational complexity, where m is the length of the grid [10]. KSS has an $O(Rn^2)$ complexity for the purely spatial model, where R represents the number of replicated analyses required by randomization tests. $O(n)$ is the computational complexity of the pyramid algorithm implemented in the WAD [72]. Please note that the measured average running time of the KSS is not strictly comparable since the software *SaTScan* was implemented in a different programming language C.

5.1.8 Discussion

I have presented a rank-based spatial clustering algorithm and demonstrated several ways in which this approach is preferable to other temporal or spatial scan algorithms. I have also demonstrated that RSC using both risk estimation models consistently outperforms the other algorithms in terms of detection timeliness while having comparable detection powers. As a result, RSC revealed itself to be a more desirable algorithm for rapid and early detection of an outbreak.

One clear advantage to RSC is that it allows us to detect outbreaks in areas which are not connected. In some cases, the threshold distance can be adjusted so as to be inversely proportional to some density measure of a baseline factor (*e.g.,* population). In others, non-residential landforms (*e.g.,* lakes, valleys, etc.) may be attached to their nearest census tract and may be considered in the analysis as well. In fact, in the next chapter, I will introduce a study conducted using a grid cell as the unit study area and search for clusters with adjacent grid cells, where the geographical areas (e.g., ZIP code areas) inside two or more connected grid cells can be clustered together when they are not connected [73].

The accuracy and precision of its detection are also significant. Either an inaccurate or unreasonably larger-than-outbreak cluster may result in barriers for epidemiologists to conduct investigations given the very limited resources they typically have available. With the results showing that RSC is superior in finding compact clusters, it suggests that RSC is an ideal candidate among spatial algorithms for biosurveillance systems.

Another advantage of RSC is that it does not return a pre-determined cluster shape. The output of RSC is a combination of a set of adjacent areas (*e.g.*, ZIP code areas). A cluster identified by RSC may be any irregular and flexible shape, and the shape can be informative and indicative. For example, an elongated cluster may suggest an aerosol release of a disease agent that is following a downwind direction.

In addition to postal code areas as used in this study, other spatial units may be applied to spatial algorithms. For example, individual cases can be aggregated into street groups for a drill-down analysis, given that a more finely grained data set is available (*e.g.*, street-level address is available for analysis). However, when no predefined demographic unit is appropriate, such as individual cases which are indicative of an environmental hazard (*e.g.*, a radiation leak) or a nosocomial infection (*i.e.*, hospital-acquired infection), artificial units have to be considered, such as circles centering on a nuclear plant or a grid structure displaying ward locations in a hospital.

For the compared BSS algorithm, I used a 24-by-24 grid structure to cover the entire study region (i.e., state of Pennsylvania), with the area of each grid cell being approximately 80 square miles. Using a rather coarse grid structure may be one of the reasons why BSS performed less well. It would make a more compelling case to compare RSC to BSS with a finer grid size. However, to have a finer grid so that each cell can cover a single ZIP code area (assuming each ZIP code area covers about 10 square miles in average) we would need a finer than 64-by-64 grid structure. One of our pilot studies showed that a BSS with a grid size of 32 required 70+ minutes. Theoretically, considering the time complexity of BSS which is $O(m^4)$, a 64-by-64 grid size algorithm might take about 2^4 times 70 minutes which is about 19 hours.

One limit of the study is it employs a simplified outbreak curve. Despite the several advantages of the outbreak simulation model used in this study mentioned in the Methods Chapter, the outbreak curve was, nonetheless, artificially constructed and therefore may not represent the complexity of real outbreaks. It will be more challenging to use sophisticated and complicated outbreak simulators because they not only model the disease specific features but also consider the stochastic effects to some extent [64][65]. There are also multivariate simulators which consider the scenarios when signals of an outbreak are present in one or more data types [55]. The utilization of these simulators for algorithm evaluation, therefore, will become another substantial study as one of the extensions of this work.

5.2 Grid-based RSC (G-RSC)

While RSC uses the postal geographic unit (i.e., ZIP code), G-RSC superimposes an *m*-by-*m* grid window on the study region. Each ZIP code in the region is assigned to one of m^2 cells in the grid based on the longitude and latitude of its geographical centroid. The searching mechanism remains the same, but the unit of searching is a cell as opposed to a postal code. A benefit to this algorithm is it relaxes the limitation of the connectivity constraint at the scale of postal codes used in the previous study. It also overcomes to some extent the bias caused by false spikes produced by noisy data from the single isolated areas.

To evaluate the performance of G-RSC, I used the same evaluation metrics discussed above. I then compared its performance with that of the Bayesian spatial scan statistic (BSS). I applied the algorithms to semi-synthetic over-the-counter (OTC) medicine sales data sets. Each data set was generated by superimposing outbreak data onto the baseline data set (which has no known outbreaks to the best of our knowledge).

5.2.1 Experiments

The experimental baseline data set and the outbreak simulator I used in this study are the same as those used in the previous study described above.

Algorithm configuration. I compared G-RSC with the Bayesian spatial scan statistic (BSS) [10]. For both algorithms I employed WAD to compute the expected values by analyzing the 12-month historical data. I used wavelet transform at resolution level 7 to approximately extract the underlying seasonal trends (2^7=128 days) from a time series [7]. I superimposed a 24-by-24 grid onto the state of Pennsylvania with each cell covering approximately 80 square miles.

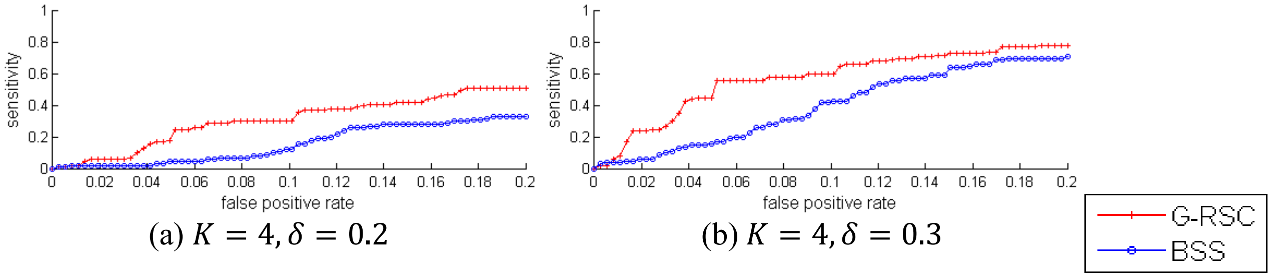
The experiments involved detecting 4 groups of independently generated outbreaks. Each group included 100 data sets, and each was injected with an outbreak having the same settings of outbreak size K and outbreak magnitude δ. The tuple, (K, δ), were chosen from the set, {(4, 0.2), (4, 0.3), (8, 0.2), (8, 0.3)}.

5.2.2 Experimental results

Figure 5.5 shows the ROC curves for the two algorithms G-RSC and BSS for each of the four groups of data sets. The areas under ROC were computed and are presented in Table 5.4. Figure 5.6 provides the corresponding AMOC curves for both algorithms.

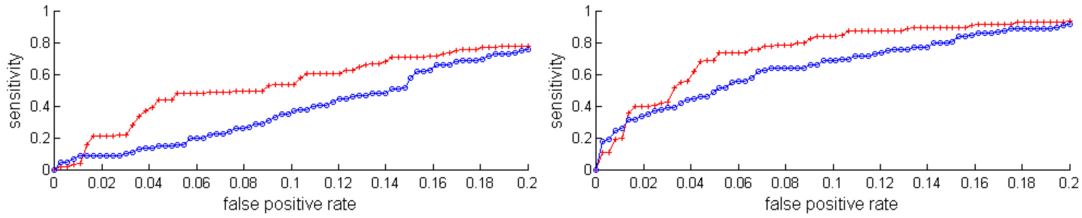
I applied the nonparametric approach proposed by DeLong et al [71] to compare the AUROCs of G-RSC and BSS statistically. The results show that in all four groups of experiments, G-RSC performed better than BSS (*p*-value=0.05), but not significantly. To compare their timeliness, I performed paired student T-tests on the variable *days-to-detect* to determine the difference between the two algorithms at a low false alarm rate, 0.1 per day (i.e., 3 false alarms are allowed per month). The results here show that G-RSC was able to detect outbreaks significantly earlier than BSS at a significance level of 0.05.

In addition to AUROC and days-to-detect, Table 5.4 also lists the average running time of the algorithms. G-RSC was shown to run more than 100 times faster than BSS. Moreover, Table 5.4 provides the average cluster PPV values and the cluster sensitivity values with 95% confidence intervals obtained at a false alarm rate of 0.1 (per day). In most of the experiments, G-RSC had a performance comparable to that of BSS.



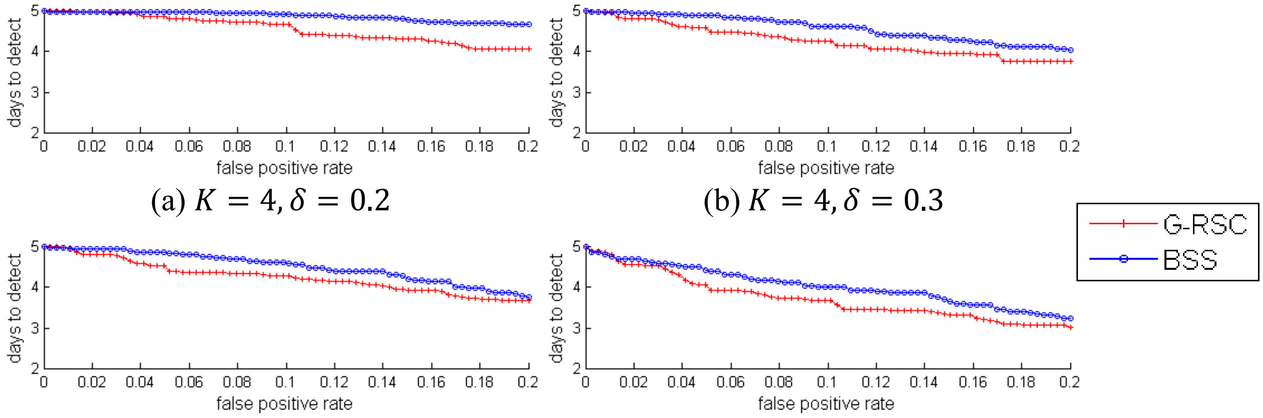
(a) $K = 4, \delta = 0.2$

(b) $K = 4, \delta = 0.3$



(c) $K = 8, \delta = 0.2$ (d) $K = 8, \delta = 0.3$

Figure 5.5 ROC curves of the four groups of experiments



(a) $K = 4, \delta = 0.2$ (b) $K = 4, \delta = 0.3$

(c) $K = 8, \delta = 0.2$ (d) $K = 8, \delta = 0.3$

Figure 5.6 AMOC curves of the four groups of experiments

Table 5.4 Comparison of AUROC, running time, cluster PPV and cluster sensitivity; the underscored results indicate the best performance and those in bold are not significantly different (at α=0.05) from the best.

Metrics		AU-ROC		Days to Detect		Avg. Run Time (secs)		Cluster PPV (95% CI)		Cluster Sensitivity (95% CI)	
Algorithms		G-RSC	BSS	G-RSC	BSS	G-RSC	BSS	G-RSC	BSS	G-RSC	BSS
$K = 4$	$\delta = 0.2$	**0.06**	**0.03**	**4.67**	4.92	**21.20**	2574.5	**0.45** **(0.34,0.58)**	**0.45** **(0.31,0.58)**	**0.83** **(0.74, 0.92)**	**0.89** **(0.81,0.98)**
	$\delta = 0.3$	**0.11**	**0.08**	**4.25**	4.62	**21.48**	2556.9	**0.45** **(0.36,0.53)**	**0.59** **(0.50,0.68)**	**0.83** **(0.77,0.89)**	**0.79** **(0.72,0.87)**
$K = 8$	$\delta = 0.2$	**0.10**	**0.07**	**4.28**	4.58	**21.48**	2540.7	**0.41** **(0.33,0.49)**	**0.51** **(0.42,0.60)**	**0.86** **(0.81,0.90)**	**0.83** **(0.77, 0.90)**
	$\delta = 0.3$	**0.15**	**0.13**	**3.67**	4.00	**21.43**	2545.9	0.41 (0.36, 0.47)	**0.54** **(0.49, 0.59)**	**0.89** **(0.86, 0.92)**	**0.89** **(0.86, 0.92)**

5.2.3 Discussion

With advances in information technology, the number of data types and data sources available for public health surveillance is consistently increasing. For example, our OTC data comprises 23 categories and ED chief complaint can be classified into multiple syndromes. In addition, data may be available at various scale levels such as county, state or even nation. Due to practical concerns, the computation time of an algorithm to finish analyzing all available data sets is important. As mentioned in the introduction chapter, most of the scan statistic algorithms suffer from excessive running times. KSS has a time complexity of $O((R+1)n^2)$ where n is the number of geographic locations in the study and R is the times of randomization required for significance testing (e.g. R is usually set to 999 (or greater) in most outbreak detection analysis). The time complexity of BSS is $O(m^4)$ when an m-by-m grid window is applied to the entire study region. When m is bigger, the computation time of BSS becomes intractable. However, G-RSC has a complexity of $O(m^2)$. The experimental results in this study showed that G-RSC ran 100+ faster than BSS with the same grid structure of size 24-by-24, which consequently would allow biosurveillance systems to analyze multiple data sets daily or even hourly.

By applying the grid structure in this study, G-RSC was able to outperform BSS in terms of both the detection power and timeliness. Since both G-RSC and BSS used grid cell as unit study area, the reason why G-RSC performed better can be explained as: when the outbreak shapes were irregular, BSS suffered from having innocent areas in the rectangular clusters which could possibly worsen its performance. It can be tested in one of the future work by comparing the performance of the two algorithms detecting rectangular shaped outbreaks.

There is no significant difference in cluster sensitivity and cluster PPV (except the experiment with K=8 and $\delta = 0.3$) between BSS and G-RSC, partially because both algorithms use a coarse resolution (a grid cell rather than a ZIP code area). However, the cluster PPV values of G-RSC were a bit lower than BSS. Since G-RSC is computationally efficient, a finer grid window may be applied.

Applying grid windows for clustering relaxes the connectivity constraint in the previous RSC. Two geographical locations can be classified into one cluster as long as they are adjacent enough (they are allocated into one cell or neighboring cells), but they do not have to be connected. This makes G-RSC capable of analyzing sparse data sets covering only small portions of a study region.

There were limitations with respect to the simulated outbreaks in this study. I only considered outbreak ZIP codes that were connected to each other, in accordance with the general belief that most infectious but non-contagious diseases are more likely to disperse to contiguous geographic locations. In addition, as in the study of the RSC algorithm described in Chapter 5.1, the same model used for producing outbreak cases was over-simplified; it may not represent all real outbreaks. Future work will employ a more sophisticated outbreak simulator, such as the Anthrax outbreak simulated by BARD or a multivariate outbreak simulator by Zhang et al [55,65].

6.0 The multivariate rank-based clustering (MRSC)

In disease surveillance, there is often more than one data source with which we wish to do surveillance. For example, the RODS system at the University of Pittsburgh [1,5,74,75,76] collects data from different sources, such as the patient registry data from emergency departments (ED), the over-the-counter (OTC) medication sales from pharmacies, and water quality data (including information about water distribution systems) from water companies. Using a multivariate approach to analyze these various types of data can have several advantages. First, if each type of data set is analyzed separately rather than collectively, the statistical power for detecting a disease outbreak inherent in all data sets may suffer, possibly due to the fact that individual data stream may be prone to noise alone. In addition to eliminating the impact of noisy data, another major reason for taking a multivariate approach to disease surveillance is that no single data source captures all the individuals affected by the outbreak. Depending on the disease, for example, some may go to their pharmacy and buy an over-the-counter medication for self-treatment; others may call their physician or a nurse hot-line, while others may visit their regular physician, or be sick enough to go to a hospital emergency room, or call an ambulance for urgent treatment [40]. At the same time, if the data sets are simply added together by taking the sum of the values in each data set, then a signal present in one data set with relatively low counts may be overwhelmed by the random noise in another data set which has relatively high counts.

As I mentioned in Chapter 3.0, new methods have been developed to improve the overall detection capabilities. However, most current algorithms are vulnerable to dramatic and unpredictable shifts in the health-care data that they monitor [48]. These shifts can occur during major public events, such as the Olympics or big conferences, as a result of population surges and public closures. Shifts can also occur during epidemics and pandemics as a result of quarantines, the worried well flooding emergency departments (e.g., during H1N1 flu pandemic [77]) or, conversely, the public staying away from hospitals for fear of nosocomial infection. Most surveillance systems are not robust to such shifts in health-care utilization because they do not adjust baselines and alert-thresholds to new utilization levels. As a result, it is necessary to consider the baseline shifts in order to deal with public-health crises and major public events that threat to undermine health-surveillance systems. In this dissertation, I want to consider this baseline shift phenomenon as one of the motivations and propose a new approach to tackle this problem.

As I have explained in Chapter 4.0, both ED and OTC data reflect, to some extent, the ways by which people seek treatment [56,57,58]. Because these treatment seeking behaviors are difficult to measure, the data streams from both data sources are equally weighted in this study. One disease outbreak can be thought of as a process that affects some subset of the data streams following some probabilistic characteristics. In this proposed model, I assume that an outbreak causes an increase in counts (for some subset of data streams). Specifically, if a flu-like outbreak is present, we may expect to see both an elevation of ED visits by patients with constitutional syndrome and an increase in OTC sales of anti-fever medications and thermometers.

Like the univariate rank-based clustering algorithm described in Chapter 5.1, the goal of the multivariate algorithm is outbreak detection and localization with fewer false alarms. That is, we want to decide if there is a disease outbreak given the data; if so, we want to determine the affected outbreak locations. More specifically, we are given a specific disease and M data streams, and each of them consists of time series data collected at a set of geographical locations z_i, for $i = 1, \dots, I$. I assume within the M data streams there are $K - 1$ data streams which are related to the modeled disease and $M - K$+1 data streams are unrelated. To model the given disease, I prepare a data set $\boldsymbol{D}$ as the following. Consider $\boldsymbol{D}$ as a matrix and D_j is column j. I use columns $D_j, j = 2, \dots, K$, to store the data streams which are related to the modeled disease, respectively; I use column D_1 to store the "other" unrelated data streams, which column consists of a time series of the summed data over all data streams subtracting the counts of the data streams $j, j = 2, \cdots, K$. From the perspective of statistical testing, this data stream can be considered a control group because the data share a common characteristic, that is, "non-relevant to the disease outbreak of interest". For example, assume we are given eight data streams (either from ED or OTC data sets) and we want to detect a flu-like outbreak that is believed to have effects on three of those data streams: 1) a time series of the counts of ED visits of patients with constitutional syndrome (CO); 2) a time series of sales of OTC anti-fever medications (AF); and 3) a time series of the counts of sales of thermometers (TH). We can model the disease in accordance with the following: D_2 consists of the data streams of CO; D_3 consists of the data stream of AF; D_4 consists of the data stream of TH;

and D_1 consists of a time series encompassing the remaining five data streams deemed irrelevant to flu-like diseases—for instance, the data streams consisting of OTC anti-diarrhea medication sales or ED visits of patients with rash syndrome.

I want to compare the set of alternative hypotheses $H_1(Z)$, each representing the occurrence of outbreak of interest in some region Z, against the null hypothesis H_0, which signifies that no such outbreak has occurred in any area in the entire study region. Like Equations (5.11) and (5.12), the Bayes' Theorem to compute the posterior probability of each $H_1(Z)$ then can be written as the following formula:

$$P(H_1(Z)|\boldsymbol{D}) = \frac{P\big(\boldsymbol{D}\big|H_1(Z)\big)P(H_1(Z))}{P(\boldsymbol{D}|H_0)P(H_0) + \sum_{Z\in Z} P\big(\boldsymbol{D}\big|H_1(Z)\big)P(H_1(Z))} \tag{6.1}$$

where $P(H_0) + \sum_{Z\in Z} P(H_1) = 1$.

When choosing the set of search regions $\mathcal{Z}$, we have different methods. For example, Kulldorff's spatial scan statistic [8][49] searches over circular or elliptic regions of continuously varying radii or axes, centered at each location s. Neill's Bayesian spatial scan statistic [10] uses an m-by-m grid to cover the entire study region and each region is composed of cells in a rectangle of various sizes and locations. In this study, I use the same heuristic methodology that I described in Chapter 5.1.3, where the unit study area is ZIP code area. I sort all of the areas in descending order based on the risk rates estimated for all study areas. The search for emerging clusters is greedy. It starts from the highest ranked area. This area itself becomes the first potential cluster. Then in the second iteration, the area with the second ranking is considered for clustering. If this area is adjacent to the first, both areas merge to a bigger cluster. If not, they remain two separate clusters. Similarly, when we consider the next area—that is the area which ranks highest among those areas not yet analyzed— the algorithm checks if it is adjacent to one or more of the previously constructed clusters. If so, the algorithm unites the new area and its adjacent clusters into one cluster; otherwise, it constructs a separate single-area cluster.

A risk rate of an area s_i given multiple data streams can now be defined as:

$$SR(z) = \frac{1}{K-1}\sum_{k=2}^{K} w_k SR_k(z) \tag{6.2}$$

where $SR_k(z)$ represents the standard score risk rate for area z estimated using data stream k which has been described in Chapter 5.1.1. w_k is the weighting factor that can be adjusted empirically to suit the user's different preferences on different data streams. Please note that the index variable k starts from 2, since the data stream 1 is the time series encompassing the remaining data streams that are unrelated to the modeled disease.

6.1 Computing likelihoods using the Multinomial-generalized-Dirichlet (MGD) model

One way to model both the disease-relevant data streams and also the control data streams is to use pair-wised Binomial-Beta distribution. More specifically, we can pair each disease-relevant data stream with each non-disease relevant data stream and model the change caused by a disease outbreak by using hierarchical Binomial-Beta distribution. If we do not have prior knowledge on how to pair the data streams, it is natural to find out all the combinations of pairing and perform the test on each of the combinations as in [48]. If we have m disease-relevant data streams and n control data streams (i.e., non-disease relevant data streams), the number of the pairs we need to consider would be mn. However, it will be convenient to have a model that considers all the data streams as a whole. Thus, this reason leads me to consider the hierarchical Multinomial-generalized-Dirichlet distribution and I will explain how in the following.

I assume that the counts in the given data streams have been generated from a hierarchical Multinomial-Generalized-Dirichlet (MGD) model. Since all these data streams are categorical and exclusive, it is natural to use multinomial distribution to model the counts as vector $\mathrm{X} = (\mathrm{X}_1, \mathrm{X}_2, \cdots, \mathrm{X}_\mathrm{K})$, where I use $\mathrm{X}_\mathrm{k}, \mathrm{k} = 2, \cdots, \mathrm{K}$ to represent the count in each data stream of interest and $\mathrm{X}_1 = \mathrm{N} - \mathrm{X}_2 - \cdots - \mathrm{X}_\mathrm{K}$ to represent the rest of the counts (N is the total count within a particular time interval over all the data streams under modeling).

In the literature of Bayesian analysis, Dirichlet distribution has been used as a prior for statistical models because it is the conjugate prior of both multinomial and categorical distributions. Spiegelhalter et al [78] used Dirichlet distrution to study the frequencies of congenital heart disease. Paulino and Pereira [79] developed a Bayesian approach to analyze incomplete categorical data that does not follow any specific pattern. Lange [80] assumed that allele frequency had a Dirichlet prior, and constructed a model to compute the forensic match probabilities. Dirichlet distribution, as a special case of generalized Dirichlet (GD) distribution, also has been widely used in geology, biology, and chemistry for handling compositional data which are subject to non-negativity and constant-sum constraints [27]. The statistical properties of such constrained random variables have been of interest in a variety of fields [81,82,83].

Generalized Dirichlet distribution has a more general covariance structure than Dirichlet distribution. This makes the generalized Dirichlet distribution to be more practical and useful [27]. Some contour graphs for Dirichlet distribution $D(\alpha, \alpha; \alpha)$,

$\alpha = 1,2,4,8,16,32$ are shown in Figure 6.1. The four contours in each graph are 0.05, 0.2, 0.5 and 0.9 contours comparing to the highest density value. No matter the value of α, the expectations of the three variables are all $\frac{1}{3}$. Figure 6.2 shows some contour graphs for generalized Dirichlet distribution for $GD(\alpha, 2\alpha; 2\alpha, 2\alpha)$, $\alpha = 1,2,4,8,16,32$. No matter the value of α, the expectations of the three variables are all $\frac{1}{3}$ too. However, by comparing Figure 6.1 and Figure 6.2, we can see that the contours of the Dirichlet distributions are symmetric, but the contours of the generalized Dirichlet distribution are not. This implies that an analyst whose prior is a generalized Dirichlet distribution can have different degree of beliefs on the random variables that have the same expected value.

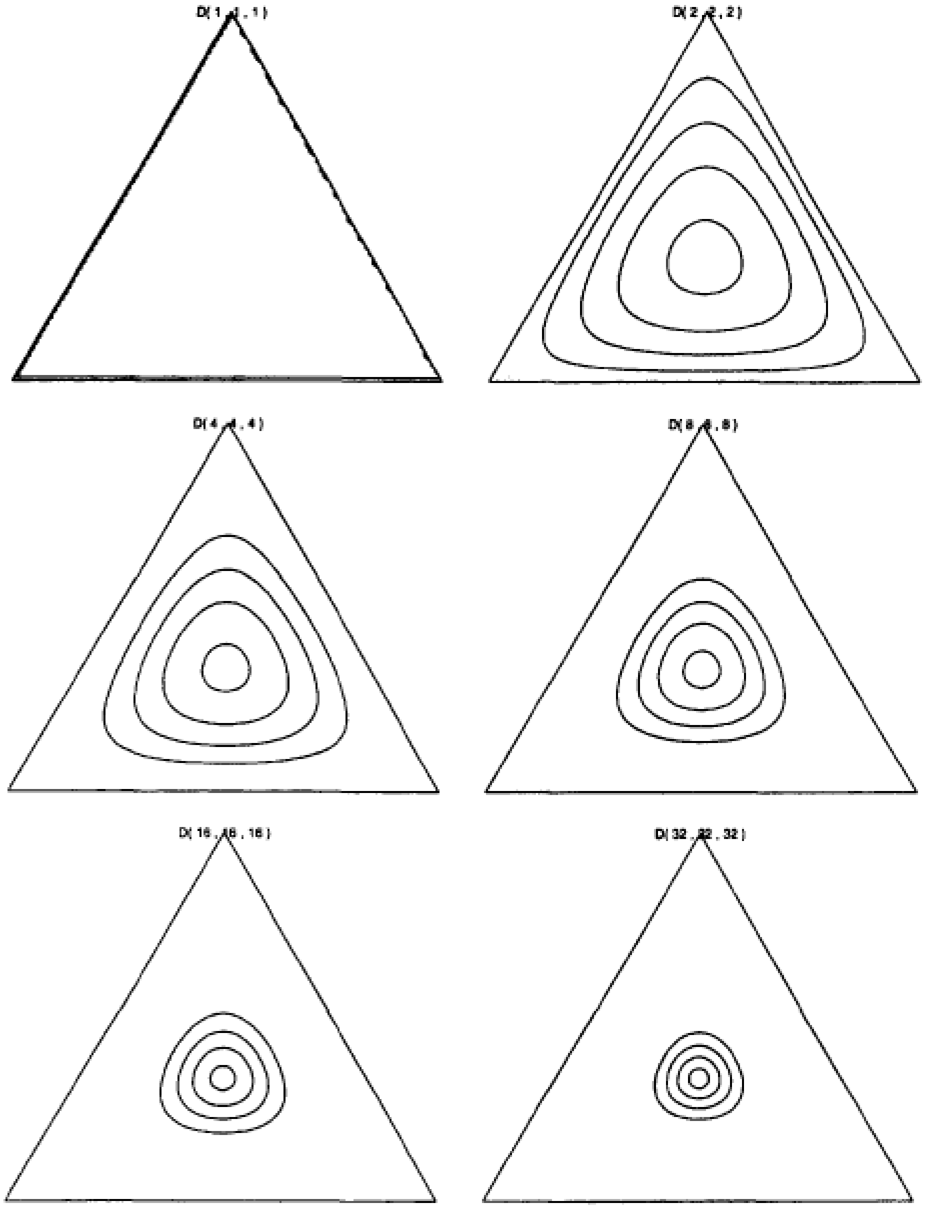


Figure 6.1 Contour graph for Dirichlet distribution. Adapted from “Generalized Dirichlet distribution in Bayesian analysis”, By Tzu-Tsung Wong, 1998.

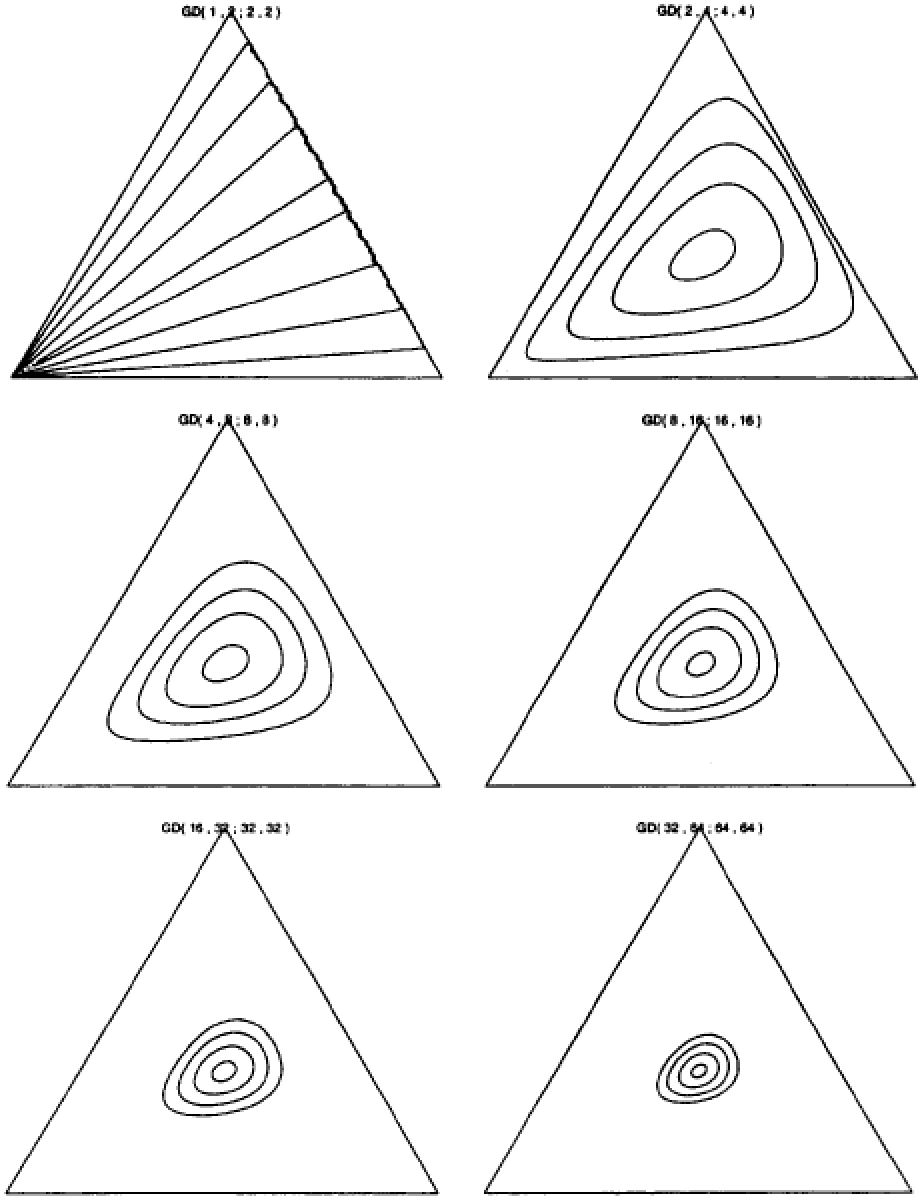


Figure 6.2 Contour graph for generalized Dirichlet distribution. Adapted from “Generalized Dirichlet distribution in Bayesian analysis”, By Tzu-Tsung Wong, 1998.

As I have described in Chapter 2.0, any two variables in $\mathrm{P} = (\mathrm{P}_1, \mathrm{P}_2, \cdots, \mathrm{P}_\mathrm{K})$ will be negatively correlated when P follows a Dirichlet distribution. However, in some domains, such as public health surveillance, two random variables may be positively correlated. For example, if P models the weights of the OTC medication sales in different categories (e.g., p_i models the weight of anti-fever medications sales, and p_j models the weight of cough/cold medications sales, i ≠ j), it is possible that $\mathrm{p_i}$ and $\mathrm{p_j}$ are both increased, i.e., positively correlated when there is an influenza outbreak occurring and the infected people are likely to have both symptoms. Hence I chose the GD distribution instead of the Dirichlet distribution based on the following two facts: 1) GD allows positive correlation among data streams, and 2) GD distribution is the conjugate prior of multinomial distribution used in our Bayesian model [27].

6.2 Bayesian inference

As described in the beginning of this chapter, to compute the posterior probability of each cluster Z using the Bayes' Theorem we need to compute the marginal likelihood $P\big(\boldsymbol{D}\big|H_1(Z)\big)$. Since our alternative hypothesis is that there is a cluster Z having an outbreak and there is no outbreak anywhere else, the marginal likelihood can then be written as $P\big(\boldsymbol{D}\big|H_1(Z)\big) = P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}})$, where $(\boldsymbol{\alpha_Z};\boldsymbol{\beta_Z}) = (\alpha_{Z,1},\cdots,\alpha_{Z,K-1};\beta_{Z,1},\cdots,\beta_{Z,K-1})$, $(\boldsymbol{\alpha_{G-Z}};\boldsymbol{\beta_{G-Z}}) = (\alpha_{G-Z,1},\cdots,\alpha_{G-Z,K-1};\beta_{G-Z,1},\cdots,\beta_{G-Z,K-1})$ and $\boldsymbol{\alpha}$'s and $\boldsymbol{\beta}$'s are the hyper-parameters of MGD distribution.

Since the observed counts $\boldsymbol{c_s} = (c_{s,1},\cdots,c_{s,K})$ of each area s in the cluster Z (i.e., $s \in Z$) follows a multinomial distribution with the parameter priors, we can then integrate over all possible latent variables $\boldsymbol{p_s}$ to compute the marginal. Since $P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})$ and $P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}})$ follow the same inference, I only derive $P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})$ in the following and write $\boldsymbol{\alpha_Z} = \boldsymbol{\alpha}$ and $\boldsymbol{\beta_Z} = \boldsymbol{\beta}$ to simplify notation.

$$P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z}) = \int_{\boldsymbol{p}} P\big(\boldsymbol{p}\sim GD(\boldsymbol{\alpha},\boldsymbol{\beta})\big)\prod_{z\in Z} P\big(c\sim MultiNom(\boldsymbol{p},n_z)\big)\,d\boldsymbol{p} \qquad (6.3)$$

where $n_z = \sum_{k=1}^{K} c_{z,k}$ is the total counts across all data streams for area z.

By explicitly writing out the formula for GD distribution and Multinomial distribution, we get:

$$\begin{aligned} &P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z}) \\ &= \varphi\int_{\boldsymbol{p}} p_K^{\beta_{K-1}-1}\prod_{k=1}^{K-1}\left[p_k^{\alpha_k-1}\left(\sum_{j=k}^{K} p_j\right)^{\beta_{k-1}-(\alpha_k+\beta_k)}\right]\prod_{z\in Z}\prod_{k=1}^{K} p_k^{c_{z,k}}\,d\boldsymbol{p} \\ &= \varphi\int_{\boldsymbol{p}} p_K^{\beta_{K-1}-1}\prod_{k=1}^{K-1}\left[p_k^{\alpha_k-1}\left(\sum_{j=k+1}^{K} p_j\right)^{\beta_{k-1}-(\alpha_k+\beta_k)}\right]\prod_{k=1}^{K} p_k^{C_{Z,k}}\,d\boldsymbol{p} \end{aligned} \qquad (6.4)$$

where $\varphi = \frac{1}{\prod_{k=1}^{K-1} B(\alpha_k,\beta_k)}\prod_{s\in Z}\left[\frac{\Gamma(\sum_{k=1}^{K} c_{z,k}+1)}{\prod_{\mathrm{k=1}}^{\mathrm{K}}\Gamma(\mathrm{c_{z,k}}+1)}\right]$ and $C_{Z,k} = \sum_{z\in Z} c_{z,k}$ for data stream D_k.

After reorganizing Equation (6.4), we get:

$$\begin{aligned} &P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z}) \\ &= \varphi\int_{\boldsymbol{p}} p_K^{\beta_{K-1}+C_{Z,K}-1}\prod_{k=1}^{K-1}\left[p_k^{\alpha_k+C_{Z,k}-1}\left(\sum_{j=k+1}^{K} p_j\right)^{\beta_{k-1}-(\alpha_k+\beta_k)}\right]d\boldsymbol{p} \end{aligned} \qquad (6.5)$$

Because

$\mathrm{GD}(\alpha_k + C_{Z,k},\beta_k + \sum_{j=k+1}^{K} C_{Z,j}) =$

$\frac{1}{\prod_{k=1}^{K-1} B(\alpha_k+C_{Z,k},\beta_k+\sum_{j=k+1}^{K} C_{Z,j})} p_K^{\beta_{K-1}+C_{Z,K}-1}\prod_{k=1}^{K-1}\left[p_k^{\alpha_k+C_{Z,k}-1}\left(\sum_{j=k+1}^{K} p_j\right)^{\beta_{k-1}-(\alpha_k+\beta_k)}\right]$ and

$\int_{\boldsymbol{p}} \mathrm{GD}(\alpha_k + C_{Z,k},\beta_k + \sum_{j=k+1}^{K} C_{Z,j})d\boldsymbol{p} = 1$ (the total area under a pdf is 1),

$P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})$ can then be computed in a closed form. Eventually, we can get:

$$P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z}) = \varphi \prod_{k=1}^{K-1} B(\alpha_k + C_{Z,k}, \beta_k + \sum\nolimits_{j=k+1}^{K} C_{Z,j})$$

$$= \prod_{z\in Z}\left[\frac{\Gamma\left(\sum_{k=1}^{K} c_{z,k} + 1\right)}{\prod_{k=1}^{K} \Gamma(c_{z,k} + 1)}\right] \prod_{k=1}^{K-1} \frac{B(\alpha_k + C_{Z,k}, \beta_k + \sum_{j=k+1}^{K} C_{Z,j})}{B(\alpha_k, \beta_k)} \tag{6.6}$$

Similarly, let $\boldsymbol{\alpha_{G-Z}} = \boldsymbol{\alpha'}$ and $\boldsymbol{\beta_{G-Z}} = \boldsymbol{\beta'}$**.** We can then write $P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}})$ as:

$$P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}}) \tag{6.7}$$

$$= \prod_{s\in G-Z}\left[\frac{\Gamma\left(\sum_{k=1}^{K} c_{s,k} + 1\right)}{\prod_{k=1}^{K} \Gamma(c_{s,k} + 1)}\right] \prod_{k=1}^{K-1} \frac{B({\alpha_k}' + C_{G-Z,k}, {\beta_k}' + \sum_{j=k+1}^{K} C_{G-Z,j})}{B({\alpha_k}', {\beta_k}')}$$

Again, the likelihood of the non-hypothesis $P(\boldsymbol{D}|H_0)$ is the following if we write $\boldsymbol{\alpha_G} = \boldsymbol{\alpha}''$ and $\boldsymbol{\beta_G} = \boldsymbol{\beta}''$ as:

$$P(\boldsymbol{D_G}|\boldsymbol{\alpha_G},\boldsymbol{\beta_G}) \tag{6.8}$$

$$= \prod_{z\in G}\left[\frac{\Gamma\left(\sum_{k=1}^{K} c_{z,k} + 1\right)}{\prod_{k=1}^{K} \Gamma(c_{z,k} + 1)}\right] \prod_{k=1}^{K-1} \frac{B({\alpha_k}'' + C_{G,k}, {\beta_k}'' + \sum_{j=k+1}^{K} C_{G,j})}{B({\alpha_k}'', {\beta_k}'')}$$

By plugging Equations (6.6), (6.7) and (6.8) into Equation (6.1) which applies the Bayes' Theorem, we can get the posterior probability of having an outbreak happening in cluster Z:

$$P(H_1(Z)|\boldsymbol{D}) = \frac{P\big(\boldsymbol{D}\big|H_1(Z)\big)P(H_1(Z))}{P(\boldsymbol{D}|H_0)P(H_0) + \sum_{Z\in Z} P\big(\boldsymbol{D}\big|H_1(Z)\big)P(H_1(Z))} \tag{6.9}$$

$$= \frac{P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}})P(H_1(Z))}{P(\boldsymbol{D_G}|\boldsymbol{\alpha_G},\boldsymbol{\beta_G})P(H_0) + \sum_{Z\in Z} P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z},\boldsymbol{\beta_Z})P(\boldsymbol{D_{G-Z}}|\boldsymbol{\alpha_{G-Z}},\boldsymbol{\beta_{G-Z}})P(H_1(Z))}$$

6.3 Estimating hyper-parameters

I use the method of moment matching to estimate the prior parameters from historical data. The first and second moments are readily obtained from the results given in [28].

$$E(p_k) = \frac{\alpha_k}{\alpha_k + \beta_k} \prod_{m=1}^{k-1} \frac{\beta_m}{\alpha_m + \beta_m}, \quad k = 1, \ldots, K-1 \tag{6.10}$$

$$Var(p_k) = E(p_k)\left[\frac{\alpha_k + 1}{\alpha_k + \beta_k + 1} \prod_{m=1}^{k-1} \frac{\beta_m + 1}{\alpha_m + \beta_m + 1} - E(p_k)\right], \quad k = 1, \ldots, K-1 \tag{6.11}$$

We can solve this for each k recursively from $\mathrm{k} = 1$ to $\mathrm{K} - 1$ and let $\varphi_\mathrm{k} = \prod_{\mathrm{m}=1}^{\mathrm{k}-1} \frac{\beta_\mathrm{m}+1}{\alpha_\mathrm{m}+\beta_\mathrm{m}+1}$ for $\mathrm{k} = 2, \cdots, \mathrm{K} - 1$ and $\varphi_1 = 1$. Eventually, we get

$$\alpha_k = \frac{\mu_k(\varphi_k - \rho_k)}{\rho_k\left(1 - \sum_{j=1}^{k-1} \mu_j\right) - \varphi_k \mu_k} \tag{6.12}$$

$$\beta_k = \frac{(1 - \sum_{j=1}^{k} \mu_j)(\varphi_k - \rho_k)}{\rho_k\left(1 - \sum_{j=1}^{k-1} \mu_j\right) - \varphi_k \mu_k} \tag{6.13}$$

where $\mu_k = E[p_k], \rho_k = \frac{E[p_k^2]}{E[p_k]}$, for $k = 1, \cdots, K - 1$.

Considering the historical data as a random sample $(X_{1j}, X_{2j}, \cdots, X_{Kj}), j = 1, \cdots, N$, we can estimate $\mu_k = \hat{\mu}_k = \frac{1}{N}\sum_{j=1}^{N} \frac{X_{kj}}{\sum_{k=1}^{K} X_{kj}}$ and

$\rho_k = \hat{\rho}_k = \frac{1}{\hat{\mu}_k} \cdot \frac{1}{N} \sum_{j=1}^{N} \left(\frac{X_{kj}}{\sum_{k=1}^{K} X_{kj}}\right)^2$ for $k = 1, \cdots, K - 1$.

6.4 Modeling outbreak effects

In order to model the alternative hypothesis, $H_1(Z)$, we need to consider the different effects of a disease outbreak on different data streams. In my proposed model, the effects of a disease outbreak on the data are determined by values π_k defined for the k data stream, respectively. In other words, given an alternative hypothesis, $H_1(Z)$, with Z being the region affected by the disease outbreak of interest, I use an MGD distribution with parameters $(\boldsymbol{\pi I \alpha_Z}; \boldsymbol{\beta_Z})$ where $\boldsymbol{I}$ is the identity matrix of size $K-1$ to model the data in Z. The vector of effects $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_{K-1})$ changes the expected values of the parameters in the multinomial distribution from $E(p_k) = \frac{\alpha_k}{\alpha_k+\beta_k}\prod_{m-1}^{k-1}\frac{\beta_m}{\alpha_m+\beta_m}$ to $E(p_k) = \frac{\pi_k\alpha_k}{\pi_k\alpha_k+\beta_k}\prod_{m=1}^{k-1}\frac{\beta_m}{\pi_m\alpha_m+\beta_m}$ by multiplying α_k by π_k. In the following, I consider two scenarios of outbreaks: 1) the outbreaks have same effects on all the disease-relevant data streams; and 2) the outbreaks have different effects on different disease-relevant data streams.

6.4.1 Modeling same outbreak effects on different disease-relevant data streams

When there is no prior knowledge which can be assessed about the characteristics of a disease outbreak, we usually assume the outbreak will have the same effects on the data streams. Consider the effect vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_{K-1})$ where I set $\pi_1 < 1$ and $\pi_k = 1$ for $k = 2, \cdots, K-1$, the expected value of p_1 will then be reduced from $\frac{\alpha_1}{\alpha_1+\beta_1}$ to $\frac{\pi_1\alpha_1}{\pi_1\alpha_1+\beta_1}$ but all the expected values of $p_2, \cdots, p_K$ will be increased by a factor of

$\Delta_k = \frac{\beta_1}{\pi_1 \alpha_1 + \beta_1} / \frac{\beta_1}{\alpha_1 + \beta_1} > 1$, for $k \geq 2$ (see Equation (6.10)). Recall in a matrix of data $\boldsymbol{D}$, I put all the data streams which are believed to be affected by the disease of interest in the column D_2 to column D_K and the rest of the unrelated data are summed into one column D_1. In this way, we are able to model the alternative hypothesis, $H_1(Z)$, since all $E(p_k)$, $k \geq 2$, of the data in region Z are increased by the factor of Δ_k.

The distribution of π_1 is not only dependent on the specific disease outbreak we are interested in but also on the intensity of the outbreak signals. Here I assume a simplified model, in which the effect variable π_1 follows a discrete uniform distribution between $[\pi_{min}, 1)$ with m possible values. For example, if a given outbreak has an effect variable π_1 which follows the discrete uniform distribution in the range $[0.1,1)$ with 9 values in between, then π_1 is equally likely to be the values in the set $\omega = \{$0.1, 0.2, 0.3, 0.4. 0.5, 0.6, 0.7, 0.8, 0.9$\}$. The marginal probability of the alternative hypothesis likelihood is then computed as:

$$P(\boldsymbol{D_Z}|H_1(Z)) = \sum_{\boldsymbol{\pi}} P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z}, \boldsymbol{\beta_Z}, \boldsymbol{\pi}) P(\boldsymbol{\pi}) \tag{6.14}$$

$$= \frac{1}{|\omega|} \sum_{\pi_1 \in \omega, \pi_k = 1, k=2,\ldots,K} P(D_Z|\tilde{\alpha}_Z, \beta_Z)$$

where $\tilde{\alpha}_{Z,k} = \pi_k \alpha_{Z,k}$.

In Chapter 6.5.2 and 6.5.3, I simulate two sets of outbreaks using the multivariate spatial temporal event simulator described in Chapter 4.3. I assume same data coverage on different data streams; thus the strength of the injected outbreaks are same for the all the disease-relevant data streams. I apply both the Multinomial-Dirichlet (MD) model and the Multinomial-generalized Dirichlet (MGD) model in an MRSC algorithm to detect the simulated outbreaks and measure the performance of detection. In Chapter 6.5.4, I compare the MRSC using an MGD model to two other multivariate algorithms, MKSS and MBSS.

6.4.2 Modeling different outbreak effects on different disease-relevant data streams

In Chapter 6.4.1, I introduced the way I model outbreaks having the same effects on different data streams. Nonetheless, it is necessary to study the scenario when different data streams contain outbreak signals with different magnitudes as well. One reason is that some diseases may affect different data streams to different extents. For example, among all the symptoms of SARS, cough and fever are most common whereas diarrhea and nausea are less common. Therefore, if a SARS outbreak occurs, the data streams recording anti-fever and anti-coughing medication sales may have stronger signals than the data stream of anti-diarrhea medication sales. Another reason to study this scenario is that coverage of the data sources may differ depending on the surveillance system used to collect data. For instance, the RODS system has different market-share coverage between OTC stores and emergency departments in hospitals.

In order to better model data streams with different signal strength, we need to find a set of π_k $(k = 1, \dots, K)$ values such that the MGD model with parameters $\boldsymbol{\pi I_K \alpha_Z}$ and $\boldsymbol{\beta_Z}$ can fit the data under the alternative hypothesis, $H_1(Z)$, with the region Z being the affected areas by the disease outbreak of interest.

Given a disease outbreak, I assume the proportionality constants of the elevations among all the disease-relevant data streams can be assessed by domain experts or learned from training data. For example, we may have the knowledge that the increase percentage of data stream a caused by a certain type of disease outbreak will be twice of that of data stream b. Mathematically, let e_k represent the expected values of multinomial distribution parameters ($E(p_k)$) before an outbreak occurs, and $\tilde{e}_k$ represent the expected values ($\tilde{E}(p_k)$) after the outbreak occurs, and let $\Delta_k = \frac{\tilde{e}_k - e_k}{e_k}$ $(k \geq 2)$ represent the increase in percentage for data stream k. In addition, as in Chapter 6.4.1, I define $\pi_1 < 1$ and as equally likely to be the values in the set $\omega =$\{0.1, 0.2, 0.3, 0.4. 0.5, 0.6, 0.7, 0.8, 0.9\}. Given the proportionality constants among Δ_k's $(k \geq 2)$ and the value of π_1, we can then solve for π_k $(k \geq 2)$. I describe the formula in the following paragraph.

Recall that $e_k = \frac{\alpha_k}{\alpha_k + \beta_k} \prod_{m=1}^{k-1} \frac{\beta_m}{\alpha_m + \beta_m}$ is recursively dependent on $e_1, \ldots, e_{k-1}$. Let $\tilde{e}_k = \frac{\pi_k \alpha_k}{\pi_k \alpha_k + \beta_k} \prod_{m=1}^{k-1} \frac{\beta_m}{\pi_k \alpha_m + \beta_m}$, and $\tilde{\vartheta}_k = \frac{\pi_k \alpha_k}{\pi_k \alpha_k + \beta_k}$. Given the constraint of the multinomial distribution where $\sum_{k=1}^{K} p_k = 1$, the total change across all data streams should be 0, that is $\sum_{k=1}^{K} \Delta_k e_k = 0$. Eventually, π_k $(k \geq 2)$ can be computed recursively as the following given e_k, $\tilde{\vartheta}_1$, and the proportionality constants a_k of Δ_k's (see 7.2Appendix D).

$$\pi_k = \frac{\beta_k}{\alpha_k} \frac{\tilde{\vartheta}_k}{1 - \tilde{\vartheta}_k} \tag{6.15}$$

where $\tilde{\vartheta}_k = \frac{\tilde{e}_k}{\prod_{m=1}^{k-1}(1 - \tilde{\vartheta}_m)}$ and $\tilde{e}_k = (1 - \frac{a_k \Delta_1 e_1}{\sum_{m=2}^{K} a_m e_m}) e_k$ for $k \geq 2$.

In Chapter 6.5.5 (Experiment IV – varied outbreak effects on different data streams), I simulate outbreaks by using the multivariate spatial temporal event simulator described in Chapter 4.3. I define different data coverage on different data streams; thus the strength of the injected outbreaks are different for the data streams. As I mentioned, the proportionality constants among Δ_k's can be assessed either by domain experts or learned from previous known outbreaks using machine learning approaches. I also describe how I learned the values of Δ_k's from the simulated outbreaks in Chapter 6.5.5 by using maximum likelihood.

6.4.3 Detecting outbreaks from multiple data streams with a time lag

Since multiple data sources are available for disease surveillance, people have been investigating the temporal correlations between different data types, such as between the OTC thermometer sales and ED visits for Influenza like illness [84]. Although temporal correlations between OTC and ED data during an outbreak are hard to estimate from data because few existing training data capture the effects of a large-scale epidemic on these data sources over the same period, it seems logical to believe that time lags exist. For example, the supposition exists that most ED event times occur later than OTC event times if cases tend to self-treat with OTC products earlier in the course of an illness and visit an ED later in the course of the same illness. In other words, over-the-counter (OTC) medications are commonly taken before or instead of seeking medical care [85,86,87]. The data stream of medication sales from OTC stores therefore might contain earlier cases of communitywide illness than the data streams of ED patient visits. In Chapter 6.5.6, I apply the MGD model to analyze data streams which have been injected with time-lagged outbreaks cases. I assume that the time lag on different data streams with respect to a certain disease is already known or can be learned from literature. The hypothesis of the study is that the MGD model can be helpful to detect outbreaks with time lags by assuming early data streams have larger elevation than the later data streams. I test this hypothesis in Experiment V (Chapter 6.5.6).

6.5 Evaluation

6.5.1 Experimental data sets

Emergency departments and drug stores are two main resources from which the RODS (Real-Time Outbreak and Disease Surveillance) system collects data [1,74,76]. The data include patient ED registration information from more than 100 health providers in Pennsylvania and also cover OTC medication purchases from more than 30,000 drug stores of 13 companies in the U.S. In all my experiments, I used seven data streams from either ED or OTC data sources in Allegheny County, Pennsylvania, which include daily counts (sales or patient visits) of anti-fever OTC, diarrhea remedies OTC, thermometer OTC, hydrocortisones OTC, gastrointestinal ED, constitutional ED and rash ED. Additional data streams, such as stomach remedies OTC, neurological ED, etc., are also available, but these disparate data streams are less related to the infectious diseases we are interested in and so they were not considered in the analysis in this study. I will further discuss this in the discussion (Chapter 6.6).

The temporal coverage of the data set is the 24 months between Jan. 1, 2007 and Dec. 31, 2008. The geographic region in this study is Allegheny County, Pennsylvania. As in my previous study, I removed partial data which resulted from an imperfect data collection process (described in the following); they would bias an algorithm's detection power significantly since they do not correctly reflect the actual behaviors of medication purchases by patients. I defined an abnormal reporting as a case when a store or an emergency department did not send any record in any of the 23 OTC categories or any of the 9 ED syndrome categories for more than 27 days (allowing for a 5% data dropping rate plus 3 federal holidays each year). I excluded the drug stores and emergency departments with abnormal reporting. In the end, my data set included data aggregated into 54 Zip code areas in Allegheny County which had been reporting data for all of the seven streams during the entire study period.

6.5.2 Experiment I – detecting flu-like disease outbreaks

In this set of experiments, we model flu-like disease outbreaks. Among the seven data streams, we believe a flu-like disease outbreak will have effects on three: anti-fever OTC (AF), thermometer OTC (TH) and constitutional ED (CO). The other four data streams are believed to remain unaffected and their daily counts are summed together to form the data stream which I call "Others". Recall both the MD and MGD models have a requirement for the order of the input data types. They model the elevations in the data streams in column $k, k \geq 2$ and the decreases in the data stream in the first column. Thus, the input data streams for the MRSC algorithm have the order of Others, AF, CO and TH.

I used the multivariate spatial-temporal outbreak simulator proposed by Zhang and Wallstrom [55] described in Chapter 4.3. This model is able to simulate outbreak data simultaneously in more than one data types by assuming that the counts for each data type follow either the same or different random processes. In this study, I used the simple version to generate outbreak data for the three data types, AF, TH and CO by assuming the cases for all the data types are drawn from linear Poisson processes with increasing means which are proportional to the data means. The behavior vector in this model was set to be (1, 1, 1, 0, 0, 0, 0, 0). This means the model does not consider any of the behaviors that one patient will contribute to more than one data types at the same time interval, i.e., that a patient will not purchase an anti-fever medication from pharmacy if he/she has been to an ED and vice versa. The spatial disease risk function was set to be flat, which means the risks of infected regions z show no differences in terms of their distance to the outbreak center.

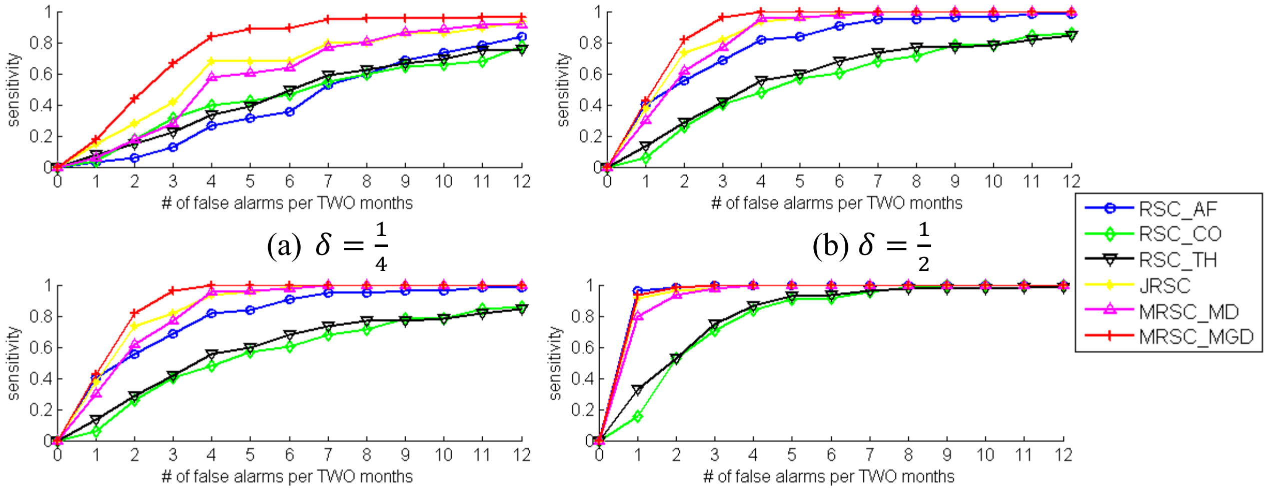
I ran 4 groups of simulations in order to compare the detection power of different methods, given that the background data injected with outbreaks had varied intensities. Each outbreak was arbitrarily set to have a $T = 7$ day duration and to infect at most 8 ZIP code areas. The total injected counts was set to be $\delta\mu_k T$, where δ was chosen from $\{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ and μ_k is the average daily counts of data stream D_k. Each group included 100 different outbreaks, and each of them was superimposed on the background data stream with a randomly selected outbreak start date.

The outbreak effect vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3, \pi_4)$ was set to be $(\pi_1, 1,1,1)$ where π_1 is equally likely to be any value in the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} (i.e., discrete uniform distribution).

I compared the MRSC algorithm's performance analyzing the four data streams simultaneously with that of the RSC algorithm's performance analyzing each data stream separately, and I also compared MRSC with the joint results of RSC algorithms (JRSC) by simply computing the product of the three posterior probabilities of a cluster with each computed using univariate RSC analyzing one of the three disease-relevant data streams. The MRSC algorithm applying the Multinomial-Dirichlet model is called MRSC_MD and the one applying the Multinomial generalized Dirichlet model is called MRSC_MGD. The univariate detectors applied to different data streams are named as RSC_AF, RSC_CO and RSC_TH, respectively.

In Figure 6.3 and Figure 6.4, I plot both the partial ROC curves and the partial AMOC curves with false positive rates less than 12 per two months, respectively. I computed the areas under the partial ROC curves and tested the difference (Table 6.1). In all four groups, MRSC_MGD had the best performance in terms of ROC curves and areas under the curves; but there are no significant differences between the MRSC_MGD and the others for all δ ($\delta = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$).

As a practical summary measure, I considered the average "days to detect" for each method, at a fixed false positive rate of 1 per month. For this measure, any missed outbreaks are penalized for the entire duration of the outbreak, and thus counted as $t = 7$ days to detect. The detection performance for each method for each of the four outbreak intensities is presented in Table 6.2.



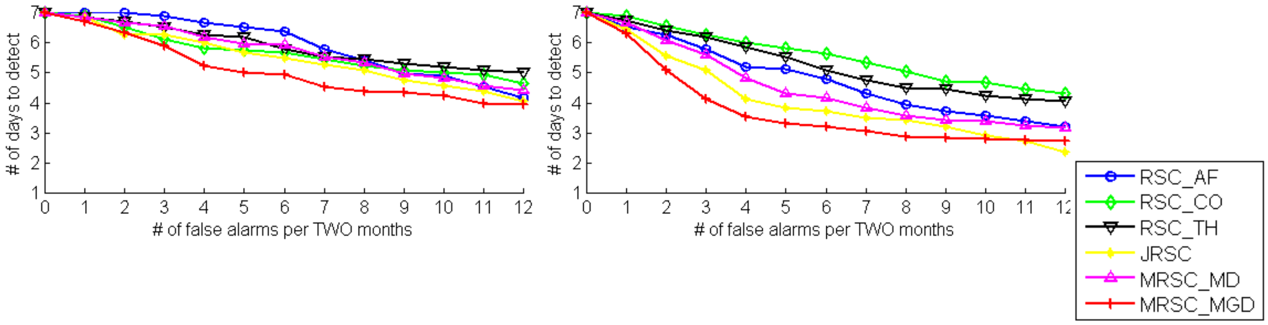
(a) $\delta = \frac{1}{4}$

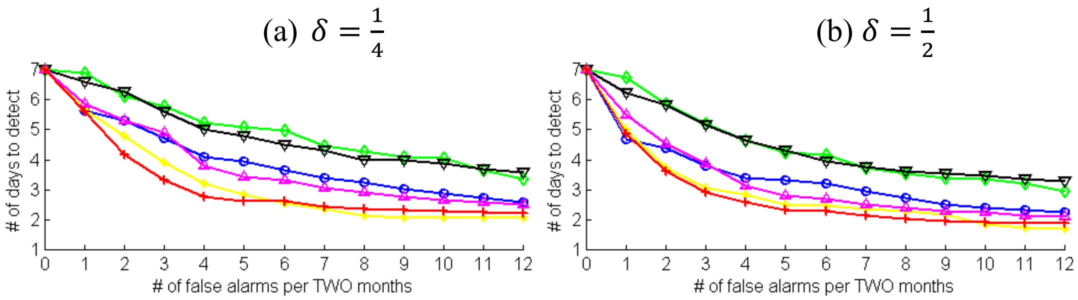
(b) $\delta = \frac{1}{2}$

(c) $\delta = \frac{3}{4}$

(d) $\delta = 1$

Figure 6.3 ROC curves of the four groups of experiments





(c) $\delta = \frac{3}{4}$ (d) $\delta = 1$

Figure 6.4 AMOC curves of the four groups of experiments

Table 6.1 Comparison of the areas under the partial ROC curves with false positive rates in the range [0, 12]. The underscored results are the best performance and those in bold are not significantly different (at $\alpha = 0.05$) from the best.

δ	MRSC_MGD	MRSC_MD	JRSC	RSC_AF	RSC_CO	RSC_TH
$\frac{1}{4}$	**0.1523**	**0.1171**	**0.1230**	**0.0822**	**0.0915**	**0.0891**
$\frac{1}{2}$	**0.1787**	**0.1683**	**0.1701**	**0.1584**	**0.1161**	**0.1212**
$\frac{3}{4}$	**0.1866**	**0.1822**	**0.1834**	**0.1842**	**0.1405**	**0.1449**
1	**0.1959**	**0.1898**	**0.1902**	**0.1928**	**0.1572**	**0.1611**

Table 6.2 Average days to detect at 1 false alarm per month, for each of the 4 groups of simulations. The underscored results are the best performance and those in bold are not significantly different (at $\alpha = 0.05$) from the best.

δ	MRSC_MGD	MRSC_MD	JRSC	RSC_AF	RSC_CO	RSC_TH
$\frac{1}{4}$	**6.33**	**6.67**	**6.32**	6.99	**6.68**	**6.72**
$\frac{1}{2}$	**5.09**	6.08	5.74	6.36	6.56	6.38
$\frac{3}{4}$	**4.18**	5.30	4.82	5.42	6.32	6.33
1	**3.60**	4.52	**3.78**	4.51	5.96	5.76

As can be seen in Table 6.2, considering multiple data streams altogether, MRSC_MGD was able to detect outbreaks earlier than MRSC_MD, JRSC and the univariate detectors in three sets of experiments when $\delta = \frac{1}{2}, \frac{3}{4}, 1$ given the false alarm rate is 1 per month. With the same false alarm rate, the achieved timeliness of MRSC_MGD is significantly better than MRSC_GD and the three univariate detectors RSC's in three out of four groups when $\delta = \frac{1}{2}, \frac{3}{4}, 1$. MRSC_MGD also has significantly better timeliness than JRSC when $\delta = \frac{1}{2}, \frac{3}{4}$.

6.5.3 Experiment II – detecting diarrhea disease outbreaks

To further test my hypothesis that multivariate approaches can be helpful for early outbreak detection, I set up a second set of experiments modeling diarrhea disease outbreaks. I believed that a diarrhea outbreak would have effects on two of the seven data streams: Diarrhea remedies OTC (DR) and Gastrointestinal ED (GI). The other five data streams I believed would remain unaffected and so their daily counts were summed together to form the data stream called "Others". The input data streams for our MRSC algorithm had the order of Others, DR and GI.

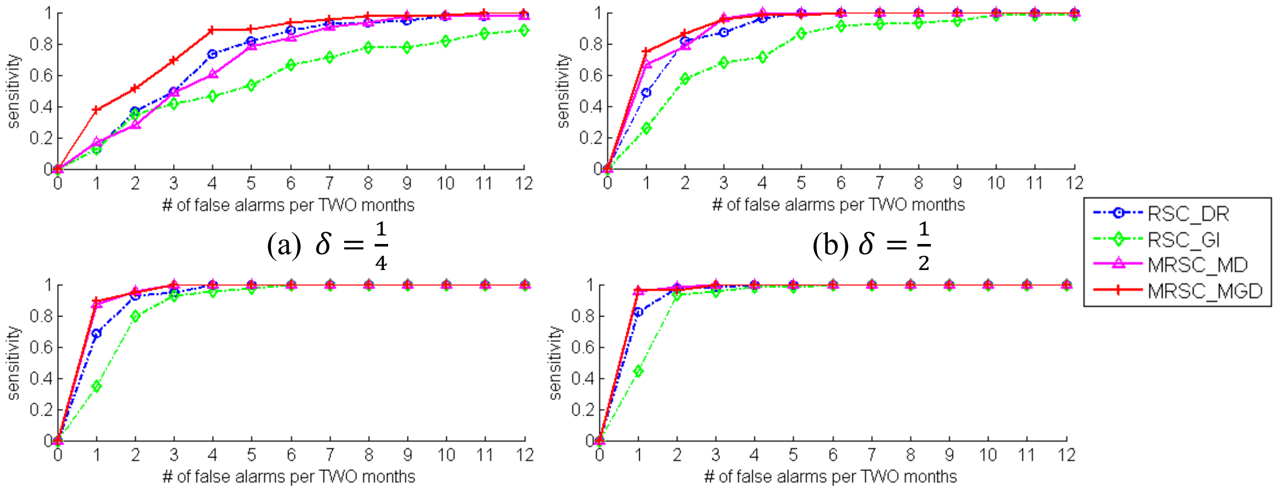
I used the same multivariate spatial-temporal event simulator proposed by Zhang and Wallstrom [55] used in Experiment I. Also, I again used the simple version to generate outbreak data for the two affected data types, DR and GI, by assuming the cases for all the data types are drawn from linear Poisson processes with increasing means proportional to the data means. The behavior vector in this model was set to be (1, 1, 0, 0), which again assumes that no patient contributes to more than one data type. The spatial disease risk function was set to be flat, meaning the risks of the infected regions s have no differences in terms of their distance to the outbreak center.

As in Experiment I, I ran 4 groups of simulations in order to compare the detection power of different methods across outbreak scenarios with varied intensities. Also as above, each outbreak was arbitrarily set to have a $T = 7$ day duration, and each infected at most 8 ZIP code areas. The total injected counts was set to be $\delta\mu_k T$, where δ was chosen from $\{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$ and μ_k is the average daily counts of data stream D_k. Each group included 100 different outbreaks, and each outbreak was superimposed on the background data stream with a randomly selected outbreak start date.

The outbreak effect vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \pi_3)$ was set to be $(\pi_1, 1,1)$ where π_1 is equally likely to be any value in the set {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9} (i.e., (discrete uniform distribution). The RSC algorithms' analyses of each data stream were named RSC_DR and RSC_GI, respectively.

In Figure 6.5 and Figure 6.6, I plot the partial ROC curves and the partial AMOC curves for the algorithms, respectively. I computed the areas under the partial ROC curves and tested the difference (Table 6.3). MRSC_MGD had the best performance in terms of ROC curves and areas under the curves in all four groups of experiments, but the differences were not significant.

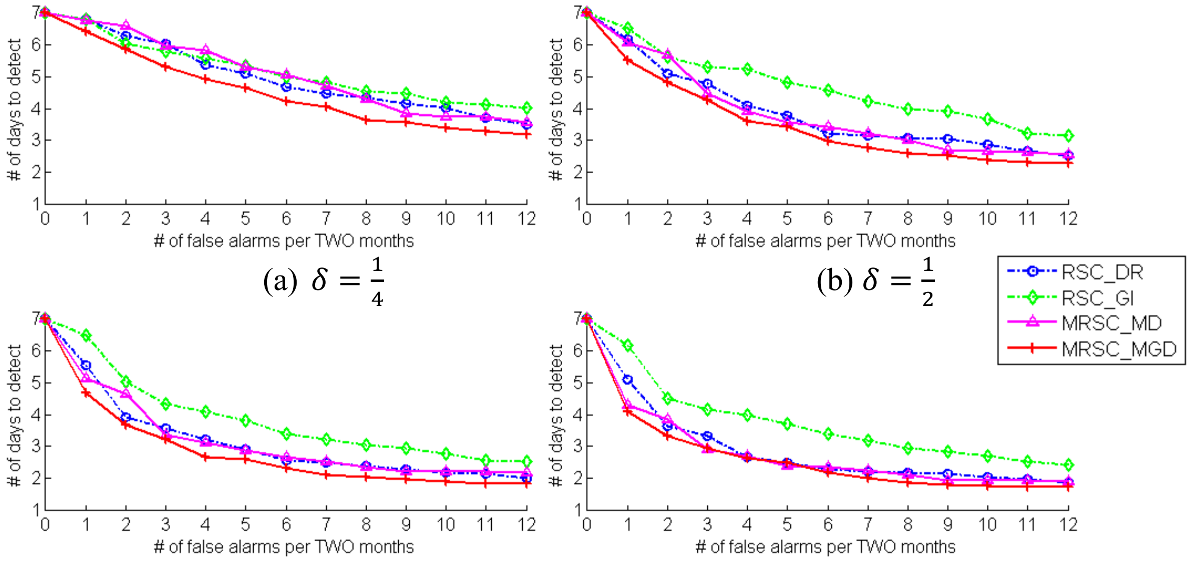
The timeliness performance for each method, for each of the four outbreak intensities, is presented in Table 6.4. Again, I measured the average "days to detect" for each method at a fixed false positive rate of 1 per month. MRSC_MGD was the best performing algorithm in all four groups where $\delta = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$. It detected the outbreaks significantly earlier than RSC_GI when $\delta = \frac{1}{2}, \frac{3}{4}, 1$ and than MRSC_MD when $\delta = \frac{1}{4}, \frac{1}{2}, \frac{3}{4}$.



(a) $\delta = \frac{1}{4}$ (b) $\delta = \frac{1}{2}$

(c) $\delta = \frac{3}{4}$ (d) $\delta = 1$

Figure 6.5 ROC curves of the four groups of experiments



(a) $\delta = \frac{1}{4}$ (b) $\delta = \frac{1}{2}$

(c) $\delta = \frac{3}{4}$ (d) $\delta = 1$

Figure 6.6 AMOC curves of the four groups of experiments

Table 6.3 Comparison of the areas under the partial ROC curves with false positive rates in the range [0, 12]. The underscored method is the best-performing method and methods in bold are not significantly different (at $\alpha = 0.05$) from the best..

δ	MRSC_MGD	MRSC_MD	RSC_DR	RSC_GI
$\frac{1}{4}$	**0.1613**	**0.1418**	**0.1458**	**0.1177**
$\frac{1}{2}$	**0.1845**	**0.1816**	**0.1792**	**0.1558**
$\frac{3}{4}$	**0.1921**	**0.1904**	**0.1868**	**0.1754**
1	**0.1947**	**0.1932**	**0.1915**	**0.1804**

Table 6.4 Average days to detect at 1 false alarm per month, for each of the 4 groups of simulations. The underscored method is the best-performing method and methods in bold are not significantly different (at α=0.05) from the best.

δ	MRSC_MGD	MRSC_MD	RSC_DR	RSC_GI
$\frac{1}{4}$	**5.86**	6.59	**6.26**	**6.03**
$\frac{1}{2}$	**4.81**	5.69	**5.10**	5.62
$\frac{3}{4}$	**3.66**	4.63	**3.93**	5.01
1	**3.33**	**3.83**	**3.64**	4.50

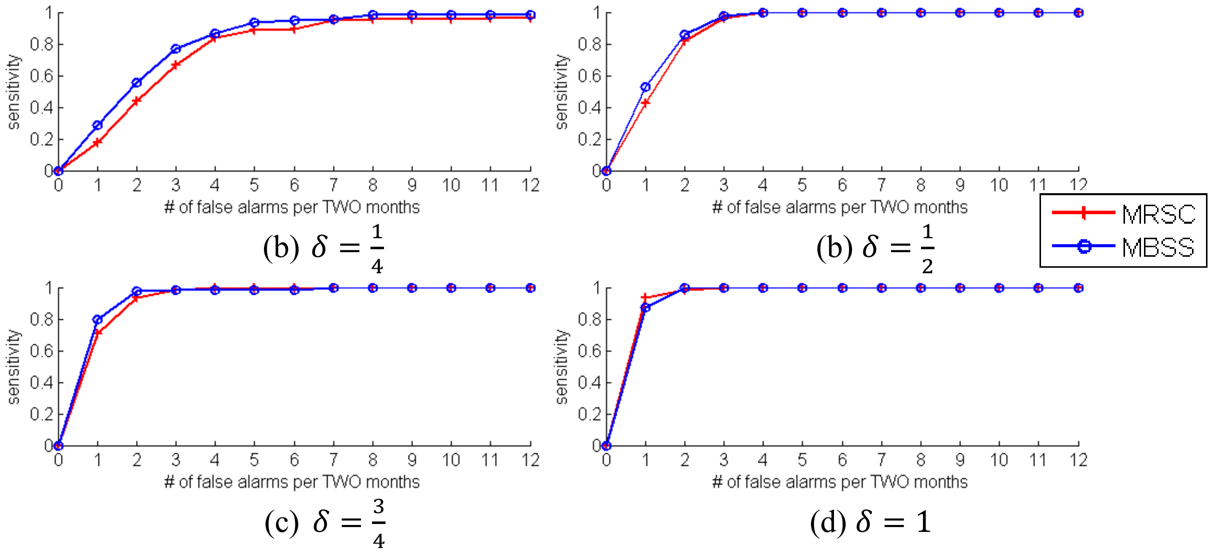
In Experiment I and II, MRSC applying the Multinomial-generalized-Dirichlet model (MRSC_MGD) consistently performed better than that applying the Dirichlet model. The results confirm the fact I mentioned in Chapter 2.3.4 that when $\boldsymbol{X} = (X_1, X_2, \cdots, X_K)$ has a Dirichlet distribution any two random variables in $\boldsymbol{X}$ will be negatively correlated. However, in our cases, two random variables may be positively correlated (e.g., elevations in all the disease-relevant data streams), and hence the Dirichlet distribution would not be the better choice to be a prior distribution in our multivariate analysis than the generalized Dirichlet distribution. Therefore, in the following experiments, I only focus on the study using the MGD model.

6.5.4 Experiment III – comparing to other multivariate algorithms

As described in Chapter 3.3, some researchers have proposed multivariate approaches for use in the field of disease surveillance. In this sub-chapter, I compare MRSC_MGD with two other multivariate algorithms named the multivariate scan statistic by Kulldorff et al (MKSS) [40] and the multivariate Bayesian scan statistic (MBSS) by Neill et al[43]. MKSS is an extension of a very well-known algorithm, the spatial scan statistic, and its implementation (SaTScan) have been used in many areas [88,89]. MBSS is an extension of the Bayesian spatial scan statistic and has been demonstrated to have better performance than MKSS [40]. Like MRSC_MGD, MBSS utilizes Bayes' Theorem to compute the posterior probability of each potential cluster. In the following, I show the results of the comparison on two experimental data sets. One is the same flu data set as used in Experiment I. The other is a data set injected with both simulated flu cases and cases caused by some background events (e.g., big conference meetings or super bowl games). I used the multivariate spatial and temporal event simulator, described in Chapter 4.3 to generate the cases in both data sets.

Simulated flu data set

MKSS and MBSS were applied to the flu data set and the results were compared with MRSC_MGD's. I applied a purely spatial model and standard Monte Carlo method (for the computation of p-value) in SaTScan (MKSS implementation) in MKSS. However, the algorithm generated 6 false alarms per month when analyzing historical data (believed to have no underlying outbreaks) with the lowest threshold p-value at 0.001, which means the algorithm cannot detect outbreaks given a false alarm rate lower than 6 per month (see 7.2Appendix B for parameter setting of SaTScan). For MBSS, the grid size was set to be 16 by 16. The ROC and AMOC curves of MRSC and MBSS are showed in Figure 6.7 and Figure 6.8 (note no performance of MKSS can be shown in the figures with false alarm rates less than 12 per TWO months).



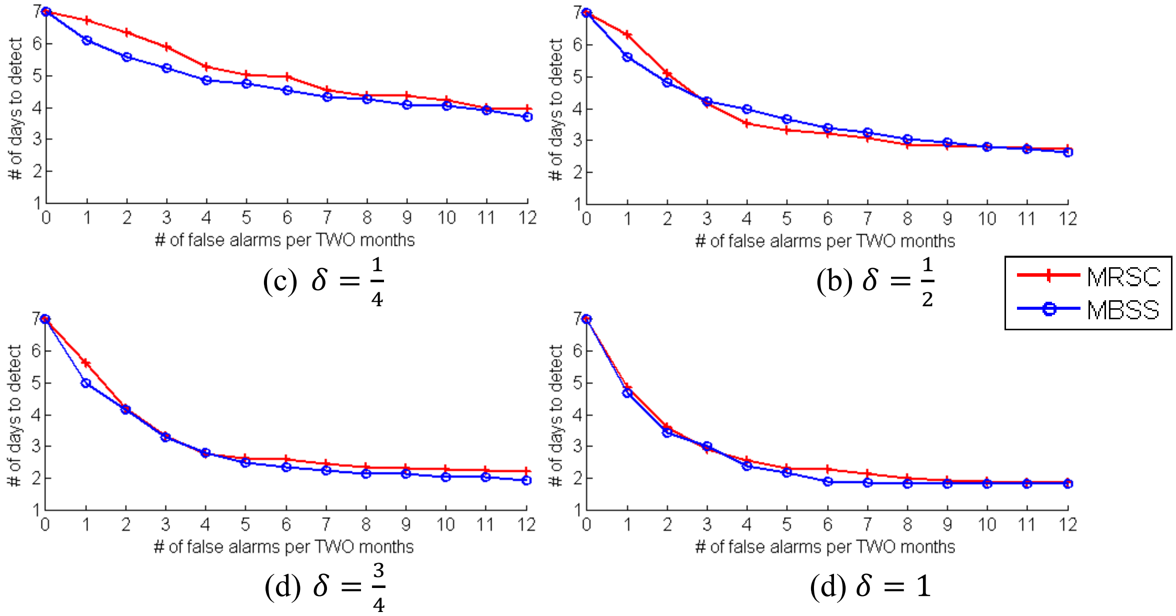
(b) $\delta = \frac{1}{4}$

(b) $\delta = \frac{1}{2}$

(c) $\delta = \frac{3}{4}$

(d) $\delta = 1$

Figure 6.7 ROC curves of MRSC_MGD and MBSS analyzing three data streams in the flu data sets.



(c) $\delta = \frac{1}{4}$

(b) $\delta = \frac{1}{2}$

(d) $\delta = \frac{3}{4}$

(d) $\delta = 1$

Figure 6.8 AMOC curves of MRSC_MGD and MBSS analyzing three data streams in the flu data sets.

Table 6.5 Comparison of the areas under the partial ROC curves with false positive rates in the range [0, 12]. The underscored method is the best-performing method and methods in bold are not significantly different (at $\alpha = 0.05$) from the best..

δ	MRSC_MGD	MBSS
$\frac{1}{4}$	**0.1523**	**0.1685**
$\frac{1}{2}$	**0.1787**	**0.1835**

$\frac{3}{4}$	**0.1866**	**0.1899**
1	**0.1959**	**0.1928**

Table 6.6 Average days to detect at 1 false alarm per month, for each of the 4 groups of simulations. The underscored method is the best-performing method and methods in bold are not significantly different (at α=0.05) from the best.

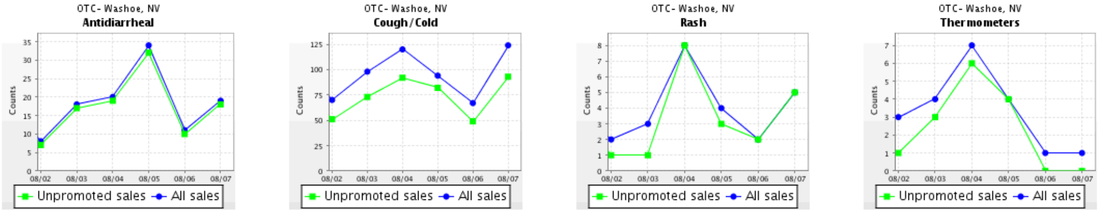
δ	MRSC_MGD	MBSS
$\frac{1}{4}$	6.33	**5.58**
$\frac{1}{2}$	**5.09**	**4.80**
$\frac{3}{4}$	**4.18**	**4.16**
1	**3.60**	**3.44**

In the experiments, the average running times for the MRSC_MGD, MBSS and MKSS are 0.54 seconds, 0.23 seconds and 2.53 seconds respectively. MBSS ran faster than two other spatial algorithms. One reason is that in [43], the likelihood ratio for each spatial location is pre-computed by the algorithm MBSS. When computing the log-likelihood ratio for a given spatial region, the algorithm only needs to sum the log-likelihood ratios for all locations in that region. The algorithm has an added benefit that the expensive likelihood ratio computations are only performed a number of times proportional to the number of locations, rather than the (much larger) number of regions. Note again that the running time of MKSS cannot be compared strictly with others because it was implemented in SaTScan v9.1 which was programmed in different language C.

We can see from the comparison that MBSS performed slightly better than MRSC_MGD. Statistically speaking, there is no significant difference between the two algorithms in most of the experiments except that MBSS detected outbreaks significantly earlier than MRSC_MGD given at most one false alarm per month is allowed when the outbreaks are weak ($\delta = \frac{1}{4}$, at $\alpha = 0.05$). One of the possible reasons that MRSC_MGD did not perform as well as MBSS did might be due to the incorporation of the non-disease-relevant data streams in the analysis. Because of the nature of MGD model applied in MRSC, the algorithm cannot detect the outbreaks if the non-disease relevant data streams also have elevated signals in the same areas where outbreak cases occur. In addition, if there is a drop of the counts in the non-disease relevant data streams, the algorithm with MGD model would also have signaled a false alarm.

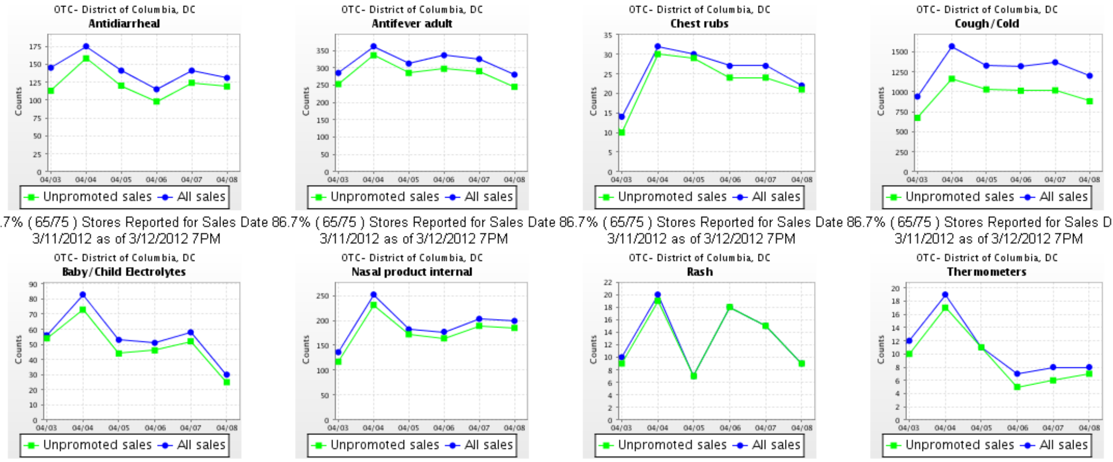
Simulated flu data set with background events

Social events, such as super bowl games, Olympics, conference meetings, etc., can cause population surges at event locations. A big population surge may cause elevated signals in biosurveillance systems. However, different from disease outbreaks, elevated signals may occur in most of the data streams, not only disease-relevant data streams. For example, the Hot August Nights event in Reno, Nevada (Washoe County) usually brings 50-60K people to the city every year during the first week of August where the city population is normally only about 220K. Our RODS system captured simultaneously increased counts of OTC anti-diarrhea, rash, anti-fever, cough/cold medication sales in Washoe during every first week of August when this event is held in Reno (Figure 6.9). Another example where increased population affects data streams is the Cherry Blossom Festival in DC between the end of March and the beginning of April every year when people go to celebrate the peak bloom period. Figure 6.10 shows eight data streams that almost have simultaneous elevations because of the event.



51.3% (20/39) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 51.3% (20/39) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 51.3% (20/39) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 51.3% (20/39) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM

Figure 6.9 Four data streams of NRDM categories (Anti-Diarrhea, Cough/Cold, Rash and Thermometers) between Aug. 2, 2008 and Aug. 7, 2008 in Washoe County Nevada.

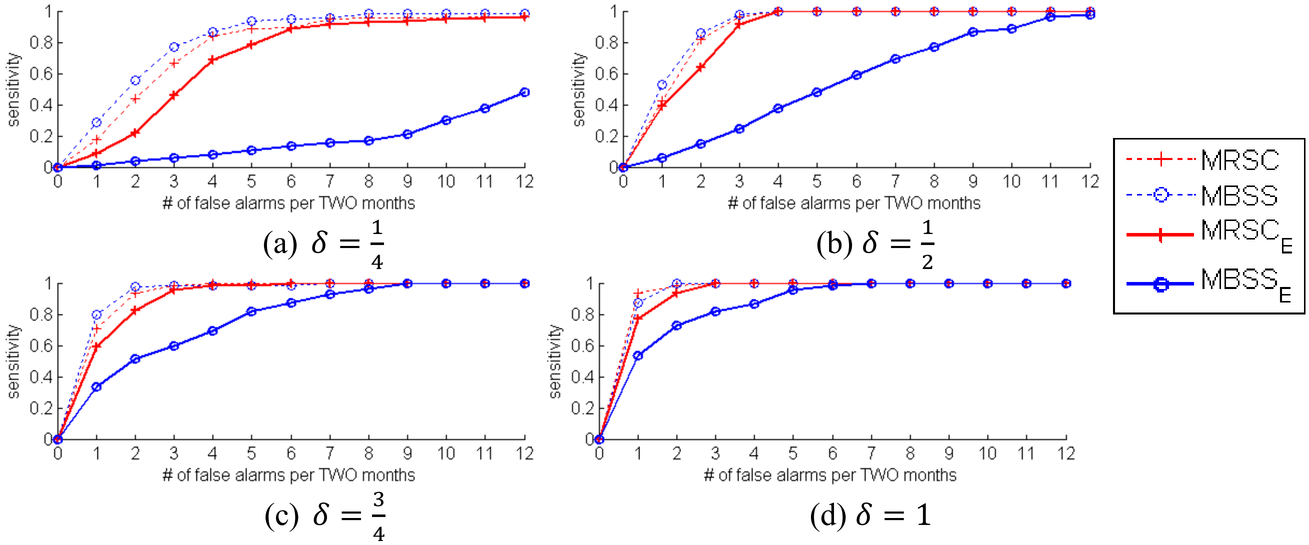


86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM

86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM 86.7% (65/75) Stores Reported for Sales Date 3/11/2012 as of 3/12/2012 7PM

Figure 6.10 Eight data streams of NRDM categories (Anti-Diarrhea, Anti-Fever Adult, Chest Rubs, Cough/Cold, Baby/Child Electrolytes, Nasal Products, Rash and Thermometers) between Apr. 3, 2011 and Apr. 8, 2011 in Washington DC.

In order to demonstrate the hypothetical advantage of MRSC_MGD by modeling not only the disease-relevant data streams but also other non-disease-relevant data streams, I simulated such background events using the multivariate spatial temporal event simulator as well (see Chapter 4.3). I assumed that cases captured during social events follow the flat template (with a random process) in both affected spatial and temporal dimensions. Since it is common for the number of cases in different data streams to increase to some similar extent due to the population surge, the magnitudes were set to be same for all the data streams including not only the disease-relevant but also the non-disease-relevant ones. Ten events were simulated and the cases were injected into all the background streams of the affected spatial areas. The magnitudes of the events were set to 1.0 ($\delta = 1.0$). Each event covered eight ZIP codes and lasted for seven days. The simulated flu outbreaks used in Experiment I were then injected on the top of the background data. I then tested the detection performances of the two multivariate algorithms MRSC_MGD and MBSS on the data sets with injected cases caused by either outbreaks or underlying events. To distinguish the algorithm performance from that of the MRSC_MGD and MBSS applied to the data sets without background events, I call these MRSC$_E$ and MBSS$_{E,}$ respectively. Both ROC curves and AMOC curves are showed in Figure 6.11 and Figure 6.12, respectively.



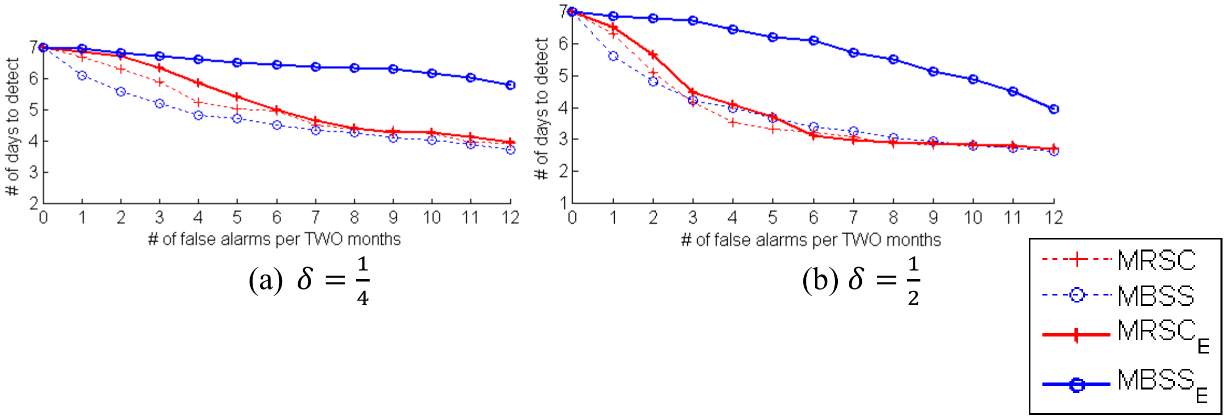
(a) $\delta = \frac{1}{4}$

(b) $\delta = \frac{1}{2}$

(c) $\delta = \frac{3}{4}$

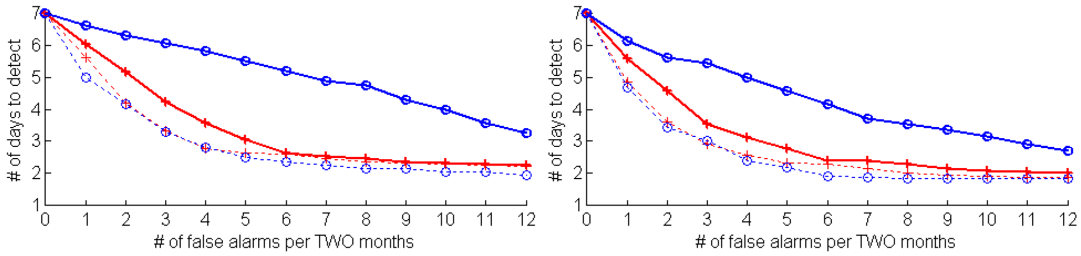
(d) $\delta = 1$

Figure 6.11 ROC curves of the algorithms, MRSC$_E$ and MBSS$_E$, representing the results of the algorithms applied on the data set injected with background events, and their previous performances on the data set without injected background events.



(a) $\delta = \frac{1}{4}$

(b) $\delta = \frac{1}{2}$



(c) $\delta = \frac{3}{4}$

(d) $\delta = 1$

Figure 6.12 AMOC curves of the algorithms.

We can see from Figure 6.11 and Table 6.7 that MRSC_E had overall greater AUC than MBSS_E and it performed significantly better than MBSS_E when $\delta = \frac{1}{4}$. I also measured the performance drop of AUC when applying these two algorithms to the data sets with simulated non-disease-related background events. As shown in Table 6.7, MRSC_MGD's performance dropped only 4.54% on average while MBSS dropped 37.0% through the four sets of experiments with different outbreak magnitudes. The performance drop of MBSS is more significant when the outbreaks are weak, as shown in Figure 6.11(a) and (b).

As shown in Figure 6.12 and Table 6.8, MRSC_E was able to detect outbreaks earlier than MBSS_E in all four sets of experiments and the timeliness was significantly better when $\delta = \frac{1}{2}, \frac{3}{4}, 1$, given that only one outbreak per month is allowed. In addition, MBSS_E took on average 1.88 days longer than MBSS to detect outbreaks, compared with MRSC_MGD_E which took 0.73 days longer than MRSC_MGD. The results showed that the MGD model used in MRSC provides more robustness when there are simultaneous increases in both disease-relevant and non-disease-relevant data streams.

Table 6.7 Areas under partial ROC curves with false positive rates in the range [0, 12] and the performance drop after including background events. δ represents outbreak intensity. The underscored method is the best-performing method and methods in bold are not significantly different (at $\alpha = 0.05$) from the best.

δ	No Background Events	With Background	Performance Drop (%)

			Events			
	MRSC_MGD	MBSS	MRSC_MGD	MBSS	MRSC_MGD	MBSS
$\frac{1}{4}$	**0.1523**	**0.1685**	**0.1384**	0.0314	9.13%	81.4%
$\frac{1}{2}$	**0.1787**	**0.1835**	**0.1738**	**0.1107**	2.74%	39.7%
$\frac{3}{4}$	**0.1866**	**0.1899**	**0.1820**	**0.1556**	2.47%	18.1%
1	**0.1959**	**0.1928**	**0.1884**	**0.1762**	3.83%	8.61%

Table 6.8 Average days to detect at 1 false alarm per month and the number of days delayed after including background events. δ represents outbreak intensity. The underscored method is the best-performing method and methods in bold are not significantly different (at α=0.05) from the best.

δ	No Background Events		With Background Events		Performance Drop (days)	
	MRSC_MGD	MBSS	MRSC_MGD	MBSS	MRSC_MGD	MBSS
$\frac{1}{4}$	6.33	**5.58**	**6.74**	**6.82**	0.41	1.24
$\frac{1}{2}$	**5.09**	**4.80**	**5.63**	6.78	0.54	1.98
$\frac{3}{4}$	**4.18**	**4.16**	**5.16**	6.29	0.98	2.13
1	**3.60**	**3.44**	**4.58**	5.61	0.98	2.17

6.5.5 Experiment IV – varied outbreak effects on different data streams

Recall in Chapter 6.4.2 I introduce how to model the outbreak effects on different disease-relevant data streams. Here I demonstrate the performance of MRSC_MGD where data streams respond to outbreaks differently (I call it VMRSC_MGD). It is easy to simulate such outbreak cases using the multivariate spatial temporal event simulator by defining different values in the data coverage vectors (see Chapter 4.3).

I compared the performance of VMRSC_MGD, which applies varied effect values, and MRSC_MGD, which applies the same effect values for all disease-relevant data streams. I simulated six groups of outbreaks with different data coverage on different data streams. Each set of effect values (π_k, $k \geq 2$) used in VMRSC_MGD was computed from the proportionality constants ($a_2, a_3, \ldots, a_K$) of the increase percentages (Δ_k's, $k \geq 2$) which were learned from simulated outbreaks using maximum likelihood. More specifically, given a set of infected areas $z \in Z$ and their historical mean counts μ_z, I used the multivariate spatial temporal event simulator to generate 100 sets of outbreaks cases for each outbreak strength δ where $\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. The data coverage vector for different data streams was also pre-defined for outbreak simulation. Thus, for each data coverage vector, we have 400 sets of outbreak cases to learn from. Given the knowledge of where these cases occurred as well, we can then calculate the proportionality constants among the multiple data streams with maximum likelihood. Table 6.9 lists the normalized proportionality constants of the increase percentages for those disease-relevant data streams learned from each outbreak sample set. The constants were then used to compute corresponding π_k values in the model, as described in Chapter 6.4.2.

Table 6.9 Normalized proportionality constants of the increase percentages for the disease-relevant data streams learned from simulated outbreaks.

Data Coverage (AF,CO,TH)	**Normalized Proportionality Constants (a_{AF}, a_{CO}, a_{TH})**	**Data Coverage (AF,CO,TH)**	**Normalized Proportionality Constants (a_{AF}, a_{CO}, a_{TH})**
(1.0,0.5,0.5)	(0.51,0.25,0.24)	(0.5,1.0,1.0)	(0.20,0.41,0.39)
(0.5,1.0,0.5)	(0.25,0.47,0.28)	(1.0,0.5,1.0)	(0.40,0.19,0.41)
(0.5,0.5,1.0)	(0.25,0.27,0.48)	(1.0,1.0,0.5)	(0.41,0.41,0.18)

Table 6.10 shows the average number of days the algorithms took to detect the outbreaks. We can see from the table that in most experiments, VMRSC_MGD performed better than MRSC_MGD in terms of detection timeliness when the data streams included outbreak signals with different strengths δ. When the injected outbreak strength was big ($\delta = 1$), VMRSC_MGD was able to detect outbreaks significantly earlier in 5 out of 6 groups of experiments because of the prior knowledge of different outbreak effects on different data streams.

Table 6.10 Average number of days to detect the outbreaks using varied effect parameters in MRSC_MGD (VMRSC_MGD) and fixed effect parameters (MRSC_MGD). The underscored method is the best-performing method and methods in bold are not significantly different (at α=0.05) from the best.

Data Coverage (AF,CO,TH)	δ	**VMRSC_MGD**	**MRSC_MGD**	**Data Coverage (AF,CO,TH)**	δ	**VMRSC_MGD**	**MRSC_MGD**
1.0,0.5,0.5	$\frac{1}{4}$	**6.03**	**6.16**	0.5,1.0,1.0	$\frac{1}{4}$	**6.64**	**6.72**
	$\frac{1}{2}$	**5.19**	**5.45**		$\frac{1}{2}$	**5.93**	**6.11**
	$\frac{3}{4}$	**4.14**	**4.43**		$\frac{3}{4}$	**5.19**	**5.39**
	1	**3.34**	3.78		1	**4.62**	5.19
0.5,1.0,0.5	$\frac{1}{4}$	**6.60**	**6.68**	1.0,0.5,1.0	$\frac{1}{4}$	**6.13**	**6.22**
	$\frac{1}{2}$	**6.07**	**6.05**		$\frac{1}{2}$	**4.92**	**5.20**
	$\frac{3}{4}$	**5.5**	**5.63**		$\frac{3}{4}$	**4.20**	4.69
	1	**5.01**	**4.91**		1	**3.59**	4.16
0.5,0.5,1.0	$\frac{1}{4}$	**6.85**	**6.83**	1.0,1.0,0.5	$\frac{1}{4}$	**6.21**	**6.26**
	$\frac{1}{2}$	**6.1**	**6.21**		$\frac{1}{2}$	**4.86**	**4.90**
	$\frac{3}{4}$	**5.58**	**5.67**		$\frac{3}{4}$	**4.22**	**4.47**
	1	**4.68**	5.18		1	**3.56**	4.03

6.5.6 Experiment V – detecting outbreaks with time lag effects

In this set of experiments, I test the hypothesis that the MGD model can be helpful for detection outbreaks with time lagged signals by setting larger effect values for the data streams which respond earlier and smaller effect values for lagged data streams. I use the flu data set described in Experiment I, which includes three disease-relevant data streams including AF, CO and TH. I set up three sets of experiments, each of which has one data stream injected with delayed outbreak cases. Table 6.11 shows the average number of days to detect the outbreaks with time lags in one data stream using MRSC_GMD with varied outbreak effect values (VMRSC_MGD) versus MRSC_MGD with the same outbreak effects. We can see from the table that VMRSC_MGD performed consistently better than MRSC_MGD in most of the experiments, and it was significantly better (at $\alpha = 0.05$) when the strength of the outbreaks $\delta = 1$ when the signal in either data stream AF or CO has lagged time.

Table 6.11 Average days to detect the outbreaks using MRSC_MGD with varied outbreak effect values and MRSC_MGD with same outbreak effect values. The underscored method is the best-performing method and methods in bold are not significantly different (at α=0.05) from the best.

# of lagged days (AF,CO,TH)	Effect Ratios Used For VMRSC_MGD (Δ_{AF},Δ_{CO},Δ_{TH})	δ	VMRSC_MGD	MRSC_MGD
(2, 0, 0)	(0.2,0.4,0.4)	$\frac{1}{4}$	**6.64**	**6.72**
		$\frac{1}{2}$	**5.96**	**6.22**
		$\frac{3}{4}$	**5.15**	5.69
		1	**4.48**	5.02
(0, 2, 0)	(0.4,0.2,0.4)	$\frac{1}{4}$	**6.16**	**6.25**

		$\frac{1}{2}$	**4.86**	**5.22**
		$\frac{3}{4}$	**4.21**	**4.59**
		1	**3.59**	4.17
(0, 0, 2)	(0.4,0.4,0.2)	$\frac{1}{4}$	**6.16**	**6.13**
		$\frac{1}{2}$	**5.09**	**5.15**
		$\frac{3}{4}$	**4.63**	**4.54**
		1	**3.67**	**3.86**

6.6 Discussion

In Experiment I and Experiment II, where I compared the performances among six algorithms: two MRSC algorithms using either MGD model or MD model, the joint RSC algorithm by multiplying the cluster posteriors computed from three disease-relevant data streams and the three univariate detectors. The results supported the hypothesis that the integration of information from multiple data streams is essential for detecting emerging outbreaks at the early stages. When detecting flu-like disease outbreaks, MRSC_MGD was able to detect the outbreak 1.05 days on average earlier than the best of the univariate detectors at a false alarm rate of one per month. When detecting diarrhea disease outbreaks, MRSC_MGD was able to detect the outbreak 0.27 days on average earlier than the best of the univariate detectors.

In addition, I found that the RSC applied to the data stream with the largest mean values will perform better than when applied to the other data streams, for example, the

RSC_AF in Experiment I and the RSC_DR in Experiment-II were able to detect outbreaks earlier than the other univariate detectors most of the time. I measured the average means and the standard deviations of these data streams, the results of which are shown in Table 6.12. I also calculated the coefficients of variation to show the normalized measures of data dispersion, which are computed as the ratios of the standard deviations to the means. As we can see from Table 6.12, AF and DR are two data streams that have minimal values for the coefficient of variation, meaning they have the least dispersion. This, to some extent, may indicate they were less influenced by the noise and thus presented data sets with better quality.

Table 6.12 Average mean and standard deviation of each study data stream of 54 ZIP codes in Allegheny County.

	AF	CO	TH	DR	GI
Avg. Mean	26.17	1.43	0.74	6.09	2.02
Avg. Std	8.56	1.37	0.94	2.91	1.56
Coefficient of Variation	0.33	0.96	1.27	0.48	0.77

In addition to the data streams we want to surveill, considering which data streams should be counted into the “Others” type in the MGD model is not a trivial problem. In other words, since we are modeling the changes of the proportions of the disease-specific data streams among all the data streams, choosing a “good” control group is important. In this study, I included seven data streams in the experiments, anti-fever OTC, diarrhea remedies OTC, thermometer OTC, hydrocortisones OTC, gastrointestinal ED, constitutional ED and rash ED. Some other data streams, such as stomach remedies OTC and neurological ED were not counted in the “Others” data type. There are two reasons for being selective for the data streams included in the “Others” data type. First, we want to improve the model’s overall detection capability which means it should be less vulnerable to dramatic and unpredictable shifts in the health-care data that it analyzes. This is because real world data are subject to all kinds of shifts that occur during major public events, such as the Olympics, as a result of population surges and public closures, and during promotions throughout the pharmacy store chains. Shifts can also occur during epidemics and pandemics, with the worried-well buying medications or flooding emergency departments but not actually having the disease. More often, public health data streams also have shifts due to the day-of-week effect. Thus, the data streams included in the "denominator" should be able to reflect the same shift effects which are unrelated to the monitored disease to some extent. The seven selected data streams share the same properties in that they are all infectious disease related and they are all subject to some non-stationary shifts such as seasonal effects or marketing strategies. Second, we do not want the mean count of the "Others" data stream in the model to be extremely big or it will overwhelm the other data streams in the model. I thus simply excluded all of the disparate and unrelated data types.

In addition, to form the control group, I used multiple available non-disease-relevant data streams. One reason is that some single control group has small sample size and we will prefer to include more controls when these data can be available. I simply summed the counts from the multiple control streams without prior knowledge about the relations among the control streams. One other way to explore is to normalize the counts such that each data stream provides same weights.

One potential advantage of MRSC_MGD over other multivariate algorithms (e.g., MKSS and MBSS) is that MRSC_MGD is able to adjust the baseline by incorporating the data type "Others", which actually helps to adjust the elevated baseline counts caused by population surges, some non-disease-related events or the day-of-week effects. For example, if a public event (e.g., a big conference) causes significant elevations on all the data streams under surveillance, algorithms only analyzing those specific data types will signal a false alarm since all of the monitored data types are presenting elevations. On the other hand, if all the data types are elevated to the same extent, the proportions of the specific data types will stay the same. In this way the MGD model becomes more robust than other algorithms which only analyze the specific data types of interest. The results shown in Experiment III – comparing to other multivariate algorithms demonstrated this advantage.

Consequently, one limitation of MRSC is that it cannot work if there is no control group available because it is designed to detect relative shifts rather than absolute shifts based on the nature of the model. It is suggested to use MBSS or other multivariate algorithms to detect absolute baseline shifts.

In Experiment III I compared MRSC_MGD with two other multivariate algorithms, MBSS and MKSS. MBSS, the better performing of the two, demonstrated more sensitivity than MRSC_MGD when there are no big background events occurring (i.e., no big elevation in the data streams other than outbreak-relevant data streams). One of the possible reasons, as I explained above, might be due to the incorporation of the non-disease-relevant data streams in the analysis. If there are some elevations occurring in the non-disease relevant data streams during the same time period as an outbreak, MRSC cannot detect the outbreak because of the nature of the MGD model. On the other hand, MBSS cannot perform as well as MRSC_MGD on data sets including simultaneous signals in all data streams which may be due to the events other than outbreaks. Since there is a trade-off in applying either algorithm, one suggestion for practice can be that we apply both algorithms to achieve both better detection sensitivity and better specificity.

In Chapter 6.4.3 I touched on the problem of detecting outbreaks having time-lag effects on different data streams by simply using an MGD model with varied outbreak effect values. The results showed improvement of detection. However, this method should only be applied when one has a prior knowledge about which data stream(s) have lagged effects from outbreaks. Another possible way to tackle time-lag effects, which can be studied in the future, is to introduce a variable called a lag window ϖ, 0$\leq \varpi \leq W, W \geq 1$. This process is similar to what we do to compute cross-correlation between two time series sequences. We can fix a time series of one data type, then slide the other time series forward by ϖ days. Recall in Chapter 6.1 where I compute the likelihood for a region Z of the data in Equation (6.3). We can re-write the equation to specify the day T we are analyzing as:

$$P\left(\boldsymbol{D_{Z,T}} \middle| \boldsymbol{\alpha_Z}, \boldsymbol{\beta_Z}\right) = \varphi \prod_{k=1}^{K-1} B(\alpha_k + C_{Z,T,k}, \beta_k + \sum\nolimits_{j=k+1}^{K} C_{Z,T,j}). \tag{6.16}$$

For a surveillance of two data streams D_2 and D_3 (in addition to the data stream “Others”, D_1) with a time lag ϖ existing in the data stream D_3, the likelihood becomes:

$$\begin{aligned} P\left(\boldsymbol{D_{Z,T}} \middle| \boldsymbol{\alpha_Z}, \boldsymbol{\beta_Z}\right)|_{K=3,\varpi} \\ &= \varphi \Big(B\big(\alpha_1 + C_{Z,T,1}, \beta_1 + C_{Z,T,2} + C_{Z,T-\varpi,3}\big) \\ &\quad + B\big(\alpha_2 + C_{Z,T,2}, \beta_2 + C_{Z,T-\varpi,3}\big)\Big). \end{aligned} \tag{6.17}$$

As I mentioned earlier in Chapter 6.4, the MGD model has a requirement for the order of the input data types. It models the elevations in the data streams in column $k, k \geq 2$ and the decrease in the data stream in the first column. Thus, in the research domain of disease outbreak detection, I put the “Others” data stream into the first column and put the disease-relevant data streams into the following columns without considering the sequence. However, it is worth noting that if we model the disease-relevant data streams in different sequences it will result in different likelihood values computed using Equation (6.3), which is written in Equation (6.18) with the derivation removed. Also recall how we estimate the hyper-parameters of MGD in Equation (6.19) and Equation (6.20), α_k and β_k for data stream D_k are computed based on the sum of $\mu_j, j = 1, .., k - 1$, which are the expected values of the weights of the previous data streams $D_1, \dots, D_{k-1}$, respectively.

$$P(\boldsymbol{D_Z}|\boldsymbol{\alpha_Z}, \boldsymbol{\beta_Z}) = \varphi \prod_{k=1}^{K-1} B(\alpha_k + C_{Z,k}, \beta_k + \sum\nolimits_{j=k+1}^{K} C_{Z,j}) \tag{6.18}$$

$$\alpha_k = \frac{\mu_k(\varphi_k - \rho_k)}{\rho_k\left(1 - \sum_{j=1}^{k-1} \mu_j\right) - \varphi_k \mu_k} \tag{6.19}$$

$$\beta_k = \frac{(1 - \sum_{j=1}^{k} \mu_j)(\varphi_k - \rho_k)}{\rho_k\left(1 - \sum_{j=1}^{k-1} \mu_j\right) - \varphi_k \mu_k} \tag{6.20}$$

In this chapter, I have discussed using the MGD model to detect diseases for which we already have prior knowledge about what data streams are likely to be affected and computing the probability of the elevations existing in these data streams simultaneously. However, it is also possible that we may want to detect some unknown disease outbreaks for which we do not have prior knowledge of what symptoms will be present for the infected population. One way to do that is to enumerate different combinations of data streams and model the elevations for each one. For example, M denotes the maximum number of data streams we can analyze using the MGD model, excluding the "Others" data stream; M_{total} is the number of total available data streams that can be incorporated in the analysis and $M_{total} > M$. The number of the combinations including m data streams except for the “Others” data stream is $\binom{M_{total}}{m}$, and the total number of the combinations will then be $\sum_{m=1}^{M}\binom{M_{total}}{m}$. Thus, we will need to run the MRSC_MGD algorithm for $\sum_{m=1}^{M}\binom{M_{total}}{m}$ times and each time analyze one combination of the data streams. However, partial prior knowledge (e.g., we know one or two symptoms of the unknown disease) will reduce the time of running the algorithm repeatedly. In literature, some research work has also been done to detect unknown diseases by using Bayesian modeling [90].

7.0 Conclusions and future work

This dissertation investigates a framework of rank-based tempo-spatial clustering (RSC) algorithms for early disease outbreak detection. It introduces a heuristic searching approach and Bayesian models for analyzing either univariate or multivariate data. In particular, this new searching approach utilizes the risk of having an outbreak for each spatial unit as heuristic information to find a cluster of areas considered all together to have a great probability of having an outbreak. In the evaluation, I demonstrated that RSC consistently outperformed other algorithms in terms of the timeliness in outbreak detection while having comparable detection powers. I conclude that RSC is a preferred algorithm for rapid and early detection of an outbreak.

This dissertation also proposes a Multinomial-generalized-Dirichlet (MGD) model which can be used in RSC for multivariate analysis (MRSC_MGD). Different from other existing multivariate algorithms [40,43], MGD models outbreak signals based on both disease-relevant data streams and non-disease-relevant data streams. The evaluation shows that MRSC_MGD has overall better performance than the univariate detectors and is more robust on the signals caused by non-outbreak events than two other multivariate algorithms. I conclude that MRSC_MGD can be a good extended algorithm to achieve both better sensitivity and better specificity for outbreak detection especially when there are non-outbreak-related events such as festivals or super bowl games.

The remainder of this chapter first summarizes the contributions of this dissertation research and then presents areas for future research.

7.1 Contributions

7.1.1 A rank-based tempo-spatial clustering algorithm

I proposed a rank-based tempo-spatial clustering algorithm for early disease outbreak detection. The main contribution in this part of the research work includes:

1) Proposed two measures to estimate the risk of having an outbreak for each spatial unit. One is called standard score (or z-score), which is computed as the number of standard deviations of the observed count varying from the expected count. This value is predicted by analyzing a time series of previous data; the other is posterior probability using Bayesian inference. These two measurements are later used as heuristic information for cluster searching;
2) Proposed a greedy searching mechanism for outbreak clusters. The searching starts from the highest ranked area (i.e., having highest risk), and iteratively adds the next ranked area into analysis. The adjacent areas are merged into a cluster. This searching mechanism is approximate but efficient;
3) Proposed a rank-based tempo-spatial clustering algorithm, RSC, utilizing the proposed greedy searching and Gamma-Poisson model for disease outbreak

detection with improved detection timeliness, cluster positive prediction value (PPV) and running time;

4) Investigated the performance of grid-based RSC (GRSC) (i.e., using grid cells as spatial units rather than exact spatial shapes) compared with that of Bayesian scan statistic algorithm (BSS), which also uses grid structures for searching.

7.1.2 A multivariate extension of RSC (MRSC)

In disease surveillance, there is an increasing trend to use more than one data source for surveillance. The meaningful use act, for example, incentivizes hospitals or health care providers to report additional data to public health agencies. The originally proposed RSC algorithm is only used in univariate analysis, therefore I also proposed a multivariate extension of RSC (MRSC) as the second part of my dissertation work. The main contributions of this part of the work include:

1) Proposed Multinomial-generalized-Dirichlet (MGD) model to capture the elevated signals only present in multiple disease-relevant data streams but not in non-disease-relevant data streams. In this model, I assume the counts in the monitored data streams follow a hierarchical Multinomial-Generalized-Dirichlet distribution. The reasons include: 1) it is natural to use multinomial distribution to model categorical data and 2) the generalized Dirichlet distribution is the conjugate prior of both multinomial and categorical distributions. This model not only takes the disease-relevant data streams into

account but also the non-disease-relevant data streams. In the evaluation, it is proven to be more robust to those signals caused by non-outbreak events.

2) Developed and evaluated MRSC, which applies the Multinomial-generalized-Dirichlet (MGD) model. This model was utilized in the cluster searching algorithm on multiple data streams. The evaluation demonstrated the advantage of the MGD model with its ability to effectively suppress false alarms caused by elevated signals that were not disease related but occured in all the analyzed data streams. I also investigated the performance of MRSC to better detect outbreaks when multiple data streams contain outbreak signals of different strengths by adjusting outbreak effect parameters. In addition, I attempted to detect the outbreaks with time lag effects on different data streams using MRSC_MGD with adjustable outbreak effect parameters.

7.2 Future work

This section describes future work and some open problems related to the dissertation research.

Recall that during the cluster searching in the RSC algorithm, clusters are merged if the shortest distance between them is less than or equal to an adjacency threshold, η, where η is set to be 0 in the study. By setting $\eta = 0$, only clusters with connected spatial areas are considered in the algorithm. However, it is often not true that an infectious disease outbreak only affects connected spatial areas. For example, some non-residential landforms, such as rivers, canyons, etc., may geographically separate the populations.

One way to overcome this limit is to use a grid cell as a unit study area and search for clusters with adjacent grid cells, where the geographical areas (e.g., ZIP code areas) inside two or more connected grid cells can be clustered together when they are not connected. I conducted such a study, reported in Chapter 5.2. However, if we still want to use each specific spatial shape as a unit, we can adjust the threshold distance to be inversely proportional to some density measure of a baseline factor (*e.g.,* population). alternatively, the non-residential landforms (*e.g.,* lakes, valleys, etc.) may be attached to their nearest census tract and may be considered in the analysis as well. Therefore, an investigation of the algorithm performance when $\eta > 0$ can be a supplement to this dissertation work. Another example where outbreak areas may be more dispersed is when people who are infected with the disease commute; commuting is a common occurrence nowadays. In Chapter 4.1.3, I considered people's commuting between their places of residence and places of work and created a data simulation model to allocate the counts of OTC sales into patients' residential ZIP code areas since the existing data set collected by NRDM only include the locations of pharmacies. There are also studies on commute models in literature [91]. However, further study is necessary in order to explicitly model this problem and use it for disease outbreak detection.

Another limitation of the study is it uses a simplified outbreak simulator for the evaluation of RSC performance. Despite the several advantages of the outbreak simulation model used in this study mentioned in the Chapter 4.2, the outbreak curve was, nonetheless, artificially constructed and therefore may not represent the complexity of real outbreaks. It would be more challenging to use sophisticated and complicated outbreak simulators because they not only model the disease specific features but also consider the stochastic effects to some extent [64][65]. Multivariate simulators should also be used as they consider real-life scenarios in which signals of an outbreak are present in one or more data types [55]. The utilization of these simulators for algorithm evaluation, therefore, will become another substantial study as one of the extensions of this work.

Recall that in Chapter 3.2, spatial or tempo-spatial disease outbreak detection algorithms other than KSS and BSS which can detect irregularly shaped outbreaks are discussed, including the fast subset scan algorithm (FSS), the flexible spatial scan statistic (FleXScan), the upper level set scan statistic (ULS) and the PANDA-CDCA-Temporal-Spatial (PCTS) algorithm. A comprehensive comparison among these algorithms and RSC would also a desired study as it could provide a bigger picture in this field and help decision makers to carefully select a suitable algorithm to use in practice with certain requirements.

In Experiment I and II, I tested the algorithm performance with prior knowledge of disease types (i.e., the disease-relevant data streams are known). We also want to tackle the problem when we do not have this kind of prior knowledge. One of the solutions is to exhaustively iterate through all the combinations of the data streams. But the running time becomes exponential to the number of available data streams.

As I mentioned in Chapter 6.6, I only touched on the problem of detecting outbreaks having time-lag effects on different data streams by simply using MGD model with varied outbreak effect values. The results showed improvement of detection. However, another possible way to tackle this problem which can be studied in the future is to model the time lag explicitly by introducing a variable called lag window ϖ,

$0 \leq \varpi \leq W, W \geq 1$. The idea is similar to what we do to compute cross-correlation between two time series sequences. We can fix a time series of one data type; then slide the other time series forward by ϖ days. The likelihood of each cluster having an outbreak inside it should be updated to incorporate this variable as well. An evaluation of this model would be valuable as a future study.

Appendix A. Parameter settings of the univariate analysis using the purely spatial model to run SaTScan v8.0

```
 [Input]
; case data filename
CaseFile=experiment_directroy\cas\4_0.2\0\1.cas
; control data filename
ControlFile=
; population data filename
PopulationFile= experiment_directroy\pop\pa_adjusted.pop
; coordinate data filename
CoordinatesFile= experiment_directroy\coordinate\pa.coo
; use grid file? (y/n)
UseGridFile=n
; grid data filename
GridFile=
; time precision (0=None, 1=Year, 2=Month, 3=Day)
PrecisionCaseTimes=0
; coordinate type (0=Cartesian, 1=latitude/longitude)
CoordinatesType=1
; study period start date (YYYY/MM/DD)
StartDate=2008/01/01
; study period end date (YYYY/MM/DD)
EndDate=2009/01/01

[Analysis]
; analysis type (1=Purely Spatial, 2=Purely Temporal, 3=Retrospective Space-Time, 4=Prospective Space-
                        Time, 5=N/A, 6=Prospective Purely Temporal)
AnalysisType=1
; model type (0=Discrete Poisson, 1=Bernoulli, 2=Space-Time Permutation, 3=Ordinal, 4=Exponential,
                        5=Normal, 6=Continuous Poisson, 7=Multinomial)
ModelType=0
; scan areas (1=High Rates(Poison,Bernoulli,STP); High Values(Ordinal,Normal); Short
                        Survival(Exponential), 2=Low Rates(Poison,Bernoulli,STP); Low
                        Values(Ordinal,Normal); Long Survival(Exponential), 3=Both Areas)
ScanAreas=1
; time aggregation units (0=None, 1=Year, 2=Month, 3=Day)
TimeAggregationUnits=0
; time aggregation length (Positive Integer)
TimeAggregationLength=1
; Monte Carlo replications (0, 9, 999, n999)
MonteCarloReps=999

[Output]
; analysis results output filename
ResultsFile=experiment_directory\result\4_0.2\0\1.out.txt
; output simulated log likelihoods ratios in ASCII format? (y/n)
SaveSimLLRsASCII=n
; output simulated log likelihoods ratios in dBase format? (y/n)
SaveSimLLRsDBase=n
; output relative risks in ASCII format? (y/n)
IncludeRelativeRisksCensusAreasASCII=n
; output relative risks in dBase format? (y/n)
IncludeRelativeRisksCensusAreasDBase=n
; output location information in ASCII format? (y/n)
```

```
CensusAreasReportedClustersASCII=y
; output location information in dBase format? (y/n)
CensusAreasReportedClustersDBase=n
; output cluster information in ASCII format? (y/n)
MostLikelyClusterEachCentroidASCII=n
; output cluster information in dBase format? (y/n)
MostLikelyClusterEachCentroidDBase=n
; output cluster case information in ASCII format? (y/n)
MostLikelyClusterCaseInfoEachCentroidASCII=n
; output cluster case information in dBase format? (y/n)
MostLikelyClusterCaseInfoEachCentroidDBase=n

[Multiple Data Sets]
; multiple data sets purpose type (0=Multivariate, 1=Adjustment)
MultipleDataSetsPurposeType=0

[Data Checking]
; study period data check (0=Strict Bounds, 1=Relaxed Bounds)
StudyPeriodCheckType=0
; geographical coordinates data check (0=Strict Coordinates, 1=Relaxed Coordinates)
GeographicalCoordinatesCheckType=0

[Spatial Neighbors]
; neighbors file
NeighborsFilename=
; use neighbors file (y/n)
UseNeighborsFile=n
; meta locations file
MetaLocationsFilename=
; use meta locations file (y/n)
UseMetaLocationsFile=n
; multiple coordinates type (0=OnePerLocation, 1=AtLeastOneLocation, 2=AllLocations)
MultipleCoordinatesType=0

[Spatial Window]
; maximum spatial size in population at risk (<=50%)
MaxSpatialSizeInPopulationAtRisk=50
; maximum spatial size in max circle population file (<=50%)
MaxSpatialSizeInMaxCirclePopulationFile=50
; maximum spatial size in distance from center (positive integer)
MaxSpatialSizeInDistanceFromCenter=1
; restrict maximum spatial size - max circle file? (y/n)
UseMaxCirclePopulationFileOption=n
; restrict maximum spatial size - distance? (y/n)
UseDistanceFromCenterOption=n
; include purely temporal clusters? (y/n)
IncludePurelyTemporal=n
; maximum circle size filename
MaxCirclePopulationFile=
; window shape (0=Circular, 1=Elliptic)
SpatialWindowShapeType=0
; elliptic non-compactness penalty (0=NoPenalty, 1=MediumPenalty, 2=StrongPenalty)
NonCompactnessPenalty=1
; isotonic scan (0=Standard, 1=Monotone)
IsotonicScan=0
```

```
[Temporal Window]
; maximum temporal cluster size (<=90%)
MaxTemporalSize=50
; include purely spatial clusters? (y/n)
IncludePurelySpatial=n
; how max temporal size should be interpretted (0=Percentage, 1=Time)
MaxTemporalSizeInterpretation=0
; temporal clusters evaluated (0=All, 1=Alive, 2=Flexible Window)
IncludeClusters=0
; flexible temporal window start range (YYYY/MM/DD,YYYY/MM/DD)
IntervalStartRange=2000/1/1,2000/12/31
; flexible temporal window end range (YYYY/MM/DD,YYYY/MM/DD)
IntervalEndRange=2000/1/1,2000/12/31

[Space and Time Adjustments]
; time trend adjustment type (0=None, 1=Nonparametric, 2=LogLinearPercentage,
                    3=CalculatedLogLinearPercentage, 4=TimeStratifiedRandomization)
TimeTrendAdjustmentType=0
; time trend adjustment percentage (>-100)
TimeTrendPercentage=0
; adjustments by known relative risks file name (with HA Randomization=1)
AdjustmentsByKnownRelativeRisksFilename=
; use adjustments by known relative risks file? (y/n)
UseAdjustmentsByRRFile=n
; spatial adjustments type (0=No Spatial Adjustment, 1=Spatially Stratified Randomization)
SpatialAdjustmentType=0

[Inference]
; prospective surveillance start date (YYYY/MM/DD)
ProspectiveStartDate=
; terminate simulations early for large p-values? (y/n)
EarlySimulationTermination=n
; adjust for earlier analyses(prospective analyses only)? (y/n)
AdjustForEarlierAnalyses=n
; perform iterative scans? (y/n)
IterativeScan=n
; maximum iterations for iterative scan (0-32000)
IterativeScanMaxIterations=0
; max p-value for iterative scan before cutoff (0.000-1.000)
IterativeScanMaxPValue=0.000

[Clusters Reported]
; criteria for reporting secondary clusters(0=NoGeoOverlap, 1=NoCentersInOther,
                    2=NoCentersInMostLikely,  3=NoCentersInLessLikely,
                    4=NoPairsCentersEachOther, 5=NoRestrictions)
CriteriaForReportingSecondaryClusters=0
; restrict reported clusters to maximum geographical cluster size? (y/n)
UseReportOnlySmallerClusters=y
; maximum reported spatial size in population at risk (<=50%)
MaxSpatialSizeInPopulationAtRisk_Reported=50
; maximum reported spatial size in max circle population file (<=50%)
MaxSizeInMaxCirclePopulationFile_Reported=50
; maximum reported spatial size in distance from center {positive integer)
MaxSpatialSizeInDistanceFromCenter_Reported=1
; restrict maximum reported spatial size - max circle file? (y/n)
UseMaxCirclePopulationFileOption_Reported=n
```

```
; restrict maximum reported spatial size - distance? (y/n)
UseDistanceFromCenterOption_Reported=n

[Additional Output]
; report critical values for .01 and .05? (y/n)
CriticalValue=n

[Elliptic Scan]
; elliptic shapes - one value for each ellipse (comma separated decimal values)
EllipseShapes=1.5,2,3,4,5
; elliptic angles - one value for each ellipse (comma separated integer values)
EllipseAngles=4,6,9,12,15

[Power Simulations]
; p-values for 2 pre-specified log likelihood ratios? (y/n)
PValues2PrespecifiedLLRs=n
; power calculation log likelihood ratio (no. 1)
LLR1=0
; power calculation log likelihood ratio (no. 2)
LLR2=0
; simulation methods (0=Null Randomization, 1=HA Randomization, 2=File Import)
SimulatedDataMethodType=0
; simulation data input file name (with File Import=2)
SimulatedDataInputFilename=
; print simulation data to file? (y/n)
PrintSimulatedDataToFile=n
; simulation data output filename
SimulatedDataOutputFilename=

[Run Options]
; analysis execution method  (0=Automatic, 1=Successively, 2=Centrically)
ExecutionType=0
; number of parallel processes to execute (0=All Processors, x=At Most X Processors)
NumberParallelProcesses=1
; log analysis run to history file? (y/n)
LogRunToHistoryFile=y
; suppressing warnings? (y/n)
SuppressWarnings=n

[System]
; system setting - do not modify
Version=8.1.1
```

Appendix B. Outbreak simulation parameter settings in the experiments

Table B.1 The parameter settings for outbreak simulations in the experiments

Chapter No.	Simulator	Parameter values	Comments
5.1.7	Linear simulator $O(t, \delta, z) = \delta \cdot t \cdot \mu_z$ For $1 \leq t \leq D$	$\delta \in \{0.2, 0.3\}$	Outbreak strength
		$D = 7$	Outbreak duration
		$K \in \{4, 8, 12\}$	Outbreak size
5.2.1	Linear simulator $O(t, \delta, z) = \delta \cdot t \cdot \mu_z$ For $1 \leq t \leq D$	$\delta \in \{0.2, 0.3\}$	Outbreak strength
		$D = 7$	Outbreak duration
		$K \in \{4, 8\}$	Outbreak size
6.5.2	Multivariate spatial-temporal event simulator (outbreaks)	AF,CO,TH	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 1.0, 0, 0, 0, 0)	Behavior probability vector
		(1.0, 1.0, 1.0)	Coverage vector
		Linear Poisson	Temporal function
		Flat/Uniform	Spatial function
6.5.3	Multivariate spatial-temporal event simulator (outbreaks)	DR,GI	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 0)	Behavior probability vector
		(1.0, 1.0)	Coverage vector
		Linear Poisson	Temporal function
		Flat/Uniform	Spatial function
6.5.4	Multivariate spatial-temporal event simulator (outbreaks)	AF,CO,TH	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 1.0, 0,0,0,0)	Behavior probability vector
		(1.0, 1.0,1.0)	Coverage vector
		Linear Poisson	Temporal function
		Flat/Uniform	Spatial function

6.5.4	Multivariate spatial-temporal event simulator (baseline shifts)	AF,CO,TH	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta = 1$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 1.0, 0,0,0,0)	Behavior probability vector
		(1.0, 1.0,1.0)	Coverage vector
		Flat/Uniform	Temporal function
		Flat/Uniform	Spatial function
6.5.5	Multivariate spatial-temporal event simulator (outbreaks)	AF,CO,TH	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 1.0, 0,0,0,0)	Behavior probability vector
		(1.0, 1.0,1.0) (1.0, 0.5,1.0) (1.0,1.0,0.5) (0.5, 0.5,1.0) (0.5, 1.0,0.5) (1.0, 0.5, 0.5)	Coverage vector
		Linear Poisson	Temporal function
		Flat/Uniform	Spatial function
6.5.6	Multivariate spatial-temporal event simulator (outbreaks)	AF,CO,TH	Outbreak data streams
		$O = \delta \cdot T \cdot \mu_z$	Total cases
		$\delta \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$	Outbreak strength
		$K = 8$	Outbreak size
		(1.0, 1.0, 1.0, 0,0,0,0)	Behavior probability vector
		(1.0, 1.0,1.0)	Coverage vector
		(2, 0, 0) (0, 2, 0) (0, 0, 2)	Time lag (days)
		Linear Poisson	Temporal function
		Flat/Uniform	Spatial function

Appendix C. Parameter settings of multivariate analysis using the purely spatial model to run SaTScan v9.1

```
[Input]
; case data filename
CaseFile=/experiment_directroy/casefile_1.cas
; control data filename
ControlFile=
; population data filename
PopulationFile=/experiment_directroy/populationfile_1.pop
; coordinate data filename
CoordinatesFile=/experiment_directroy/coordinatesfile.geo
; use grid file? (y/n)
UseGridFile=n
; grid data filename
GridFile=
; time precision (0=None, 1=Year, 2=Month, 3=Day, 4=Generic)
PrecisionCaseTimes=0
; coordinate type (0=Cartesian, 1=latitude/longitude)
CoordinatesType=1
; study period start date (YYYY/MM/DD)
StartDate=2008/01/01
; study period end date (YYYY/MM/DD)
EndDate=2009/01/01
[Analysis]
; analysis type (1=Purely Spatial, 2=Purely Temporal, 3=Retrospective Space-Time, 4=Prospective Space-
Time, 5=Spatial Variation in Temporal Trends, 6=Prospective Purely Temporal)
AnalysisType=1
; model type (0=Discrete Poisson, 1=Bernoulli, 2=Space-Time Permutation, 3=Ordinal, 4=Exponential,
5=Normal, 6=Continuous Poisson, 7=Multinomial)
ModelType=0
; scan areas (1=High Rates(Poison,Bernoulli,STP); High Values(Ordinal,Normal); Short
Survival(Exponential), 2=Low Rates(Poison,Bernoulli,STP); Low Values(Ordinal,Normal); Long
Survival(Exponential), 3=Both Areas)
ScanAreas=1
; time aggregation units (0=None, 1=Year, 2=Month, 3=Day, 4=Generic)
TimeAggregationUnits=0
; time aggregation length (Positive Integer)
TimeAggregationLength=1
[Output]
; analysis results output filename
ResultsFile=/oradata02/jque_data/mSaTScan/allegheny_adjusted_standardMC/results/0_1.out
; output simulated log likelihoods ratios in ASCII format? (y/n)
SaveSimLLRsASCII=n
; output simulated log likelihoods ratios in dBase format? (y/n)
SaveSimLLRsDBase=n
; output relative risks in ASCII format? (y/n)
IncludeRelativeRisksCensusAreasASCII=y
; output relative risks in dBase format? (y/n)
IncludeRelativeRisksCensusAreasDBase=n
; output location information in ASCII format? (y/n)
CensusAreasReportedClustersASCII=y
; output location information in dBase format? (y/n)
CensusAreasReportedClustersDBase=n
; output cluster information in ASCII format? (y/n)
MostLikelyClusterEachCentroidASCII=y
```

```
; output cluster information in dBase format? (y/n)
MostLikelyClusterEachCentroidDBase=n
; output cluster case information in ASCII format? (y/n)
MostLikelyClusterCaseInfoEachCentroidASCII=n
; output cluster case information in dBase format? (y/n)
MostLikelyClusterCaseInfoEachCentroidDBase=n
[Multiple Data Sets]
; multiple data sets purpose type (0=Multivariate, 1=Adjustment)
MultipleDataSetsPurposeType=0
; case data filename (additional data set 2)
CaseFile2=/experiment_directroy/casefile_2.cas
; case data filename (additional data set 3)
CaseFile3=/experiment_directroy/casefile_3.cas
; control data filename (additional data set 2)
ControlFile2=
; control data filename (additional data set 3)
ControlFile3=
; population data filename (additional data set 2)
PopulationFile2=/experiment_directroy/populationfile_2.pop
; population data filename (additional data set 3)
PopulationFile3=/experiment_directroy/populationfile_3.pop
[Data Checking]
; study period data check (0=Strict Bounds, 1=Relaxed Bounds)
StudyPeriodCheckType=0
; geographical coordinates data check (0=Strict Coordinates, 1=Relaxed Coordinates)
GeographicalCoordinatesCheckType=0
[Spatial Neighbors]
; neighbors file
NeighborsFilename=
; use neighbors file (y/n)
UseNeighborsFile=n
; meta locations file
MetaLocationsFilename=
; use meta locations file (y/n)
UseMetaLocationsFile=n
; multiple coordinates type (0=OnePerLocation, 1=AtLeastOneLocation, 2=AllLocations)
MultipleCoordinatesType=0
[Spatial Window]
; maximum spatial size in population at risk (<=50%)
MaxSpatialSizeInPopulationAtRisk=50
; maximum spatial size in max circle population file (<=50%)
MaxSpatialSizeInMaxCirclePopulationFile=50
; maximum spatial size in distance from center (positive integer)
MaxSpatialSizeInDistanceFromCenter=1
; restrict maximum spatial size - max circle file? (y/n)
UseMaxCirclePopulationFileOption=n
; restrict maximum spatial size - distance? (y/n)
UseDistanceFromCenterOption=n
; include purely temporal clusters? (y/n)
IncludePurelyTemporal=n
; maximum circle size filename
MaxCirclePopulationFile=
; window shape (0=Circular, 1=Elliptic)
SpatialWindowShapeType=0
; elliptic non-compactness penalty (0=NoPenalty, 1=MediumPenalty, 2=StrongPenalty)
NonCompactnessPenalty=0
```

```
; isotonic scan (0=Standard, 1=Monotone)
IsotonicScan=0
[Temporal Window]
; maximum temporal cluster size (<=90%)
MaxTemporalSize=50
; include purely spatial clusters? (y/n)
IncludePurelySpatial=n
; how max temporal size should be interpretted (0=Percentage, 1=Time)
MaxTemporalSizeInterpretation=0
; temporal clusters evaluated (0=All, 1=Alive, 2=Flexible Window)
IncludeClusters=0
; flexible temporal window start range (YYYY/MM/DD,YYYY/MM/DD)
IntervalStartRange=2000/1/1,2000/12/31
; flexible temporal window end range (YYYY/MM/DD,YYYY/MM/DD)
IntervalEndRange=2000/1/1,2000/12/31
[Space and Time Adjustments]
; time trend adjustment type (0=None, 1=Nonparametric, 2=LogLinearPercentage,
3=CalculatedLogLinearPercentage, 4=TimeStratifiedRandomization, 5=CalculatedQuadraticPercentage)
TimeTrendAdjustmentType=0
; time trend adjustment percentage (>-100)
TimeTrendPercentage=0
; adjustments by known relative risks file name (with HA Randomization=1)
AdjustmentsByKnownRelativeRisksFilename=
; use adjustments by known relative risks file? (y/n)
UseAdjustmentsByRRFile=n
; spatial adjustments type (0=No Spatial Adjustment, 1=Spatially Stratified Randomization)
SpatialAdjustmentType=0
; time trend type - SVTT only (Linear=0, Quadratic=1)
TimeTrendType=0
[Inference]
; prospective surveillance start date (YYYY/MM/DD)
ProspectiveStartDate=2000/12/31
; p-value reporting type (Default p-value=0, Standard Monte Carlo=1, Early Termination=2, Gumbel p-
value=3)
PValueReportType=1
; report Gumbel p-values
ReportGumbel=n
; early termination threshold
EarlyTerminationThreshold=50
; adjust for earlier analyses(prospective analyses only)? (y/n)
AdjustForEarlierAnalyses=n
; perform iterative scans? (y/n)
IterativeScan=n
; maximum iterations for iterative scan (0-32000)
IterativeScanMaxIterations=10
; max p-value for iterative scan before cutoff (0.000-1.000)
IterativeScanMaxPValue=0.05
; Monte Carlo replications (0, 9, 999, n999)
MonteCarloReps=999
[Clusters Reported]
; criteria for reporting secondary clusters(0=NoGeoOverlap, 1=NoCentersInOther,
2=NoCentersInMostLikely,  3=NoCentersInLessLikely, 4=NoPairsCentersEachOther, 5=NoRestrictions)
CriteriaForReportingSecondaryClusters=0
; restrict reported clusters to maximum geographical cluster size? (y/n)
UseReportOnlySmallerClusters=n
; maximum reported spatial size in population at risk (<=50%)
```

```
MaxSpatialSizeInPopulationAtRisk_Reported=50
; maximum reported spatial size in max circle population file (<=50%)
MaxSizeInMaxCirclePopulationFile_Reported=50
; maximum reported spatial size in distance from center {positive integer)
MaxSpatialSizeInDistanceFromCenter_Reported=1
; restrict maximum reported spatial size - max circle file? (y/n)
UseMaxCirclePopulationFileOption_Reported=n
; restrict maximum reported spatial size - distance? (y/n)
UseDistanceFromCenterOption_Reported=n
[Additional Output]
; report critical values for .01 and .05? (y/n)
CriticalValue=n
; report cluster rank (y/n)
ReportClusterRank=n
; print ascii headers in output files (y/n)
PrintAsciiColumnHeaders=n
[Elliptic Scan]
; elliptic shapes - one value for each ellipse (comma separated decimal values)
EllipseShapes=1.5,2,3,4,5
; elliptic angles - one value for each ellipse (comma separated integer values)
EllipseAngles=4,6,9,12,15
[Power Simulations]
; p-values for 2 pre-specified log likelihood ratios? (y/n)
PValues2PrespecifiedLLRs=n
; power calculation log likelihood ratio (no. 1)
LLR1=0
; power calculation log likelihood ratio (no. 2)
LLR2=0
; simulation methods (0=Null Randomization, 1=HA Randomization, 2=File Import)
SimulatedDataMethodType=0
; simulation data input file name (with File Import=2)
SimulatedDataInputFilename=
; print simulation data to file? (y/n)
PrintSimulatedDataToFile=n
; simulation data output filename
SimulatedDataOutputFilename=
[Run Options]
; analysis execution method  (0=Automatic, 1=Successively, 2=Centrically)
ExecutionType=0
; number of parallel processes to execute (0=All Processors, x=At Most X Processors)
NumberParallelProcesses=1
; log analysis run to history file? (y/n)
LogRunToHistoryFile=n
; suppressing warnings? (y/n)
SuppressWarnings=n
[System]
; system setting - do not modify
Version=9.1.1
```

Appendix D. Derivation of the outbreak effect parameters

Let vector $\boldsymbol{\pi} = (\pi_1, \pi_2, \dots, \pi_{K-1})$ represents the outbreak effect values for all the study data streams, and it is used in the model $MGD(\boldsymbol{\pi I_K \alpha_S}, \boldsymbol{\beta_S})$ to model the alternative hypothesis, $H_1(Z)$. Let e_k represent the expected values of multinomial distribution parameters ($E(p_k)$) before an outbreak occurs, and $\tilde{e}_k$ represent the expected values ($\tilde{E}(p_k)$) after the outbreak occurs, and let $\Delta_k = \frac{\tilde{e}_k - e_k}{e_k}$ $(k \geq 2)$ represent the increase percentage for data stream k. Recall that $e_k = \frac{\alpha_k}{\alpha_k + \beta_k} \prod_{m=1}^{k-1} \frac{\beta_m}{\alpha_m + \beta_m}$ is recursively dependent on $e_1, \dots, e_{k-1}$. Let $\tilde{e}_k = \frac{\pi_k \alpha_k}{\pi_k \alpha_k + \beta_k} \prod_{m=1}^{k-1} \frac{\beta_m}{\pi_k \alpha_m + \beta_m}$ and $\tilde{\vartheta}_k = \frac{\pi_k \alpha_k}{\pi_k \alpha_k + \beta_k}$. We can then rewrite them as the following.

$$\tilde{e}_k = \tilde{\vartheta}_k \prod_{m=1}^{k-1} (1 - \tilde{\vartheta}_m) \tag{D.1}$$

$$\pi_k = \frac{\beta_k}{\alpha_k} \frac{\tilde{\vartheta}_k}{1 - \tilde{\vartheta}_k} \tag{D.2}$$

In order to compute $\pi_k, (k \geq 2)$ in Equation (D.2), we need to compute $\tilde{\vartheta}_k$. We can rewrite Equation (D.1) again in the following.

$$\tilde{\vartheta}_k = \frac{\tilde{e}_k}{\prod_{m=1}^{k-1} (1 - \tilde{\vartheta}_m)} \tag{D.3}$$

Given the constraint of the multinomial distribution where $\sum_{k=1}^{K} E(p_k) = 1$ (i.e., $\sum_{k=1}^{K} e_k = 1$, and $\sum_{k=1}^{K} \tilde{e}_k = 1$ as well), the total change across all data streams should be 0, which is:

$$\sum_{k=1}^{K} \Delta_k e_k = 0 \tag{D.4}$$

Because the proportionality constants among Δ_k's ($k \geq 2$) are known, we can let $\Delta_k = a_k\sigma$ where a_k's ($k \geq 2$) are known. Using Equation (D.4), σ and Δ_k, ($k \geq 2$) can be computed as in the following because the value of π_1 (i.e., Δ_1) is known (Recall as in Chapter 6.4.1, I define $\chi_1 < 1$ and is equally likely to be the values in the set $\omega =${0.1, 0.2, 0.3, 0.4. 0.5, 0.6, 0.7, 0.8, 0.9})

$$\sigma = \frac{-\Delta_1 e_1}{\sum_{m=2}^{K} a_m e_m} \quad \text{(D.5)}$$

$$\Delta_k = \frac{-a_k \Delta_1 e_1}{\sum_{m=2}^{K} a_m e_m} \quad \text{(D.6)}$$

Recall $\Delta_k = \frac{\tilde{e}_k - e_k}{e_k}$ ($k \geq 2$), $\tilde{e}_k$, $k \geq 2$ can then be computed in the following using Equation (D.6).

$$\tilde{e}_k = (1 - \frac{a_k \Delta_1 e_1}{\sum_{m=2}^{K} a_m e_m}) e_k \quad \text{(D.7)}$$

Now we can solve π_k, ($k \geq 2$) using Equation (D.2) and (D.7).

References

1. Wagner MM, Tsui FC, Espino J, Hogan W, Hutman J, Hersh J, et al. National Retail Data Monitor for public health surveillance. Morbidity & Mortality Weekly Report. 2004; 53(Suppl): p. 40-42.
2. Tsui FC, Espino JU, Dato VM, Gesteland PH, Hutman J, Wagner MM. Technical description of RODS: a real-time public health surveillance system. J Am Med Inform Assoc. 2003; 10(5): p. 399-408.
3. Hunter JS. The Exponentially Weighted Moving Average. Journal of Quality Technology. 1986; 18: p. 155-162.
4. Hawkins DM, Olwell DH. Cumulative Sum Charts and Charting for Quality Improvement Heidelberg: Springer.
5. Wagner MM, Moore AW, Aryel RM. Handbook of biosurveillance: Academic Press; 2006.
6. Buckeridge DL, Burkom H, Campbell M, Hogan WR, Moore AW. Algorithms for rapid outbreak detection: a research synthesis. Journal of Biomedical Informatics. 2005; 38(2): p. 99-113.
7. Zhang J, Tsui FC, Wagner MM. Detection of outbreaks from time series data using wavelet transform. In AMIA Synposium; 2003. p. 410-414.
8. Kulldorff M. A spatial scan statistic. Commun Stat Theory Methods. 1997; 26(6): p. 1481-1496.
9. Huang L, Kulldorff M, Gregorio D. A spatial scan statistic for survival data. Biometrics. 2007; 63: p. 109-118.
10. Neill DB, Moore AW, Cooper GF. A Bayesian spatial scan statistic. Advances in Neural Information Processing Systems. 2005; 18: p. 1003-1010.
11. Takahashi K, Kulldorff M, Tango T, Yih K. A flexibly shaped space-time scan statistic for disease outbreak detection and monitoring. International Journal of Health Geographics. 2008; 7: p. 14.
12. Jain A, Murty M, Flynn P. Data clustering: a review. ACM, Computing Surveys. 1999; 31(3): p. 264-323.
13. Zeng D, Chang W, Chen H. A comparative study of spatio-temporal hotspot analysis techniques in security informatics. In IEEE Intelligent Transportation Systems Conference; 2004. p. 106-111.
14. Wong WK, Cooper GF, Dash DH, Levander JD, Dowling JN, Hogan WR, et al. Bayesian biosurveillance using multiple data streams. Morbidity and Mortality Weekly Report Supplement. 2005; 54: p. 63-69.
15. Mandl D, Overhage MJ, Wagner M, Lober B, Sebastiani P, Mostashari F, et al. Implementing syndromic surveillance: a practical guide informed by the early

experience. J Am Med Inform Assoc. 2004; 11(2): p. 141-150.

16. Rubin B, Gelman A, Carlin B, Stern H. Bayesian Data Analysis Boca Raton: Chapman & Hall/CRC; 2003.
17. Bernardo JM, Smith AF. Bayesian Theory: Wiley; 2000.
18. Johnson NL, Kotz S, Kemp AW. Univariate Discrete Distributions: Wiley; 1993.
19. Merran E, Hastings N, Peacock B. Statistical Distributions. 3rd ed. New York: Wiley; 2000.
20. Honkela. Nonlinear Switching State-Space Models. [Online]. Available from: http://www.cis.hut.fi/ahonkela/dippa/node95.html#fig:dirichletpdf.
21. Novick MR, Grizzle JE. A Bayesian approach to the analysis of data from clinical trials. J. Amer. Statist. Assoc. 1965; 60: p. 81-96.
22. Lochner RH, Basu AP. Bayesian analysis of time truncated samples. Technical Report. Madison: University of Wisconsin; 1985.
23. Ferguson TS. A Bayesian analysis of some non-parametric problems. Ann. Statist. 1973; 1: p. 209-230.
24. Good IJ. A Bayesian significance test for multinomial distributions. J. R. Statist. Soc. B. 1967; 29: p. 399-431.
25. Lindley DV. Bayesian statistics, a review. In Society for Industrial and Applied Mathematics; 1971; Philadelphia.
26. Lochner RH. A generalized Dirichlet distribution in Bayesian life testing. J. Royal Statist. Soc. 1975; 37: p. 103-113.
27. Wong TT. Generalized Dirichlet distribution in Bayesian analysis. Applied Mathematics and Computation 97. 1998;: p. 165-181.
28. Connor RJ, Mosimann JE. Concepts of independence for proportions with a generalization of the Dirichlet distribution. J. Amer. Statist. Assoc. 1969; 64: p. 194-206.
29. Takahashi K, Kulldorff M, Tango T, Yih K. A flexibly shaped space-time scan statistic for disease outbreak detection and monitoring. Int J Health Geogr. 2008 Apr; 7(14).
30. Patil GP, Taillie C. Upper level set scan statistic for detecting arbitrarily shaped hotspots. Environmental and Ecological Statistics. 2004; 11: p. 183-197.
31. Levine N. CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations. In Geocomputation 99; 1999; Fredericksburg, VA.
32. Hogan WR, Cooper GF, Wallstrom GL, Wagner MM, Depinay JM. The Bayesian aerosol release detector: An algorithm for detecting and characterizing outbreaks caused by an atmospheric release of Bacillus anthracis. Statistics in Medicine. 2007; 26(29): p. 5225–5252.
33. Que J, Tsui FC. A multi-level spatial clustering algorithm for detection of disease outbreaks. In AMIA Annual Symposium Proceedings; 2008; Washinton DC. p. 611-615.
34. Cooper GF, Dash DH, Levander JD, Wong W, Hogan WR, Wagner MM. Bayesian biosurveillance of disease outbreaks. In the 20th Annual Conference on Uncertainty

in Artificial Intelligence; 2004. p. 94-103.

35. Jiang X, Cooper GF. A Bayesian spatio-temporal method for disease outbreak detection. JAMIA. 2010; 17: p. 462-471.
36. Neill DB, Moore AW. Rapid detection of significant spatial clusters. In Proceedings of the 10th ACM SIGKDD Conference on Knowledge Discovery and Data Mining; 2004. p. 256-265.
37. Neill DB, Moore AM, Pereira F, Mitchell T. Detecting significant multidimensional spatial clusters. Advances in Neural Information Processing Systems. 2005; 17: p. 969-976.
38. Neill DB. Fast subset scan for spatial pattern detection. Journal of the Royal Statistical Society: Series B (Statistical Methodology). 2012; 74(2): p. 337-360.
39. Jiang X, Cooper GF. A recursive algorithm for spatial cluster detection. In AMIA Symposium Proceedings; 2007. p. 369-373.
40. Kulldorff M, Mostashari F, Duczmal L, Yih KW, Kleinman K, Platt R. Multivariate scan statistics for disease surveillance. Statistics in Medicine. 2007; 26(8): p. 1824-1833.
41. Burkom H, Coberly J, Murphy S, Elbert Y, Hurt-Mullen K. Public health monitoring tools for multiple data streams. In Proceedings of the National Syndromic Surveillance Conference; 2004; Boston, MA.
42. Edgington ES. A normal curve method for combinging probability values from independent experiments. Journal of Psychology. 1972;(82): p. 85-89.
43. Neill DB, Cooper GF. A multivariate Bayesian scan statistic for early event detection and characterization. Machine Learning. 2010; 29: p. 261-282.
44. Wong WK, Moore AW, Cooper GF, Wagner MM. Rule-based anomaly pattern detection for detecting disease outbreaks. In Proceedings of the 18th National conference on artificial Intelligence; 2002: MIT Press.
45. Wong WK, Moore AW, Cooper GF, Wagner MM. Bayesian network anomaly pattern detection for disease outbreaks. In the 20th Internation Conference on Machine Learning; 2003. p. 808-815.
46. Das K, Schneider J, Neill D. Anomaly Pattern Detection in Categorical Datasets. In The 14th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; 2008.
47. Burkom HS. Biosurveillance applying scan statistics with multiple, disparate data sources. Journal of Urban Health. 2003; 80(2 suppl. 1): p. i57-65.
48. Reis BY, Kohane IS, Mandl KD. An epidemiological network model for disease outbreak detection. PLoS Medicine. 2007; 4: p. 210.
49. Kulldorff M, Huang L, Pickle L, Duczmal L. An elliptic spatial scan statistic. Statistics in Medicine. 2006.
50. Neill DB, McFowland III E, Zheng H. Fast subset scan for multivariate spatial biosurveillance. Emerging Health Threats Journal. 2011; 4(s7): p. 37-38.
51. Que J, Tsui FC. Rank-based spatial clustering: an algorithm for rapid outbreak detection. J Am Med Inform Assoc. 2011; 18: p. 218-224.

52. Siegrist D, Pavlin JA. BioALIRT biosurveillance testbed evaluation. In Syndromic surveillance: reports from a national conference; New York, NY. p. 152-158.

53. Kulldorf M, Heffernan R, Hartman J, Assunção R, Mostashari F. A space–time permutation scan statistic for disease outbreak detection. PLoS Med. 2005; 2(3): p. e59.

54. Que J, Tsui FC. Evaluation of two spatial algorithms for detection of disease outbreaks in two states. In Modern Engineering and Technology Seminar; 2010; Taiwan.

55. Zhang M, Kong X, Wallstrom L. Simulation of multivariate spatial-temporal outbreak data for detection algorithm evaluation. Lecture Notes in Computer Science, BioSecure. 2008; 5354: p. 155-163.

56. Hogan WR, Tsui FC, Ivanov O, Gesteland PH, Grannis S, Overhage M, et al. Detection of Pediatric Respiratory and diarrheal Outbreaks from Sales of Over-the-counter Electrolyte Products. Journal of the American Medical Informatics Association. 2003; 10(6): p. 555-562.

57. Magruder SF. Evaluation of over-the-counter pharmaceutical sales as a possible early warning indicator of human disease. Johns Hopkins APL technical digest. 2003; 24(4): p. 349-353.

58. Welliver RC, Cherry JD, Boyer KM, Deseda-Tous JE, Krause PJ, Dudley JP, et al. Sales of Nonprescription Cold Remedies: A Unique Method of Influenza Surveillance. Pediatric Research. 1979; 13: p. 1015-1017.

59. Buckeridge DL. A method for evaluating outbreak detection in public health surveillance systems that use administrative data. Thesis. Standford University; 2005.

60. Watkins RE, Eagleson S, Beckett S, Garner G, Veenendaal B, Wright G, et al. Using GIS to create synthetic disease outbreaks. 2007; 7(4).

61. Mandl KD, Reis B, Cassa C. Measuring outbreak-detection performance by using controlled feature set simulations. MMWR Morb Mortal Wkly Rep. ; (Suppl 53): p. 130-136.

62. Neill DB, Moore AM, Sabhnani M, Daniel K. Detection of emerging space-time clusters. In Porc. 11th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining; 2005.

63. Buckeridge DL, Burkom H, Moore AM, Pavlin JA, Cutchis P, Hogan W. Evaluation of syndromic surveillance systems -- design of an epidemic simulation model. In Morbidity and Mortality Weekly Report (MMWR); 2004. p. 137-143.

64. Buckeridge DL, Owens DK, Switzer P, Frank J, Musen MA. Evaluating detection of an inhalational anthrax outbreak. Emerging Infectious Diseases. 2006; 12(12): p. 1942-1949.

65. Hogan WR, Cooper GF, Wagner MM, Wallstrom GL. An inverted Gaussian plume model for estimating the location and amount of release of airborne agents from downwind atmospheric concentrations. RODS Laboratory Technical Report. ; 2005.

66. Kulldorff M, Zhang Z, Hartman J, Heffernan R, Huang L, Mostashari F. Benchmark data and power calculations for evaluating disease outbreak detection methods. In MMWR Morb Mortal Wkly Rep; 2004. p. 144-151.

67. Que J, Tsui FC, Espino JU. A z-score based multi-level spatial clustering algorithm for the detection of disease outbreaks. Lecture Notes in Computer Science, BioSecure. 2008; 5453: p. 108-118.
68. Fawcett T. ROC graphs: notes and practical considerations for researchers. Technical report. Palo Alto, USA: HP Laboratories; 2004.
69. Fawcett T, Provost F. Activity monitoring: noticing interesting changes in behavior. In 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 1999. p. 53-62.
70. Forsberg L, Bonetti M, Jeffery C, Ozonoff A, Pagano M. Distance-based methods for spatial and spatio-temporal surveillance. 2nd ed.: Wiley; 2005.
71. DeLong ER, DeLong DM, Clarke-Pearson DL. Comparing the areas under two or more correlated receiver operating characteristic curves: A nonparametric approach. Biometrics. 1988; 44: p. 837-845.
72. Burrus SC, Gopinath A, Guo H. Introduction to wavelets and wavelet transforms: Prentice Hall; 1997.
73. Que J, Tsui FC. Evaluation of two spatial algorithms for detection of disease outbreaks in eight states. Unpublished manuscript, available upon request. 2012.
74. Wagner MM, Tsui FC, Pike J, Pike L. Design of a clinical notification system. In Proceedings of AMIA Annual Symposium; 1999. p. 989-993.
75. Wagner MM, Tsui FC, Espino JU, Dato VM, Sitting DF, Caruana RA, et al. The emerging science of very early detection of disease outbreaks. Journal of Public Health Management & Practice. 2001; 7(6): p. 51-59.
76. Tsui FC, Wagner MM, Dato VM, Chang CC. Value of ICD-9 coded chief complaints for detection of epidemics. In Proceedings of AMIA Annual Symposium; 2001. p. 711-715.
77. Park M. CNN Health. [Online].; 2009. Available from: http://articles.cnn.com/2009-05-02/health/worried.well.hospitals_1_h1n1-swine-flu-emergency-departments?_s=PM:HEALTH.
78. Spiegelhalter DJ, Harris NL, Bull K, Franklin RCG. Empirical-evaluation of prior beliefs about frequencies - Methodology and a case-study in congenital heart disease. J. Amer. Statist. Assoc. 1994; 89: p. 435-443.
79. Paulino CDM, Pereira CAD. Bayesian methods for categorical-data under informative general censoring. Biometrika 82. ;: p. 439-446.
80. Lange K. Applications of the Dirichlet distribution for forensic match probabilities. Genetica 96. 1995;: p. 107-117.
81. Mauldon JG. Random division of an interval. In Proceedings of the Cambridge Philosophical Society. p. 331-358.
82. Mauldon JG. A generalization of the Beta-distrbution. In Annals of Mathematical Statistics. p. 502-520.
83. Good TJ. The estimation of probabilities. In ; 1965; Cambridge, Mass: MIT Press.
84. Aslam S, Ajit I, Adams S, Faigen Z. Correlation between Over-The-Counter Thermometer Sales and Emergency Room Department Visits for Influenza like Illness. In APHA 138th Annual Meeting & Expo; 2010; Denver, CO.

85. Das D, Metzger K, Heffernan R, Balter S, Weiss D, Mostashari F. Monitoring over-the-counter medication sales for early detection of disease outbreaks -- New York City. MMWR. 2005 August 26: p. 41-46.

86. Vingilis E, Brown U, Hennen B. Common colds: reported patterns of self-care and health care use. Can Fam Physician. 1999; 45: p. 2644-6, 2649-52.

87. Metzger KB, Hajat A, Crawford M, Mostashari F. How many illnesses does one emergency department visit represent? using a population-based telephone survey to estimate the syndromic multiplier. MMWR. 2004 September 24: p. 106-11.

88. Wang B, Phillips M, Schrieber R, Wilkinson D, Mishra N, Tarjan R. Spatial scan statistics for graph clustering. In 8th SIAM Intenational Conference on Data Mining; 2008.

89. Jung I, Kulldorff M, Klassen AC. A spatial scan statistic for ordinal data. Stat Med. 2007; 26(7): p. 1594-1607.

90. Shen Y, Cooper GF. Bayesian modeling of unknown diseases for biosurveillance. In AMIA Annu Symp Proc; 2009. p. 589-593.

91. Cami A, Wallstrom GL, Hogan WR. Effect of commuting on the detection and characterization performance of the Bayesian Aerosol Release Detector. In Proceedings of the International Workshop on Biomedical and Health Informatics (BHI 2008); 2008. p. 91-98.

92. Jacquenot. An open exchange for the MATLAB and Simulink user community. [Online].; 2008. Available from: http://www.mathworks.com/matlabcentral/fileexchange/22444-minimum-distance-between-two-polygons&watching=22444.