Detection of Spatial and Spatio-Temporal Clusters

Daniel B. Neill

June 5, 2006

CMU-CS-06-142

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis committee:
Andrew Moore, Chair
Tom Mitchell
Jeff Schneider
Gregory Cooper (University of Pittsburgh)
Andrew Lawson (University of South Carolina)

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

Copyright ©2006 **Daniel B. Neill**

This research was supported by the National Science Foundation under grant number IIS-0325581. Additionally, the author is a recipient of an NSF Graduate Research Fellowship. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies, either expressed or implied, of the NSF or the U.S. government.

Keywords: cluster detection, data mining, algorithms, biosurveillance, fMRI

Abstract

This thesis develops a general and powerful statistical framework for the automatic detection of spatial and space-time clusters. Our “generalized spatial scan” framework is a flexible, model-based framework for accurate and computationally efficient cluster detection in diverse application domains. Through the development of the “fast spatial scan” algorithm and new Bayesian cluster detection methods, we can now detect clusters hundreds or thousands of times faster than previous approaches. More timely detection of emerging clusters (with high detection power and low false positive rates) was made possible by development of “expectation-based” scan statistics, which learn baseline models from past data then detect regions that are anomalous given these expectations. These cluster detection methods were applied to two real-world problem domains: the early detection of emerging disease epidemics, and the detection of clusters of activity in fMRI brain imaging data. One major contribution of this work is the development of the SSS system for nationwide disease surveillance, currently used in daily practice by several state and local health departments. This system receives data (including emergency department records and medication sales) from over 20,000 stores and hospitals nationwide, automatically detects emerging clusters of disease, and reports these results to public health officials. Through retrospective case studies and semi-synthetic testing, we have shown that our system can detect outbreaks significantly faster than previous disease surveillance methods.

Acknowledgements

First of all, I would like to thank my advisor, Andrew Moore. Andrew has contributed to this work in many ways, and has taught me a tremendous amount. It was his energy and enthusiasm that drew me to Carnegie Mellon, and led me down my current research path. Second, I would like to thank my committee members, Greg Cooper, Andrew Lawson, Tom Mitchell, and Jeff Schneider, for many helpful comments and stimulating discussions. It has been a pleasure working closely with Greg on the Bayesian Biosurveillance project, and with Tom on the analysis of brain imaging data. Third, I would like to thank my colleagues in the Auton Laboratory (Carnegie Mellon) and RODS Laboratory (University of Pittsburgh) for their support and friendship, for enlightening discussions and valuable research collaborations. I would especially like to thank Mike Wagner from RODS, without whom much of our biosurveillance research would not have been possible. Fourth, I would like to thank the many other people who have contributed to my education and research experiences. Special thanks go to my undergraduate research advisor, David Kraines, who has been a valuable source of advice and friendship over the years. Most importantly, I would like to thank my family and friends, for all of their love and support.

Contents

Chapter 1

Spatial cluster detection

1.1 Introduction

This thesis develops new statistical and computational methods for the automatic detection of spatial and space-time clusters. The basic goal of cluster detection is to automatically detect regions of space that are “anomalous,” “unexpected,” or otherwise “interesting.” These anomalous spatial patterns could correspond to a variety of phenomena, depending on the application domain: we may want to detect outbreaks of disease, clusters of stars or galaxies, brain tumors, deposits of precious metals, or a multitude of other possibilities.

This work will focus on one very general formulation of the cluster detection problem: finding regions of space where the values of some quantity (the “count”) are significantly higher than expected, given some other “baseline” information. For example, in the public health domain, we may wish to detect spatial clusters of disease cases (or some related observable quantity, such as hospital visits or medication sales) that are indicative of an emerging epidemic. Our main emphasis in this domain is prospective disease surveillance, with the goal of detecting emerging outbreaks of disease as early as possible. In the brain imaging domain, we wish to detect clusters that correspond to regions of increased or decreased brain activity. This could be used to detect brain regions that have been damaged by strokes or degenerative diseases, or to detect clusters of brain activity that allow us to differentiate between cognitive tasks: for example, we could automatically determine whether a person is reading a book or watching a movie, simply by monitoring functional magnetic resonance imaging (fMRI) images of their brain activity. In both of these applications, we have two main tasks. First, we must identify the locations, shapes, sizes, and other parameters of potential clusters, i.e. pinpointing and characterizing those spatial areas which are most relevant. Second, we must determine whether each of these anomalous regions is due to a genuine and relevant cluster, or simply a chance occurrence. In many application domains, both false positives (incorrectly reporting a cluster) and false negatives (failing to report a true cluster) have high costs: thus we want to avoid detecting insignificant or irrelevant clusters, while maintaining high power to detect any relevant clusters that do occur.

In other words, the goal of cluster detection is to answer two essential questions: is anything interesting (or unexpected) going on, and if so, where? This task can be broken down into two parts: first figuring out what we expect to see, then determining which regions deviate significantly from our expectations. In our typical formulation of the cluster detection problem, we are given a set of points s_i in space, where each point s_i has an associated *count* c_i and *baseline* b_i. Both “counts” and “baselines” can be broadly defined, depending on the application domain under consideration. For example, in the public health domain, the count c_i may represent the number of disease cases in a given area, while the baseline might be the “at-risk” population of that area. Alternatively, rather than being given the baselines in advance, we might have to infer these baselines from historical data. In any case, our main goal is to detect spatial regions S (each containing a set of one or more locations s_i) such that the counts inside S are significantly higher than expected, given the baselines. For example, in the disease surveillance domain, these may correspond to areas of high disease rate or high relative risk. This formulation allows us to be very flexible in how clusters are defined: we can choose domain-appropriate quantities for the count and baseline, choose a set of regions to search over, and incorporate either very general or very specific models of clusters and of the baseline data as appropriate for the given domain. Though we have focused here on finding spatial and spatio-temporal overdensities (higher than expected counts in space or space-time data), many other types of spatial patterns (underdensities, overdispersion, spatial and temporal correlations, etc.) may also be detectable using this general framework.

In addition to discovering these patterns, we wish to determine whether each such pattern is *significant* or if it is likely to have occurred by chance. To do so, we can either compute the *statistical significance* (p-value) of potential clusters, or in a Bayesian setting, we can compute the *posterior probability* of each cluster. In each of these cases, our method works by *hypothesis testing*: we test the null hypothesis H_0 of no clusters against a set of alternative hypotheses $H_1(S)$, each representing a cluster in some region S, and find regions where an alternative hypothesis is likely (e.g. the null hypothesis is rejected, or has low posterior probability). The models of the null and alternative hypotheses are highly dependent on the application domain under consideration, but our methods are sufficiently flexible to be used for a wide variety of such models. We typically create models based on careful study of the application domain, derive the resulting score function (e.g. likelihood ratio of the alternative vs. null hypothesis), and find the “most significant” regions (the regions with the highest values of this score function). We then use techniques such as randomization testing to compute the statistical significance of each such region, allowing us to tell which are likely to be “true” clusters and which are likely to have occurred by chance. By using sufficiently rich models of a domain, we can also distinguish between various causes of a statistically significant cluster in that domain, enabling us to detect clusters due to “relevant” causes (such as a disease outbreak) while eliminating clusters due to noisy data or a variety of other “irrelevant” factors.

The cluster detection problem presents both statistical and computational challenges. The statistical challenge is to accurately detect relevant clusters, while keeping false positives to a minimum. The computational challenge is to detect these clusters very rapidly even for massive real-world datasets. To deal with these challenges, we have developed both new statistical methods, for better and more accurate cluster detection, and new algorithmic techniques, for rapid and efficient detection of clusters. By integrating these novel spatial statistical methods and fast spatial algorithms, we have created a powerful and general framework for automatic cluster detection. Most importantly, this framework is sufficiently general to be usable for a wide variety of applications (ranging from medicine and public health to astrophysics and neuroscience), and sufficiently flexible to be easily adapted to new application domains. Here we apply our framework to two critical, real-world problems: the early detection of emerging disease epidemics, enabling more rapid epidemiological response and thus potentially saving many lives, and the detection of clusters in medical images, for purposes such as tumor detection and the monitoring of brain activity.

In the remainder of this chapter, I discuss the problem of cluster detection in more detail, and motivate the statistical methodology that will be used to solve this problem. In Section 1.2, I present several concrete examples of the cluster detection problem, focusing on applications to disease surveillance and medical imaging. In Section 1.3, I compare cluster detection to related problems in machine learning and data mining, including clustering and anomaly detection. In Section 1.4, I discuss the various issues that arise in cluster detection, and motivate the use of methods based on the *spatial scan statistic* [78]. In Section 1.5, I present the spatial scan statistic in more detail, and discuss some limitations of this approach. Finally, in Section 1.6, I describe the main contributions of the thesis, and outline the structure of the remainder of this work. Parts of this chapter have been adapted from our chapter in the *Handbook of Biosurveillance* [115]; I wish to thank my co-author Andrew Moore and editor Michael Wagner for their contributions to this work.

1.2 Applications of cluster detection

Our discussion of cluster detection will focus primarily on two application domains, disease surveillance and medical imaging. These domains are discussed in the following subsections, and considered in more detail in Chapters 6 and 7 respectively. Cluster detection is also useful in a variety of other application domains, ranging from astrophysics to forest ecology. For example, in the astrophysical domain, we might want to find a region of space that contains a higher than expected density of stars or galaxies with a given set of properties. Similarly, in forest ecology, we might want to find areas with clusters of certain types of trees, or other plants and animals. In these domains, we might use baseline information such as the total population of stars or trees respectively, adjusted for relevant covariates. Some other possible applications include the processing of radar traces (e.g. for military surveillance and reconnaissance) and the detection of terrorist groups from social network data. Many other possible application domains are discussed by Kulldorff [80], and we also consider a variety of applications in our discussion of future work (Chapter 8).

1.2.1 Cluster detection in biosurveillance

One essential application of cluster detection is in the public health domain, with the goal of detecting anomalous clusters of disease cases. These methods may be used for a variety of purposes, ranging from detection of a bioterrorist attack (an intentional release of a pathogen such as anthrax or bubonic plague) to identifying environmental risk factors for diseases such as childhood leukemia [122, 153, 88]. We focus primarily on the detection of emerging clusters of disease; these outbreaks may be caused by a naturally occurring disease epidemic (e.g. influenza), bioterrorist attack (e.g. anthrax), or environmental hazard (e.g. radiation leak). Thus we wish to perform *prospective disease surveillance*, analyzing public health data on a daily (or even hourly) basis with the goal of detecting emerging outbreaks as quickly as possible. Timely detection of outbreaks must be achieved while keeping the number of false alarms to a minimum, and thus we must be able to accurately distinguish between clusters corresponding to outbreaks and those corresponding to other irrelevant causes. By detecting outbreaks rapidly and automatically, we hope to allow more rapid epidemiological response (e.g. distribution of vaccines, public health warnings), potentially reducing the rates of mortality and morbidity.

In disease surveillance, we are given the number of disease cases of some given type in each spatial location on each day. In our typical surveillance task, we have count data aggregated at the zip code level for data privacy reasons. Thus we have a set of spatial locations s_i, where each s_i represents the longitude and latitude of a zip code centroid, and the corresponding count c_i may represent the number of disease cases of a specific type (e.g. influenza). We must also have some baseline information b_i indicating how many cases we expect to see in each zip code: this could be the underlying at-risk population of the zip code (typically denoted by p_i) or an expected count inferred from historical data. We compare these approaches in detail in Chapter 2; as we show in Chapter 4, the latter, expectation-based approach enables us to achieve more timely detection of disease outbreaks than the traditional, population-based approach.

While cluster detection can be applied to monitoring for patterns of a specific disease, we often want to perform the more general task of disease-independent monitoring: detecting anomalous clusters corresponding to any type of disease, including those of previously unknown diseases. Our typical approach to this task is *syndromic surveillance*, where we monitor data corresponding to disease symptoms. In this case, the count c_i for a given zip code s_i can be the number of emergency department visits with a given type of chief complaint (e.g. respiratory, gastrointestinal), the number of over-the-counter medication sales of a specific type (e.g. cough and cold, fever), or some other observable quantity (e.g. 911 calls, school and work absenteeism). By discovering regions with abnormally high counts of some syndrome, we can detect any type of outbreak which causes that syndrome. In addition to this increased generality, syndromic surveillance also allows us to achieve more timely detection of outbreaks, since we can detect an outbreak even before a definitive diagnosis of any given outbreak type. The utility of syndromic surveillance, and the many challenges associated with this task, are discussed in detail in Chapter 6.

As disease surveillance is a canonical example of the cluster detection task with great practical utility, we focus primarily on this task throughout our work. We consider the many statistical and computational challenges of cluster detection in this domain, and many of our solutions to these challenges can also be directly applied to other application domains. We consider statistical issues in Chapters 2, 4, and 5, presenting a general framework for cluster detection which can be applied not only to disease surveillance but to many other domains. We consider computational issues in Chapters 3 and 5, enabling us to develop general algorithms for accelerating the cluster detection task and scaling it to large datasets. In Chapter 6, we provide a detailed discussion of disease surveillance, and describe our SSS system, which is currently being used in daily practice for spatial surveillance of nationwide public health data.

1.2.2 Cluster detection in medical imaging

Automatic cluster detection has many possible applications in the medical imaging domain. One of the most important such applications is the early detection of cancerous or pre-cancerous tumors. For example, brain tumors may be detected from magnetic resonance imaging (MRI) data, or early signs of breast cancer may be discovered from mammography data. Cluster detection methods may also be useful in detecting other chronic health problems: for example, detecting diabetic retinopathy (a leading cause of blindness) from retinal exams. In these application domains, we may use several types of baseline data for comparison, including images previously taken from the same patient or "aggregate" images created from many other patients; alternatively, a "purely spatial" scan may be performed to detect high-density regions without reference to a baseline state.

In addition to the detection of abnormalities in structural images, we can also obtain useful information from *functional imaging*. For example, functional magnetic resonance imaging (fMRI) can be used to measure blood flow in the brain, creating a three-dimensional picture of brain activity. By detecting regions of increased or decreased brain activity, we could automatically discover areas that have been damaged by strokes or by degenerative diseases such as Alzheimer's and Parkinson's. Another exciting application of cluster detection is the discovery of regions of brain activity corresponding to different cognitive states. In this domain, our goals are to distinguish between subjects performing different tasks, and to discover which regions of the brain are most active in performing each task. For example, we may want to tell whether the subject is reading a book, or watching a movie, based only on their fMRI image. For this task, we may compare the subject's brain image to an image of that subject's brain under some "control condition" (such as fixating on a cursor), or simply compare two experimental conditions.

A typical fMRI image is a $64 \times 64 \times 14$ grid[1] of "voxels," where the measured "activation" of each voxel corresponds to the amount of activity in that region of the brain. Thus for fMRI cluster detection tasks, we typically have a count c_i and a baseline b_i for each voxel s_i, where c_i corresponds to the measured amount of fMRI activation in that voxel under the experimental condition, and b_i corresponds to the measured amount of fMRI activation in that voxel under the null or control condition. We note that fMRI data is typically three-dimensional, and we might also want to use time as a fourth dimension, comparing sequences of fMRI "snapshots" under the experimental and control conditions. Since the standard algorithmic framework for the spatial scan assumes only two dimensions, this demonstrates the importance of developing efficient algorithms for multidimensional spatial cluster detection. We discuss new algorithms for very fast detection of multidimensional clusters in Chapter 3, and apply these to brain imaging in Chapter 7.

1.3 Cluster detection and related problems

The cluster detection task is related to both *clustering* and *anomaly detection*, but is distinct from each. Like clustering, the goal of cluster detection is to find "clusters" (groups of data points), but rather than simply partitioning the entire dataset into groups, we search for spatial regions (each containing some set of points) where some quantity is significantly higher than expected, adjusting for quantities such as an underlying population or baseline. In clustering, the number of clusters is often fixed, while in cluster detection one of the main goals is to accurately decide whether there are *any* significant clusters, and if so, to compute where and how many clusters there are. In this respect, cluster detection is more similar to anomaly detection: we are searching for groups of points with counts that are sufficiently high to be "surprising" or "unexpected" under the assumption that no clusters exist.

The difference between cluster detection and anomaly detection is that, while anomaly detection typically focuses on single data points and asks whether each point is anomalous, cluster detection focuses on finding spatial groups or patterns which are anomalous, even if each individual point in the group might not be surprising on its own. For example, one typical (and useful) approach to anomaly detection is to learn a joint probability distribution over all features of the data, and then to detect individual records which have low probability given the model. This method has been used for a variety of applications, such as biosurveillance and network intrusion detection. A variety of methods can be used to model the "normal" data, ranging from mixture models [45] to Bayesian net-

[1] Note that this was the available resolution of fMRI images for our experiments; other fMRI images may have higher or lower spatial resolutions.

works [160] to neural networks [17]. While these methods can detect individually anomalous data points, much less work has been devoted to detecting anomalous groups or patterns. One exception is What's Strange About Recent Events (WSARE) [159, 160, 161], which detects anomalous association rules; however, this method does not take spatial locations or spatial proximity into account. Thus cluster detection differs from traditional anomaly detection methods because it does not simply detect individually anomalous locations, but incorporates information from multiple locations to detect anomalous regions of space.

We now return to the question of how cluster detection compares to clustering. As noted above, clustering and cluster detection have very different goals (partitioning data into groups versus finding statistically anomalous regions). However, some clustering methods, commonly referred to as "density-based" clustering, partition the data based on the density of points in space. Thus, the highest density partitions found by these methods will be areas with an excess of points, corresponding to areas with a higher than expected count c_i in our model. As a result, these partitions may correspond to the anomalous spatial regions that we are interested in detecting.

A variety of density-based clustering methods have been proposed. Two of the most well-known are DBSCAN [46] and CLIQUE [4], each of which works by finding small dense regions and aggregating these high-density regions together in bottom-up fashion. DBSCAN searches for points which have many other points nearby (at least m points within distance ϵ, where m and ϵ are user-specified input parameters), while CLIQUE aggregates points to a uniform grid and searches for grid cells containing a high proportion of points (greater than some user-specified parameter τ). The set of all such "dense" points or cells is then used to form clusters: DBSCAN aggregates nearby dense points, then also includes the other points in the ϵ-neighborhood of these points, while CLIQUE defines a cluster as a maximal set of connected dense cells. Many other density-based clustering approaches build on these two methods: MAFIA [59] is an extension of CLIQUE to non-uniform grids, DENCLUE [68] is similar to DBSCAN but uses local maxima of the density function as its starting points from which clusters are built, and STING [155] is a grid-based algorithm that uses quadtree decomposition to efficiently approximate DBSCAN's results. Han et al. [64] provide an excellent survey of these and other clustering methods; another closely related method is bump hunting [49], which uses a greedy heuristic search (iteratively removing or adding some portion of the data such that density is maximized) to locate dense regions.

Density-based clustering approaches have some advantages over our (and other) cluster detection methods: they are fast to compute, have more flexibility in defining cluster shape, and are often usable for massive and high-dimensional datasets. However, density-based clustering is not adequate for the cluster detection task for a variety of reasons. First, we do not simply want to find overdensities of counts, but also to draw substantial conclusions about the regions we find: in particular, whether each region represents a significant cluster or is likely to have occurred by chance. In fields such as disease surveillance, it is essential to minimize the number of false positives, while maintaining high power to detect any true clusters (e.g. disease outbreaks) that arise. Thus hypothesis testing (whether by statistical significance testing in a frequentist setting, or by computing posterior probabilities of potential clusters in a Bayesian setting) is an essential part of the cluster detection problem, but density-based clustering methods cannot give us this information.

Second, cluster detection methods attempt to draw conclusions about entire regions, rather than aggregating single cells as in density-based clustering. This broader focus allows cluster detection to be more sensitive for detecting small (but significant) changes in counts, if the effects are sufficiently large in spatial extent. For example, our spatial scan methods are able to detect a 20% increase in the underlying disease rate of a region, while both clustering approaches and human observers may have trouble with this task. The key is that, though none of the individual counts are sufficiently elevated to be significant by themselves, the increase can be perceived when counts are aggregated at the region level.

Finally, density-based clustering methods cannot deal adequately with spatially (and temporally) varying baselines, because they are specific to the notion of density as number of points per unit area.[2] Adjusting for variable baselines is particularly essential for real-world disease surveillance, where our expected counts will vary based on population, seasonal trends, and other covariates. Our cluster detection approaches allow us to deal with counts and baselines in a principled probabilistic framework, finding the global optimum of any score function (e.g. likelihood ratio statistic) that distinguishes clusters from non-clusters, and thus identifying the most likely cluster given the counts and baselines.

Thus, while density-based clustering and anomaly detection are closely related to the cluster detection problem, neither of these methods are able to perform important aspects of the cluster detection task, including aggregation of information across multiple spatial locations, finding whether detected regions are significant, adjusting for varying baselines, and generalizing to the models and statistics which are most appropriate for any given application domain. In the remainder of this thesis, we motivate and describe cluster detection approaches based on a generalization of the *spatial scan statistic* [78], which enable us to achieve all of these desired criteria.

1.4 Motivation for the spatial scan statistic

Let us consider the example of disease surveillance, assuming that we are given the count (number of disease cases) c_i, as well as the expected count (mean μ_i and standard deviation σ_i), for each zip code s_i. How can we tell whether any zip code has a number of cases that is significantly higher than expected? One simple possibility would be to perform a separate statistical test for each zip code, and report all zip codes that are significant at some level α. For example, we might want to detect all zip codes with observed count more than three standard deviations above the mean ($p < .0013$). However, there are two main problems with this simple approach. First, treating each zip code separately prevents us from using information about the *spatial proximity* of adjacent zip codes. For instance, while a single zip code with count two standard deviations higher than expected might not be sufficiently surprising to trigger an alarm, we would probably be interested in detecting a cluster of adjacent zip codes each with count two standard deviations higher than expected. Thus, the first problem with performing separate statistical tests for each zip code is reduced power to detect clusters spanning multiple zip codes: we cannot detect such increases unless the amount of increase is so large as to make each zip code individually significant. A second, and somewhat more subtle, problem is that of *multiple hypothesis testing*. We typically perform statistical tests to determine if an area is significant at some fixed level α, such as $\alpha = 0.05$, which means that if there is no abnormality in that area (i.e., the “null hypothesis” of no clusters is true) our probability of a false alarm is at most α. A lower value of α results in less false alarms, but also reduces our chance of detecting a true cluster. Now let us imagine that we are searching for disease clusters

[2] While we could simply normalize the counts in a density-based clustering approach by dividing each count by its associated baseline, this approach is inadequate because a given overdensity of counts (e.g. 10% higher than expected) is more significant for larger values of count and baseline.

in a large area containing 1000 zip codes, and that there happen to be no outbreaks today, so any areas we detect are false alarms. If we perform a separate significance test for each zip code, we expect each test to trigger an alarm with probability $\alpha = 0.05$. But because we are doing 1000 separate tests, our expected number of false alarms is $1000 \times 0.05 = 50$.[3] Moreover, if these 1000 tests were independent, we would expect to get at least one false alarm with probability $1 - (1 - 0.05)^{1000} \approx 1$. Of course, counts of adjacent zip codes are likely to be correlated, so the assumption of independent tests is not usually correct. The main point here, though, is that we are almost certain to get false alarms every day, and the number of such false alarms is proportional to the number of tests performed. One way to correct for multiple tests is the Bonferroni method [20]: if we want to ensure that our probability of getting any false alarms is at most α, we report only those regions which are significant at level $\frac{\alpha}{N}$, where N is the number of tests. The problem with the Bonferroni method is that it is too conservative, reducing the power of the test to detect true clusters. In our example, with $\alpha = 0.05$ and $N = 1000$, we only signal an alarm if a region's p-value is less than 0.00005, and thus only very obvious clusters can be detected.

As an alternative to this simple method, we can choose a set of regions to search over, where each region consists of a set of one or more zip codes. We can define the set of regions based on what we know about the size and shape of potential clusters; we can either fix the region shape and size, or let these vary as desired. We can then do a separate test for each region rather than for each zip code. This resolves the first problem of the previous method: assuming we have chosen the set of regions well, we can now detect clusters whether they affect a single zip code, a large number of zip codes, or anything in between. However, the disadvantage of this method is that it makes the multiple hypothesis testing problem even worse: the number of regions searched, and thus the number of tests performed, is typically much larger than the number of zip codes. In principle, the number of regions could be as high as 2^Z, where Z is the number of zip codes, but in practice the number of regions searched is much smaller (because we want to enforce constraints on the connectedness, size, and shape of regions). For example, if we consider circular regions centered at the centroid of some zip code, with continually varying radius (assuming that a region contains all zip codes with centroids inside the circle), the number of distinct regions is proportional to Z^2. For the example above, this would give us one million regions to search, creating a huge multiple hypothesis testing problem; less restrictive constraints (such as testing ellipses rather than circles) would require testing an even larger number of regions.

This method of searching over regions, without adjusting for multiple hypothesis testing, was first used by Openshaw et al. [122] in their Geographical Analysis Machine (GAM). The GAM searches for disease outbreaks by testing a large number of overlapping circles of fixed radius, and drawing all of the significant circles on a map; Figure 1.1 gives an example of what the output of the GAM might look like. Because we expect a large number of circles to be drawn even if there are no outbreaks present, the presence of detected clusters is not sufficient to conclude that there is an outbreak. Instead, the GAM can be used as a descriptive tool for outbreak detection: whether any outbreaks are present, and the location of such outbreaks, must be inferred manually from the number and spatial distribution of detected clusters. For example, in Figure 1.1, the large number of overlapping circles in the upper right of the figure may indicate an outbreak, while the other circles might be due to chance. The problem is that we have no way of determining whether any given circle or set of circles is statistically significant, or whether they are due to chance and multiple

[3] This is true by linearity of expectation, regardless of whether the 1000 tests are independent.

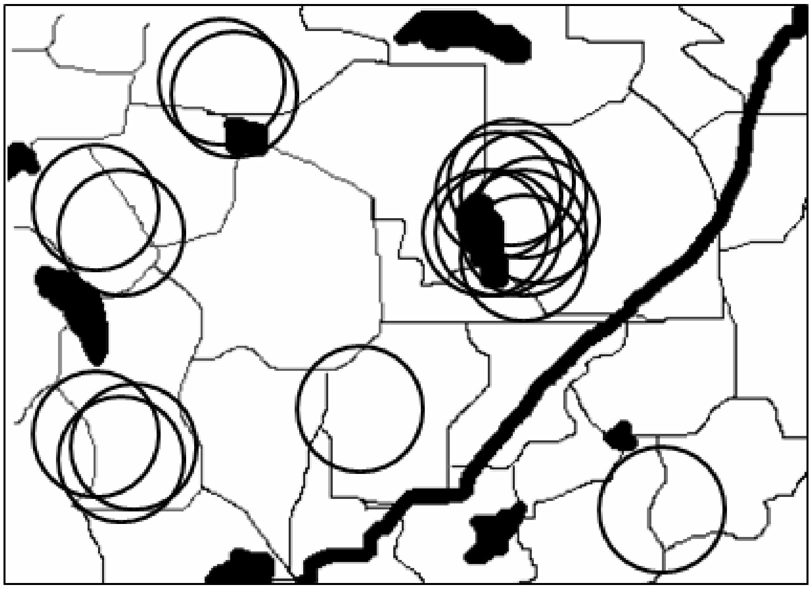


Figure 1.1: Example output of the Geographical Analysis Machine, with significant regions shown as circles.

testing; it is also difficult to precisely locate those circles which are most likely to correspond to true outbreaks. Besag and Newell [15] propose a related approach, where the search is performed over circles containing a fixed number of disease cases; this approach also suffers from the multiple hypothesis testing problem, but again is valuable as a descriptive method for visualizing potential disease clusters.

The scan statistic was first proposed by Naus [108] as a solution to the multiple hypothesis testing problem. Let us assume we have a score of some sort for each region: for example the Z-score, $Z = \frac{c-\mu}{\sigma}$. The Z-score is the number of standard deviations that the observed count c is higher than the expected count μ; a large Z-score indicates that the observed number of cases is much higher than expected. Rather than triggering an alarm if any region has Z-score higher than some fixed threshold, we instead find the distribution of the *maximum* score of all regions under the null hypothesis of no clusters. This distribution tells us what we should expect the most alarming score to be when the system is executed on data in which there are no clusters present (i.e. no outbreaks, in the case of disease surveillance). Then we compare the score of the highest-scoring (most significant) region on our data against this distribution to determine its statistical significance (or p-value). In other words, the scan statistic attempts to answer the question, “If there were no clusters, and we searched over all of these regions, how likely would we be to find any regions that score at least this high?” If the analysis shows that we would be very unlikely to find any such regions under the null hypothesis, we can conclude that the discovered region is a significant cluster. The main advantage of the scan statistic approach is that we can adjust correctly for multiple hypothesis testing: we can fix a significance level α, and ensure that the probability of having any false alarms on a given day is at most α, regardless of the number of regions searched. Moreover, because the scan statistic accounts for the fact that our tests are not independent, it will typically have much higher detection power than a Bonferroni-corrected method. In some applications, the scan statistic results in a most powerful statistical test [78].

Although the scan statistic focuses on finding the single most significant region, it can also be used to find multiple regions: secondary clusters can be examined, and their significance found, though the test is typically somewhat conservative for these. The technical difficulty, though, is finding the distribution of the maximum region score under the null hypothesis. Turnbull [146] solved this problem for circular regions of fixed population, using the maximum number of cases in a circle as the test statistic, and using the method of randomization testing (discussed below) to find the statistical significance of discovered regions. The disadvantage of this approach is that it requires a fixed population size circle, and thus a multiple hypothesis testing problem still exists if we want to search over regions of multiple sizes or shapes. Similarly, Anderson and Titterington [8] propose a scan statistic which searches over fixed size rectangles. Kulldorff and Nagarwalla [88, 78] solved the problem for variable size regions using a likelihood ratio test: the test statistic is the maximum of the likelihood ratio under the alternative and null hypotheses, where the alternative hypothesis represents clustering in that region and the null hypothesis assumes no clusters. We discuss their method, the "spatial scan statistic," in the following section.

1.5 Detailed description of the spatial scan statistic

The spatial scan statistic, first presented by Kulldorff and Nagarwalla [88, 78], is a powerful and general method for spatial cluster detection. It is in common use by the public health community for finding significant spatial clusters of disease cases, for purposes ranging from detection of bioterrorist attacks to identification of environmental risk factors. For example, scan statistics have been applied to find spatial clusters of chronic diseases such as breast cancer [84] and leukemia [69], as well as work-related hazards [83], West Nile virus [107] and various other types of outbreak. Kulldorff has implemented the spatial scan statistic in his SaTScan software [87], available at www.satscan.org, and this software is widely used in the public health domain.

In its original formulation, Kulldorff's statistic assumes that we have a set of spatial locations s_i, and are given a count c_i and a population p_i corresponding to each location. For example, each s_i may represent the centroid of a census tract, the corresponding count c_i may represent the number of respiratory emergency department visits in that census tract, and the corresponding population p_i may represent the "at-risk population" of that census tract, derived from census population and possibly adjusted for covariates. The statistic makes the assumption that each observed count c_i is drawn randomly from a Poisson distribution with mean $q_i p_i$, where p_i is the (known) at-risk population of that area, and q_i is the (unknown) risk, or underlying disease rate, of that area. The risk is the expected number of cases per unit population: that is, we expect to see a number of cases equal to the product of the population and the risk, but the observed number of cases may be more or less than this expectation due to chance. Thus our goal is to determine whether observed increases in count in a region are due to increased risk, or chance fluctuations. The Poisson distribution is commonly used in epidemiology to model the underlying randomness of observed case counts, making the assumption that the variance is equal to the mean. If this assumption is not reasonable (i.e. counts are "overdispersed" with variance greater than the mean, or "underdispersed" with variance less than the mean), we should instead use a distribution which separately models mean and variance, such as the Gaussian or negative binomial distributions. We also assume that each count c_i is drawn independently, though the model can be extended to account for spatial correlations between nearby locations.

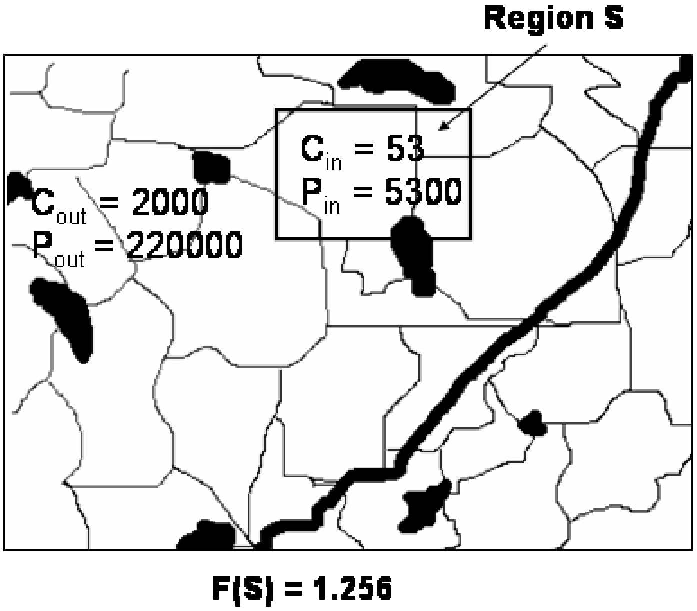


Figure 1.2: Evaluation of the score function $F(S)$ for the given region S.

1.5.1 Kulldorff's model

As discussed above, Kulldorff's spatial scan statistic attempts to detect spatial regions where the underlying disease rates q_i are significantly higher inside the region than outside the region. Thus we wish to test the null hypothesis H_0 ("the underlying disease rate is spatially uniform") against the set of alternative hypotheses $H_1(S)$: "the underlying disease rate is higher inside region S than outside region S". More precisely, we have:

H_0: $c_i \sim$ Poisson$(q_{all}p_i)$ for all locations s_i, for some constant q_{all}.
$H_1(S)$: $c_i \sim$ Poisson$(q_{in}p_i)$ for all locations s_i in S, and $c_i \sim$ Poisson$(q_{out}p_i)$ for all locations s_i outside S, for some constants $q_{in} > q_{out}$.

Note that the counts c_i and populations p_i are known a priori, while the values of the disease rates q_{in}, q_{out}, and q_{all} are unknown; these latter values will be inferred from the data by maximum likelihood estimation.

The test statistic that we use is the likelihood ratio, that is, the likelihood (denoted by Pr) of the data under the alternative hypothesis $H_1(S)$ divided by the likelihood of the data under the null hypothesis H_0. This gives us, for any region S, a score function $F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$. For Kulldorff's statistic, we obtain $F(S) = \left(\frac{C_{in}}{P_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{P_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{P_{all}}\right)^{-C_{all}}$, if $\frac{C_{in}}{P_{in}} > \frac{C_{out}}{P_{out}}$, and $F(S) = 1$ otherwise; this formula is derived in Chapter 2. In this equation, C_{in} and C_{out} represent the aggregate count $\sum c_i$ inside and outside region S, and P_{in} and P_{out} represent the aggregate population $\sum p_i$ inside and outside region S, respectively. We also define $C_{all} = C_{in} + C_{out}$ and $P_{all} = P_{in} + P_{out}$. See Figure 1.2 for an example of the evaluation of $F(S)$ for a region. Kulldorff [78] proved that this likelihood ratio statistic is individually most powerful for finding a single region of elevated disease rate: for the given model assumptions (i.e. the hypotheses H_0 and $H_1(S)$ given above), for a fixed false alarm rate, and for a given set of regions searched, it is more likely to detect the cluster than any other test statistic.

1.5.2 Finding the most significant regions

Given the above test statistic $F(S)$, the spatial scan statistic method can be easily applied by choosing a set of regions S, calculating the score function $F(S)$ for each of these regions, and obtaining the highest scoring region S^* and its score $F^* = F(S^*)$. We can imagine this procedure as moving a "spatial window" (like the rectangle drawn in Figure 1.2) all around the search area, changing the size and shape of the window as desired, and finding the window which gives the highest score $F(S)$. Even though there are an infinite number of possible window positions, sizes, and shapes, we only need to evaluate the score function a finite number of times, since any two regions containing the same set of spatial locations s_i will have the same score. The region with the highest score $F(S)$ is the "most significant region," i.e. the region which is most likely to have been generated under the alternative hypothesis rather than the null hypothesis, and thus the region which is most likely to be a cluster. We typically search over the set of all "spatial windows" of a given shape and varying size, for example, circular regions [78], square regions [111], or rectangular regions [112]. Searching over a set of regions which includes both compact and elongated regions (e.g. rectangles or ellipses) has the advantage of higher power to detect elongated clusters resulting from wind dispersal of pathogens, but because the number of regions to search is increased, this also makes the scan statistic more difficult to compute. Computational issues are discussed in more detail in Chapter 3.

1.5.3 Statistical significance testing

Once we have found the regions with the highest scores $F(S)$, we must still determine which of these "potential clusters" are likely to be "true clusters" resulting from a disease outbreak, and which are likely to be due to chance. To do so, we calculate the statistical significance (p-value) of each potential cluster, and all clusters with p-value less than some fixed significance level α are reported. Because of the multiple hypothesis testing problem discussed above, we cannot simply compute separately whether each region score $F(S)$ is significant, because we would obtain a large number of false positives, proportional to the number of regions searched. Instead, for each region S, we ask the question, "If this data set were generated under the null hypothesis H_0, how likely would we be to find any regions with scores higher than $F(S)$?" To answer this question, we use the method known as *randomization testing*: we randomly generate a large number of "replicas" under the null hypothesis, and compute the maximum score $F^* = \max_S F(S)$ of each replica. We typically use Monte Carlo randomization [43] to generate these replicas, but permutation testing [60] can also be used to test the null hypothesis of exchangeability of counts. More precisely, in the Monte Carlo approach, each replica is a copy of the original search area that has the same population values p_i as the original, but has each value c_i randomly drawn from a Poisson distribution with mean $\frac{C_{all}}{P_{all}} p_i$, where C_{all} and P_{all} are respectively the total number of cases and the total population for the original search area. Thus the assumption under the null hypothesis is that all counts are generated with a uniform disease rate, equal to the observed disease rate $q = \frac{C_{all}}{P_{all}}$ for the original dataset.

Once we have obtained F^* for each replica, we can compute the statistical significance of any region S by comparing $F(S)$ to these replica values of F^*, as shown in Figure 1.3. The p-value of

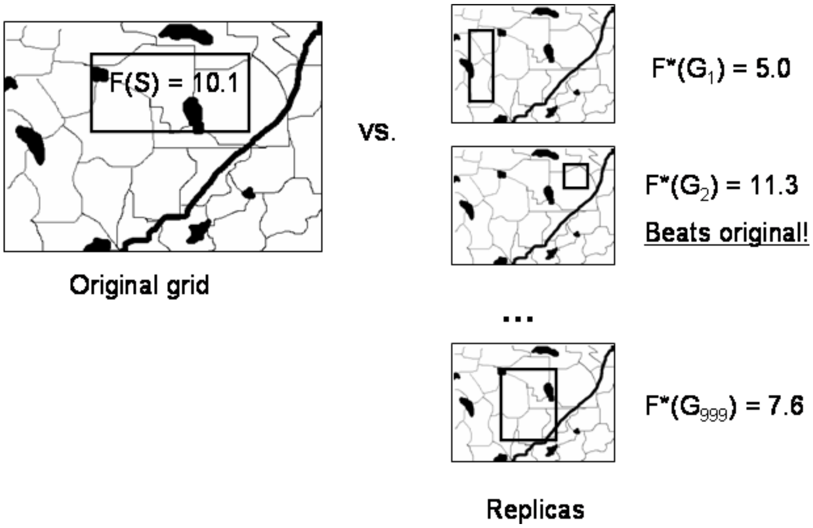


Figure 1.3: Example of randomization testing for computing the statistical significance of region S. If seven of the 999 replicas have higher scores than $F(S)$, then the p-value of S is $\frac{7+1}{999+1} = 0.008$.

region S can be computed as $\frac{R_{beat}+1}{R+1}$, where R is the total number of replicas created, and R_{beat} is the number of replicas with F^* greater than $F(S)$. If this p-value is less than our significance level α, we conclude that the region is significant (likely to be a true cluster); if the p-value is greater than α, we conclude that the region is not significant (likely to be due to chance). We typically start from the most significant region S^* and test regions in order of decreasing $F(S)$, since if a region S is not significant, no region with lower $F(S)$ will be significant. We note that the randomization testing approach given here has the benefit of bounding the overall false positive rate: regardless of the number of regions searched, the probability of any false alarms is bounded by the significance level α. Also, the more replications performed (i.e. the larger the value of R), the more precise the p-value we obtain; a typical value would be $R = 999$. However, since the run time is proportional to the number of replications, this dramatically increases the amount of computation necessary.

We note that, if we could compute a closed-form distribution for the test statistic F^* under the null hypothesis, this would allow much faster computation of statistical significance by making randomization testing unnecessary. Much work has been done on deriving distributions of the one-dimensional and two-dimensional scan statistics, typically assuming a fixed scan region and uniform underlying measure. Examples of such work include Naus [108], Loader [97], and Alm [6, 7]; more details are given in Glaz et al. [57, 58]. Nevertheless, the distribution of the scan statistic is not known in the general case of non-uniform underlying populations and varying region size and shape, and thus randomization testing is still necessary. Recent empirical results by Abrams et al. [1] suggest that the null distribution of Kulldorff's statistic is fit well by a Gumbel extreme value distribution; thus they propose running a smaller number of replications under the null (e.g. $R = 99$) to find the mean and variance, and using the inferred Gumbel distribution to calculate p-values. At present, however, we believe that our Bayesian spatial scan statistic, presented in Chapter 5, is the only known spatial scan method that does not require randomization in the general case.

Another alternative to randomization testing would be to perform a separate significance test for each spatial region, and then to correct for multiple hypothesis testing by using the Bonferroni correction [20], the False Discovery Rate (FDR) criterion [12], or one of the many other methods in the multiple testing literature. However, all of these methods either assume independence of tests, or alternatively, are conservative bounds which hold for arbitrary dependencies. The spatial scan performs tests for a large set of overlapping spatial regions, and this overlap creates a complex dependency structure for the multiple tests. As a result, methods that assume independent tests are unable to bound the false positive rate, while bounds that hold for arbitrary dependencies are far too conservative, resulting in reduced detection power. The use of randomization testing correctly accounts for the complex dependency structure, maximizing detection power while providing provable bounds on false positive rate under the null.

1.5.4 Limitations of the spatial scan statistic

The spatial scan statistic is a powerful method for cluster detection, and as such it has the potential to be a valuable tool for finding clusters not only in the public health context, but also in many other application domains. However, the utility of the spatial scan for disease surveillance and its applicability to other domains have been limited by several factors. First, the spatial scan requires us to search over a huge set of regions for each of a large number of Monte Carlo replications. As a result, this method does not scale well to large datasets: for many real-world applications, the traditional spatial scan method is computationally infeasible. Even for moderate-sized datasets, the spatial scan may take hours or days to run: for example, Kulldorff's SaTScan software was unable to run on a dataset with 600,000 records and 17,000 distinct spatial locations, and required four hours to run on a smaller dataset with 60,000 records and 8,400 distinct spatial locations [114]. This lack of scalability limits the usefulness of spatial scanning to relatively small datasets and non-time-critical applications; new computational methods must be developed to make the spatial scan computationally feasible for large-scale surveillance tasks (e.g. nationwide disease surveillance) where rapid detection time is critical. Additionally, computational considerations limit the types of clusters that can be found: for example, Kulldorff's algorithm [78] limits the search to compact (circular) clusters, and has low power to detect elongated regions. A search over elongated regions (e.g. rectangles) would take several weeks for nationwide public health data, which is far too slow for our outbreak detection task. We solve these problems by proposing two distinct algorithms for making the spatial scan fast and scalable, enabling us to rapidly search over elongated and multidimensional rectangular clusters. Our fast spatial scan, discussed in Chapter 3, reduces the search time per replication by only searching a small fraction of regions (those which might have high scores) and proving that the other regions do not need to be searched. This results in speedups of 100-1000x with no loss of accuracy, i.e. the fast spatial scan returns exactly the same region and p-value as a naïve search over rectangles, but much faster. Our Bayesian spatial scan, discussed in Chapter 5, avoids the need for randomization testing, thus only searching the original dataset rather than the large number of replica datasets and also resulting in a 1000x speedup.

A second limitation of the spatial scan statistic is the inflexibility of its statistical model. Kulldorff [78] proposed binomial and Poisson scan statistic models, but did not consider how the scan statistic might be generalized to an arbitrary application domain where these models might not be accurate or appropriate. Most importantly, the traditional spatial scan approach is insufficient for syndromic disease surveillance for several reasons. By assuming that disease counts will be proportional to population under the null hypothesis of no outbreaks, the statistic fails to account for spatial or temporal variation in the underlying disease rate. In practice, we see large amounts of spatial variation (due to factors such as the age and health of the population, environmental hazards, etc.) as well as temporal variation (due to day of week effects, seasonal trends, holidays, weather, promotional sales of medications, etc.) All of these factors lead to reduced detection power in the disease surveillance domain; other application domains will also have a variety of such confounding factors and causes of "false positives" which impede our ability to accurately detect true clusters. The traditional approach is not sufficiently flexible to model and incorporate these factors into the cluster detection task. We solve this problem by proposing the "generalized spatial scan" framework discussed in Chapter 2, and we consider how many of the confounding factors can be included as part of our models. All of our new statistics (e.g. the "expectation-based space-time scan statistics" of Chapter 4, the "Bayesian scan statistic" of Chapter 5, and many others) are special cases of this general framework which allow more accurate detection of relevant and useful clusters in real-world application. These new statistics also allow us to address several other limitations of the traditional method, by enabling us to incorporate prior information, combine multiple data streams, and differentiate between "relevant" and "irrelevant" causes of a statistically significant cluster.

1.6 Contributions of this work

This work makes four main contributions to the state of the art in cluster detection: development of a powerful and widely applicable statistical framework for detecting clusters, development of spatial algorithms and data structures for very fast detection of clusters, application of these statistics and algorithms to make real-world contributions to disease surveillance and brain imaging, and extension of the range of problems to which cluster detection methods can be applied. First, we have developed the *generalized spatial scan* framework, a flexible, model-based framework for computationally efficient cluster detection in diverse application domains. One very useful application of this framework is an *expectation-based* approach, where we infer the expected count of each spatial location from historical data using time series analysis, then find spatial regions with higher than expected counts. For example, we can detect disease outbreaks by daily monitoring of over-the-counter drug sales, inferring how many sales we expect to see based on historical sales data, and detecting regions where the recent sales are abnormally high. We have demonstrated that the expectation-based disease surveillance approach can detect emerging epidemics faster than traditional methods. Even earlier detection was achieved by extending our framework to the *space-time* case, enabling us to detect clusters which may arise either quickly or gradually, and developing new statistical techniques for detecting *emerging clusters*, where the effects of the cluster increase over time.

A second contribution of this work is the development of the *fast spatial scan* algorithm for cluster detection, which incorporates new multi-resolution search methods and a novel spatial data structure (the "overlap-kd tree") to make cluster detection methods 100-1000x faster with no loss of accuracy. This algorithm enables us to perform cluster detection in under an hour for massive datasets which would otherwise require weeks of computation. The fast spatial scan has been incorporated into our generalized spatial scan framework, making this framework computationally feasible (and very fast) for disease surveillance and many other real-world detection problems. By extending the fast spatial scan to elongated clusters and multi-dimensional datasets, we have vastly increased the set of application domains to which cluster detection methods can be applied; these extensions also enable us to perform fast space-time cluster detection and to use non-spatial attributes (such as patient age and gender) as additional search dimensions. We believe that the overlap-kd tree data structure will also be useful for accelerating spatial search algorithms for a variety of other problem domains.

A third contribution of this thesis is the development of a Bayesian cluster detection approach, the *Bayesian spatial scan*. This approach was shown to have higher power to detect clusters than the typical frequentist hypothesis testing approach, as well as being hundreds of times faster (i.e. comparable in speed to the fast spatial scan). The Bayesian approach also has several other advantages over the frequentist method: since it computes the posterior probability of each potential cluster, its results are easy to interpret and visualize, and (as discussed in Chapter 5) it can also be extended more easily to the multivariate case.

In addition to developing general statistical and algorithmic methods for automatic cluster detection, we have applied these methods to make several important contributions to the disease surveillance and brain imaging domains. In retrospective case studies on known disease outbreaks, our methods demonstrated impressive results: for example, we were able to detect an outbreak of gastroenteritis in Walkerton, Ontario, a full day faster than other automatic disease surveillance systems. Similar results were obtained in semi-synthetic testing, i.e. detection of simulated outbreaks injected into real-world data. Through case studies in the brain imaging domain, we also demonstrated the ability of the system to detect relevant clusters of brain activity. The most important "applied" contribution of this thesis is the development and deployment of a system for nationwide prospective disease surveillance. Every day, this system receives emergency department and over-the-counter drug sales data from over 20,000 stores and hospitals nationwide, uses our automatic cluster detection methods to find potential outbreaks of disease, and makes these results available to state and local public health officials through a web-based graphical interface. We currently have several public health departments using our software to help them detect epidemics, and their feedback has been valuable for the iterative development of our system and the underlying models and methods. We are also working to integrate our cluster detection methods with several other systems and methods for large-scale disease surveillance.

In the remainder of this thesis, I will discuss these statistical and algorithmic contributions in more detail. Chapter 2 presents our generalized spatial scan framework for cluster detection, and considers how this framework can be applied to detect useful and relevant clusters in real-world application. Chapter 3 presents our fast spatial scan algorithm, and demonstrates that this algorithm enables us to detect clusters 100-1000x faster on real datasets without any loss of accuracy. Chapter 4 extends our cluster detection methods to the detection of emerging space-time clusters, and shows that these methods achieve accurate and timely detection of emerging outbreaks of disease. Chapter 5 describes our Bayesian spatial scan statistic, which allows us to incorporate prior knowledge and observations of multiple data streams together in a principled probabilistic framework; we demonstrate that this results in both higher detection power and much faster run time in practice. Chapters 6 and 7 apply our methods to two application domains, disease surveillance and brain imaging, and demonstrate that we can detect useful and relevant clusters in each domain. Finally, Chapter 8 concludes by discussing several important areas for future work.

Chapter 2

A general statistical framework for cluster detection

2.1 Introduction

Spatial cluster detection has two main goals: to identify the locations, shapes, and sizes of potentially anomalous spatial regions, and to determine whether each of these potential clusters is more likely to be a "true" cluster or simply a chance occurrence. In other words, we wish to answer the questions, is anything unexpected going on, and if so, where? This task can be broken down into two parts: first figuring out what we expect to see, and then determining which regions deviate significantly from our expectations. For example, in the application of disease surveillance, we examine the spatial distribution of disease cases, and our goal is to determine whether any regions have sufficiently high case counts to be indicative of an emerging disease epidemic in that area. Thus we first infer the baseline (e.g. at-risk population, or expected number of cases) for each spatial location, then determine which (if any) regions have significantly more cases than expected. While we could conceivably perform a separate statistical test for each spatial location, this simple approach fails to account for the spatial proximity of locations, and suffers from a severe problem of *multiple hypothesis testing*. As discussed in Chapter 1, if we were to perform a separate hypothesis test at level α for each spatial location, the total number of false positives that we expect would be $Y\alpha$, where Y is the total number of locations tested. For large Y, we are almost certain to get huge numbers of false alarms; alternatively, we would have to use a threshold α so low that the power of the test would be drastically reduced.

To deal with these problems, Kulldorff [78] proposed the spatial scan statistic. This method searches over a given set of spatial regions (where each region consists of a set of locations), finding those regions which are most likely to be generated under the "alternative hypothesis" of clustering rather than the "null hypothesis" of no clustering. A likelihood ratio test is used to compare these hypotheses, and randomization testing is used to compute the p-value of each detected region, correctly adjusting for multiple hypothesis testing. Thus, we can both identify potential clusters and determine whether each is significant.

Our recent work on spatial cluster detection has two main emphases: first, to generalize Kulldorff's spatial scan statistic to a larger class of underlying models, enabling us to derive useful and accurate statistics for a wide variety of application domains, and second, to make these methods computationally tractable even for massive real-world datasets. Here we focus primarily on the first goal, developing a general statistical framework which is applicable and useful for a wide variety of application domains. Many of the statistics we derive are also computationally efficient, in that they can be computed simply from some additive sufficient statistics of the region under consideration. Moreover, we have integrated the "fast spatial scan" algorithms discussed in the next chapter into this general framework, thus enabling both accurate and very fast cluster detection.

In the remainder of this chapter, I present our general statistical methodology for spatial cluster detection. In Section 2.2, I present the "generalized spatial scan" framework, and consider the general issues and questions that arise in applying this framework to any specific problem domain. In Section 2.3, I present four simple models which may be used within this framework, and derive computationally efficient scan statistics for each model. These four models share several simplifying assumptions, but differ in two respects: how the baseline information is interpreted ("expectation-based" versus "population-based" approaches) and how counts are assumed to be generated. Finally, in Section 2.4, I present three more complex models, which may be useful in domains where the simplifying assumptions of Section 2.3 are not valid. Parts of this chapter have been adapted from our paper in the 2005 KDD Workshop on Data Mining Methods for Anomaly Detection [113]. I wish to thank my co-author Andrew Moore for his contributions to this work.

2.2 The generalized spatial scan framework

In this section, we present the "generalized spatial scan" framework for spatial cluster detection. As is suggested by its name, this framework is a generalization of Kulldorff's spatial scan statistic [78] which allows much greater flexibility in the underlying models, statistics, and algorithms. This has several important advantages over the original spatial scan. First, different application domains require different models of the data, and rely on different types of baseline information; statistics that have high power to detect clusters in one domain might perform poorly in a different application. Thus it is highly advantageous to have a framework where we can simply "plug in" new domain models and derive statistics which are useful for detecting relevant clusters in the new domain. Not only can we choose the models which are most appropriate (for instance, deciding whether to account for overdispersion and spatial correlation of counts), but we can also choose to detect different types of clusters (for instance, clusters with higher than expected counts compared to the counts outside the cluster, or compared to historical data). A second advantage of the general framework is an iterative development approach: we can start out with simple models, putting these techniques into daily practice in a new application domain, then examine the resulting clusters that are detected. We can then adapt the model appropriately to increase detection power and reduce false positives in that domain. Many real-world datasets contain a variety of data irregularities and other unexpected and unmodeled phenomena, and thus simpler models might pick up these irregularities rather than the clusters we are actually interested in detecting. By adjusting our models to account for these phenomena, we can ensure reasonable false positive rates while still maintaining high power to detect any real clusters which may occur. The final advantage of our general framework is the careful consideration of tradeoffs between computational tractability and the relevance of detected clusters. In addition to presenting a variety of statistics which are both useful and computationally tractable, we can also use the "fast spatial scan algorithm" discussed in Chapter 3 to detect these clusters hundreds or thousands of times faster. Integration of these fast algorithms into the general framework not only makes our general cluster detection techniques more useful in real practice, but also extends the scope of our methods by allowing detection of elongated, rotated, and multi-dimensional clusters.

The generalized spatial scan framework consists of the following six steps:

1. Obtain data for a set of spatial locations s_i.
2. Choose a set of spatial regions to search over, where each spatial region S consists of a set of spatial locations s_i.
3. Choose models of the data under H_0 (the null hypothesis of no clusters) and $H_1(S)$ (the alternative hypothesis assuming a cluster in region S).
4. Derive a "score function" $F(S)$ based on $H_1(S)$ and H_0.
5. Find the "most interesting" regions, i.e. those regions S with the highest values of $F(S)$.
6. Determine whether each of these regions is "interesting," either by performing significance testing or calculating posterior probabilities.

We now consider each step of this framework in detail, giving some idea of the relevant decisions that must be made when applying our methods to a new application domain. In Chapters 6 and 7, we discuss two such application domains, disease surveillance and brain imaging; here we discuss the methods more generally, considering those issues which apply to any domain.

2.2.1 Obtain data for a set of spatial locations s_i

The spatial scan statistic assumes that we are given data for a set of spatial locations s_i. Typically, these locations are assumed to be points in some d-dimensional Euclidean space, with the coordinates of each point given. In the disease surveillance domain, for example, we are typically given data aggregated at the zip code level, and taking the latitude and longitude of the zip code centroid gives us a point in two-dimensional space.[1] In fMRI brain imaging, on the other hand, we are typically given activation data for a uniform $64 \times 64 \times 14$ grid of voxels, and thus each location is a point (with integer coordinates) in three-dimensional space.

For each spatial location s_i, we must have two quantities, a *count* c_i, and a *baseline* b_i. In the disease surveillance domain, the count may represent the number of disease cases of some specific type corresponding to spatial location s_i, while the baseline may represent some quantity such as the expected number of cases of that type or the at-risk population. In any case, the goal of our method is to find regions where the counts are higher than expected, given the baselines.

We typically assume that the counts are given in advance, while the baselines may be either given (e.g. population from census data) or inferred (e.g. from historical data or expert knowledge). For example, one simple way of inferring baselines would be to estimate today's expected count b_i in a zip code by the mean daily count in that zip code over the past D days. For many datasets, more complicated methods of time series analysis should be used to infer baselines; for example, in the over-the-counter drug sales data, we must account for both seasonal and day-of-week effects.

[1] Because a zip code is actually an irregular region in space rather than a single point, an alternative would be to assume that cases are spread over the entire zip code area, either uniformly or according to some known distribution of population.

Various time series methods for inferring baseline values from historical data are considered in Chapter 4.

Two typical approaches to obtaining baselines are the *population-based method*, where we expect each count to be *proportional* to its baseline under the null hypothesis, and the *expectation-based method*, where we expect each count to be *equal* to its baseline under the null hypothesis. As we discuss in Section 2.3, these two approaches require the use of different models and statistics, and give different results under certain circumstances; here we focus on some possible ways of obtaining baseline values for each approach.

In the population-based method, baselines b_i typically represent the underlying *population* of location s_i. These populations could be obtained from census data, and may be adjusted for known covariates to give an "at-risk" population. For example, Kleinman et al. use a generalized linear mixed models approach to adjust population for day of week, seasonality, and other factors [76, 75]. Another possibility is to derive population estimates by measuring the value of some other "baseline" quantity, which we expect to be proportional to population regardless of whether the null hypothesis is true. One example would be the sales of a product such as soda or bottled water. These "activity-based" estimates of population have the disadvantages of higher variability and more noise, but can deal with more rapid or short-term changes in population and availability (e.g. for seasonal tourist destinations such as beach or ski resorts).

In the expectation-based method, each baseline b_i typically represents the *expected count* of location s_i under the null hypothesis of no clusters. These expected values are often derived from the time series of historical data, forecasting the expected value of the current data using some method of time series analysis. Another possibility is to obtain the expected count by monitoring some "control" condition, which we expect to be generated from the same distribution under the null. In brain imaging, for example, we can use subjects fixating on a cursor as a control condition, comparing this to an experimental condition where subjects read words or view pictures. A third option is to obtain expected counts using a combination of some measure of population and a constant of proportionality; for example, the population could be derived from census or activity-based estimates, while the constant of proportionality could be derived from global historical data.

In Section 2.3, we discuss the population-based and expectation-based approaches in more detail, and derive the appropriate models and score functions for each approach.

2.2.2 Choose a set of spatial regions to search over, where each spatial region S consists of a set of spatial locations s_i

We want to choose a set of regions that corresponds well with the shape and size of the clusters we are interested in detecting. In general, the set of regions should cover the entire space under consideration (otherwise we will have no power to detect clusters in non-covered areas) and adjacent regions should overlap (otherwise we will have reduced power to detect clusters that lie partly in one region and partly in another). We typically consider the set of all regions of some fixed shape (e.g. circle, ellipse, rectangle), allowing the location and dimensions of that region to vary; what shape to choose depends on both statistical and computational considerations. If we search too few regions, we will have reduced power to detect clusters that do not closely match any of the regions searched; for example, if we search over square or circular regions, we will have low power to detect highly elongated clusters. On the other hand, if we search too many regions, our power to detect any particular subset of these regions is reduced because of multiple hypothesis testing. Additionally, the runtime of the algorithm is proportional to the number of regions searched, and thus choosing too large a set of regions will make the method computationally infeasible.

Our typical approach in epidemiological domains is to map the spatial locations to a grid, and search over the set of all rectangular regions on the grid. Additionally, non-axis-aligned rectangles can be detected by searching over multiple rotations of the data. The two main advantages of this approach are its ability to detect elongated clusters (this is important in epidemiology because disease clusters may be elongated due to wind or water dispersion of pathogens) and also its computational efficiency. Use of a grid structure allows us to evaluate any rectangular region in constant time, independent of the size of the region, using the well-known "cumulative counts" trick. Additionally, we can gain huge computational speedups by applying the "fast spatial scan" algorithm, allowing us to search many fewer regions without any loss of accuracy. Both the cumulative counts trick and the fast spatial scan algorithm are discussed in Chapter 3.

2.2.3 Choose models of the data under H_0 (the null hypothesis of no clusters) and $H_1(S)$ (the alternative hypothesis assuming a cluster in region S). Derive a "score function" $F(S)$ based on $H_1(S)$ and H_0

These are perhaps the most difficult steps in our method, as we must choose models which are both efficiently computable and relevant to the application domain under consideration. For our models to be efficiently computable, the score function $F(S)$ should be computable as a function of some additive sufficient statistics of the region S being considered.[2] Typically these statistics are the total count of the region, $C(S) = \sum_S c_i$, and the total baseline of the region, $B(S) = \sum_S b_i$. If this is not the case, the model may still be useful for small datasets, but will not scale well to larger sources of data. For our models to be relevant, any simplifying assumptions that we make must not reduce our power to distinguish between the "cluster" and "no cluster" cases, to too great an extent. Of course, any efficiently computable model is very unlikely to capture all of the complexity of the real data, and these unmodeled effects may have either small or large impacts on detection performance. Thus we typically use an iterative design process, beginning with very simple models, and examining their detection power (ability to distinguish between "cluster" and "no cluster") and calibration (number of false positives reported in day-to-day use). If a model has high detection power but poor calibration, then we have a choice between increasing model complexity and artificially recalibrating the model (i.e. based on the empirical distribution of scores); however, if detection power is low, then we have no choice but to figure out which unmodeled effects are harming performance, and deal with these effects one by one. Some such effects (e.g. missing data) can be dealt with by pre-processing, and others (e.g. clusters caused by single anomalous locations) can be dealt with by post-processing (filtering the set of discovered regions to remove those caused by known effects), while others (such as overdispersion and correlation of counts) must actually be included in the model itself. In Chapter 6, we discuss several of these effects present in the over-the-counter sales data, and how we have dealt with each; here we focus on the general framework and then present two simple and efficiently computable approaches.

As noted above, we must choose models of how the data is generated, both under the null hypothesis H_0 (assuming that no clusters are present) and under the set of alternative hypotheses $H_1(S)$, each representing a cluster in some region S. Once we have chosen these models, we must make two choices regarding how to derive the corresponding statistics: whether to use a *frequentist*

[2] More precisely, we must have $F(S) = F(X_1(S) \ldots X_n(S))$, where each $X_j(S) = \sum_{s_i \in S} f(c_i, b_i)$.

or *Bayesian* approach, and whether to use *maximum likelihood* or *marginal likelihood* parameter estimates.

The most common statistical framework for the spatial scan is a frequentist, hypothesis testing approach. In this approach, assuming that the null hypothesis and each alternative hypothesis are point hypotheses (with no free parameters), we can use the likelihood ratio $F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$ as our test statistic. This likelihood ratio statistic represents the likelihood of the data assuming a cluster in region S, divided by the likelihood of the data assuming no clusters. A more interesting question is what to do when each hypothesis has some parameter space Θ: let $\theta_1(S) \in \Theta_1(S)$ denote parameters for the alternative hypothesis $H_1(S)$, and let $\theta_0 \in \Theta_0$ denote parameters for the null hypothesis H_0. There are two possible answers to this question. In the more typical, *maximum likelihood* framework, we use the estimates of each set of parameters that maximize the likelihood of the data:

$$F(S) = \frac{\max_{\theta_1(S) \in \Theta_1(S)} \Pr(\text{Data} \mid H_1(S), \theta_1(S))}{\max_{\theta_0 \in \Theta_0} \Pr(\text{Data} \mid H_0, \theta_0)}$$

In many cases, such as in Kulldorff's statistic [78], this will lead to an individually most powerful statistical test under the given model assumptions. We then perform randomization testing using the maximum likelihood estimates of the parameters under the null hypothesis, $\theta_{rep} = \arg\max_{\theta_0 \in \Theta_0} \Pr(\text{Data} \mid H_0, \theta_0)$, as discussed below. In the *marginal likelihood* framework, on the other hand, we instead average over the possible values of each parameter:

$$F(S) = \frac{\int_{\theta_1(S) \in \Theta_1(S)} \Pr(\text{Data} \mid H_1(S), \theta_1(S)) \Pr(\theta_1(S))}{\int_{\theta_0 \in \Theta_0} \Pr(\text{Data} \mid H_0, \theta_0) \Pr(\theta_0)}$$

This, however, makes randomization testing very difficult in the frequentist approach. An alternative method (discussed in detail in Chapter 5) is a Bayesian approach, in which we use the marginal likelihood framework to compute the likelihood of the data under each hypothesis, then combine these likelihoods with the prior probabilities of an cluster in each region S. Thus our test statistic is the posterior probability of a cluster in each region:

$$F(S) = \frac{\Pr(\text{Data} \mid H_1(S)) \Pr(H_1(S))}{\Pr(\text{Data})} \propto \Pr(\text{Data} \mid H_1(S)) \Pr(H_1(S))$$

where $\Pr(H_1(S))$ is the *prior probability* of a cluster in region S, and $\Pr(\text{Data} \mid H_1(S))$ is the data likelihood assuming a cluster in S. The marginal likelihood of the data is typically difficult to compute, but in Chapter 5, we present an efficiently computable Bayesian statistic using Poisson counts and conjugate Gamma priors.

Thus we now have two efficiently computable approaches within our general framework: the frequentist approach (using the likelihood ratio statistic with maximum likelihood parameter estimates, and computing statistical significance by randomization), and the Bayesian approach (using marginal likelihood). In Sections 2.3 and 2.4, we focus on the frequentist approach in more detail, and give examples of how new and useful scan statistics can be derived. More examples of developing and applying new scan statistics within this framework are given in the discussion on space-time statistics in Chapter 4.

2.2.4 Find the "most interesting" regions, i.e. those regions S with the highest values of $F(S)$

Once we have decided on a set of regions S to search, and derived a score function $F(S)$, the "most interesting" regions are those that maximize $F(S)$. In the frequentist spatial scan framework, these are the most significant spatial regions; in the Bayesian framework, these are the regions with highest posterior probabilities. The simplest method of finding the most interesting regions is to compute the score function $F(S)$ for every region. An alternative to this naïve approach is to use the fast spatial scan algorithms discussed in Chapter 3, which allow us to reduce the number of regions searched, but without losing any accuracy. The idea is that, since we only care about the most significant regions, i.e. those with the highest scores $F(S)$, we do not need to search a region S if we can prove that it will not have a high score. Thus we start by examining large regions S, and if we can show that none of the smaller regions contained in S can have high scores, we do not need to actually search each of these regions. Thus, we can achieve the same result as if we had searched all possible regions, but by only searching a small fraction of these. Further speedups are gained by the use of multiresolution data structures, which allow us to efficiently move between searching at coarse and fine resolutions; we discuss these methods in detail in Chapter 3.

2.2.5 Determine whether each of these regions is "interesting," either by performing significance testing or calculating posterior probabilities

For the frequentist approach, once we have found the highest scoring regions S, we must calculate the statistical significance of each discovered region by *randomization testing*. As discussed in Chapter 1, our goal is to perform statistical significance testing in such a way that, if the dataset has been generated under the null hypothesis (i.e. there are no clusters present), our probability of incorrectly detecting any clusters is bounded by some constant α, regardless of the number of regions tested. In other words, a region would be significant at $\alpha = .05$ only if its score is so high that 95% of the time under the null hypothesis, *no* region would have that high a score. In order to bound the overall false positive rate in this way, we randomly create a large number R of replica datasets by sampling under the null hypothesis H_0, given our maximum likelihood parameter estimates $\theta_{rep} = \arg\max_{\theta_0} \Pr(\text{Data} \mid H_0, \theta_0)$ for the null. For example, for Kulldorff's scan statistic, we generate counts independently from $c_i \sim \text{Poisson}(q_{all} b_i)$, using the maximum likelihood estimate $q_{all} = \frac{C_{all}}{B_{all}}$ from the original dataset. We then calculate the maximum region score $F^* = \max_S F(S)$ for each replica dataset. Now, for each potential cluster S, we count the number of replica datasets R_{beat} with F^* higher than $F(S)$. From this, we can calculate the p-value of region S as $p(S) = \frac{R_{beat}+1}{R+1}$. Then all regions S with $p(S) < \alpha$ are significant at level α, while all other regions are not significant. Since, for a given dataset, the p-value of region S decreases monotonically with increasing score $F(S)$, we can start by testing only the highest scoring region S^* of the original dataset. If the p-value of S^* is less than α, we can conclude that the discovered region is unlikely to have occurred by chance, and is thus a significant spatial cluster; we can then examine secondary clusters. Otherwise, no significant clusters exist.

For the Bayesian approach, on the other hand, no randomization testing is necessary. Instead, we can compute the posterior probability of each potential cluster by dividing its score $\Pr(\text{Data} \mid H_1(S))\Pr(H_1(S))$ by the total probability of the data, $\Pr(\text{Data}) = \Pr(\text{Data} \mid H_0)\Pr(H_0) + \sum_S \Pr(\text{Data} \mid H_1(S))\Pr(H_1(S))$. We can then report all clusters with posterior probability greater than some predetermined threshold P_{thresh}, or simply "sound the alarm" if the total posterior probability of all clusters S is sufficiently high (greater than some threshold P_{alarm}). Because we do not need to perform randomization testing in the Bayesian method, we need only to search over all regions for the original dataset, rather than the original dataset and a large number (typically $R = 999$) of replicas. Thus the Bayesian approach is approximately 1000x faster than the (naïve) frequentist approach, as we show empirically in Chapter 5. However, we can apply the fast spatial scan described above to achieve similar speedups for the frequentist approach: in this case, we still have to search over all replica datasets, but can do a much faster search on each. We compare the speed of the fast frequentist and Bayesian approaches in detail in Chapter 5, and consider how these two approaches might be combined to achieve real-time spatial cluster detection in Chapter 8.

2.3 Some simple scan statistics

We now derive scan statistics for four different models, including Kulldorff's original scan statistic, using the general framework discussed above. In each case, the derived score function is obtained using the likelihood ratio $F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$, with maximum likelihood estimates of any free parameters. All four of these models make several simplifying assumptions, including independently distributed counts, uniform rates under the null, and a uniform cluster model. These assumptions enable us to derive score functions $F(S)$ that are efficiently computable as a function of some sufficient statistics of region S. The disadvantage, however, is that violations of these assumptions will negatively impact our ability to detect clusters. As noted above, more complex models may be necessary in such cases; some examples of these models are given in Section 2.4.

We now consider each of the three simplifying assumptions in more detail. First, we assume that each location's count c_i is drawn independently from some distribution $\text{Dist}(\theta_i, q_i)$, where θ_i represents the set of baseline parameters of that location, and q_i represents some underlying "rate" parameter. This assumption is violated when counts are spatially correlated; one possible method of accounting for these correlations is the *region-aggregated time series* (RATS) approach discussed in Chapter 4. Second, we make the assumption that the rate parameter q_i is uniform under the null hypothesis: if no clusters are present, then every location has the same rate q. Thus we assume that any spatial variation in counts under the null (e.g. due to different underlying populations) is accounted for by our baseline parameters θ_i, and our methods are designed to detect any additional variation not reflected in these baselines. One difficulty with this is that we may pick up variations that are statistically significant but not large enough to be interesting in practice. The *thresholded scan statistics* discussed in Section 2.4, and the *T-filter* discussed in Chapter 6, are two possible methods of dealing with this problem. Another difficulty is that we may pick up variations caused by data irregularities in some single location (or a few locations). Methods of dealing with these irregularities include the *Bernoulli-Poisson scan statistic* discussed in Section 2.4, and the *L-filter* discussed in Chapter 6. Our third assumption is a uniform cluster model, where the effect of a cluster is to uniformly increase the expected counts within that cluster by some multiplicative constant (the amount of increase is unknown). We have considered several models that allow for spatial and temporal variation in rate: two such models are the *non-parametric scan statistic*, discussed in Section 2.4, and the *emerging cluster scan statistic*, discussed in Chapter 4.

In the remainder of this section, we make the three simplifying model assumptions discussed above, and use these to derive simple and efficiently computable statistics. In Subsections 2.3.1 and 2.3.2, we consider two decisions that must be made when choosing a model: whether to use

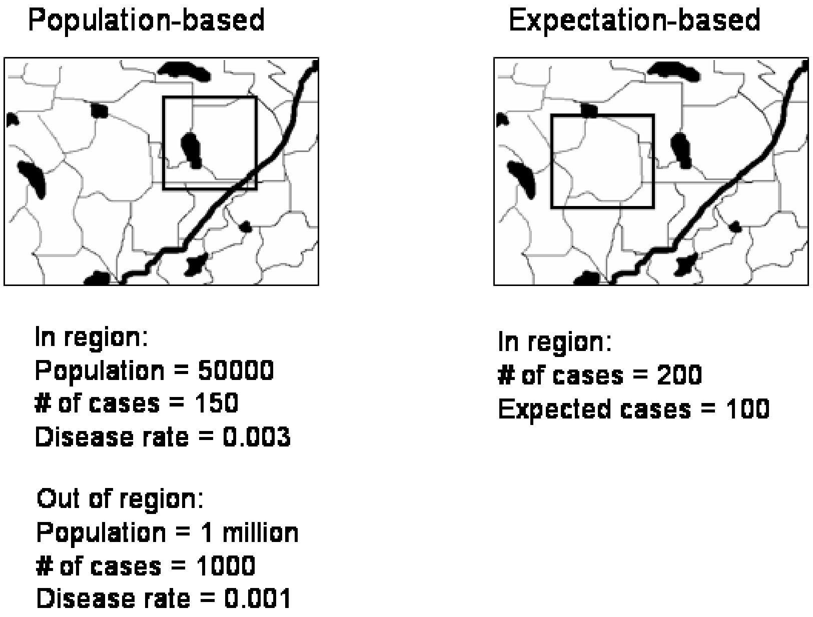


Figure 2.1: Population-based and expectation-based scan statistic approaches.

an "expectation-based" or "population-based" approach, and what distribution to assume. In Subsections 2.3.3 and 2.3.4, we derive the expectation-based and population-based statistics under the typical assumption of Poisson-distributed counts. Finally, in Subsections 2.3.5 and 2.3.6, we derive the expectation-based and population-based statistics for Gaussian-distributed counts, allowing us to model counts that can be overdispersed or underdispersed.

2.3.1 The expectation-based and population-based approaches

The spatial cluster detection task requires us to answer two main questions: is anything unexpected going on, and if so, where? In order to both discover unexpected clusters and infer their locations, we must first have some information about what we expect to see: this information is represented by the baseline b_i of each spatial location s_i. As discussed in Section 2.2, we may obtain baselines from a variety of sources, including census population data, historical counts, or data from a control group. The most important distinction we must draw is between two ways of interpreting these baselines: the *population-based* approach, where we expect counts to be *proportional* to baselines under the null hypothesis of no clusters, and the *expectation-based* approach, where we expect counts to be *equal* to baselines under the null. These two approaches are illustrated in Figure 2.1. For both approaches, we typically assume that each count c_i is generated from some distribution with mean equal to b_i times some unknown "rate" parameter q_i, but the interpretations of the baselines b_i and rates q_i are very different in these two approaches.

In the population-based approach, the baselines b_i typically represent the *population* corresponding to each spatial location s_i. This population can be either given (e.g. from census data) or inferred (e.g. from sales data), and can be adjusted for any known covariates such as age of population, risk factors, seasonality, and weather effects. The corresponding q_i represents the "underlying rate," or expected ratio of count to baseline, for that location. For example, in disease surveillance, it is common to speak of the "underlying disease rate," or expected number of disease cases per unit population. Note that the underlying rate q_i is an unknown quantity which is distinct from the (known) "observed rate" $\frac{c_i}{b_i}$, but we can use the observed rates to make statistical inferences about the underlying rates. In the population-based approach, we wish to detect clusters S where the observed rates are *significantly* higher inside S than outside S, allowing us to conclude with high probability that the underlying rates q_i are higher inside S than outside S. Thus, for each region S with an observed rate that is higher inside the region than outside the region, we must perform statistical testing to decide between two possible explanations: either a) the underlying rate is higher inside than outside, or b) the underlying rate is the same inside and outside, and the difference is due to chance. In the first case, S is a significant cluster, while in the second case, S is not significant. More precisely, under the simplifying assumption of uniform rates, we wish to test the null hypothesis that the rate is uniform everywhere (all q_i are equal to some constant q_{all}) against the set of alternative hypotheses with $q_i = q_{in}$ inside some region S and $q_i = q_{out}$ outside S, for some constants $q_{in} > q_{out}$.

In the expectation-based approach, the baselines b_i represent the *expected count* in each spatial location s_i. These are typically inferred from the time series of previous counts, adjusting for any relevant effects such as day-of-week and seasonality. The corresponding q_i represents the underlying "relative risk," or ratio of actual count to expected count. Our goal, then, is to discover regions with actual counts significantly greater than expected counts, or equivalently, observed relative risk significantly greater than 1. Again, we must distinguish between significant clusters (where the observed relative risk is large enough to conclude that the underlying relative risk is greater than 1) and non-significant regions (where we conclude that the underlying relative risk equals 1, and the higher-than-expected counts are due to chance). More precisely, under the simplifying assumption of uniform rates, we wish to test the null hypothesis that $q_i = 1$ everywhere against the set of alternative hypotheses with $q_i = q_{in}$ inside S and $q_i = 1$ outside S, for some constant $q_{in} > 1$.

Whether to use an expectation-based approach or a population-based approach depends both on the type and quality of data, as well as the types of clusters we are interested in detecting. As noted above, the expectation-based approach should be used when we can accurately estimate the expected count in each spatial location, either based on a sufficient amount of historical data, or based on sufficient data from a null or control condition; in these cases, expectation-based statistics will have higher detection power than population-based statistics. On the other hand, if we only have *relative* (rather than absolute) information about what we expect to see, a population-based approach should be used. For example, we may expect twice as many counts in location A as in location B, but we may not know exactly what to expect in either location: in this case, population-based statistics are appropriate. Similarly, if we have some historical data, but not enough to accurately estimate global trends (such as seasonal and day of week effects), it might be better to use a population-based statistic since this approach is more robust to misestimation of the global expectation.

The expectation-based and population-based approaches also give very different results in two important scenarios. First, if counts throughout the entire search region are much higher than expected, the expectation-based approach will find these increases very significant. However, the population-based approach will only find the increases significant if there is spatial variation in the amount of increase: otherwise, no significant increase will be detected. As an extreme example, consider a situation where every count is ten times its expected value: while the expectation-based approach would "sound the alarm" in response to this surprising data, the population-based approach would entirely ignore the increase (since the *ratio* of counts inside and outside any subregion of the search area has remained constant). More commonly, the population-based approach has somewhat reduced power for detecting clusters with large spatial extent. For example, if half of the search region has a 20% increase in counts, this potential cluster would be compared to the null hypothesis of a 10% increase in counts over the entire search region, and thus it appears to be a much smaller (and potentially non-significant) increase. Whether to use the expectation-based or population-based approach in this scenario depends on how we interpret the case of a global increase: if we assume that such increases have resulted from large clusters (and are therefore relevant to detect), the expectation-based approach should be used, but if we assume that such increases have resulted from unmodeled and irrelevant global trends (and should therefore be ignored), then it is more appropriate to use the population-based approach.

A second scenario where the two approaches differ is when the counts in one area are much lower than expected, and the other counts are normal. The expectation-based approach would not trigger an alarm in response to this situation, since no region counts are significantly higher than expected. The population-based approach, on the other hand, would trigger an alert in the "normal" counts because they are significantly higher (as compared to their underlying baselines) than the other "low" counts. Thus in public health data, the population-based approach may trigger false alarms in response to holiday effects that cause decreased counts (e.g. lower sales of over-the-counter drugs) in a subset of the spatial locations. Again, whether to use the expectation-based or population-based approach in this scenario depends on what types of clusters are considered interesting. If lower-than-expected counts in an area are assumed to be due to irrelevant factors that do not affect our expectations of the other counts, the expectation-based approach should be used, and if these decreases are assumed to be global trends that lower our expectations elsewhere, the population-based approach is more appropriate.

2.3.2 The Poisson and Gaussian models

In addition to choosing between expectation-based and population-based approaches, we must also choose a model of how the counts c_i are generated. In the public health domain, the most common model is Poisson-distributed counts: we assume that each count (i.e. number of disease cases) c_i has been drawn independently from a Poisson distribution with some (unknown) mean μ_i. This distribution has been justified in several ways: as a discretization of a Poisson process (assuming constant rate in time and/or space), as an approximation to the binomial (where each person becomes sick with some probability p) for large population and low disease rate, or as an improvement over the binomial for cases where each individual can be counted more than once (e.g. individuals can visit the emergency room multiple times, or buy multiple units of medication). In general, the Poisson model is appropriate for integer counts, assuming that the variance of the distribution is equal to the mean. If counts are overdispersed (variance higher than mean) or underdispersed (variance lower than mean), a different distribution should be used. Negative binomial distributions can be used to model overdispersed counts, while more complex distributions such as the Conway-Maxwell-Poisson [138] can be used to model counts which may be either overdispersed or underdispersed. Since both of these distributions are more difficult to work with, we typically use the Gaussian distribution as an approximation when working with overdispersed or underdispersed counts.

In the brain imaging domain, the most common model is Gaussian-distributed counts: we assume that each count (e.g. measured fMRI activation in the given region of the brain) has been drawn independently from a Gaussian distribution with some (unknown) mean μ_i and standard deviation σ_i. The Gaussian distribution has been justified in many contexts, since the sum of a large number of i.i.d. random variables converges to a Gaussian. Moreover, in the brain imaging domain, usually preprocessing is done, and this preprocessing makes the data more Gaussian. Sometimes the independence assumption is dropped, and this leads to the *Gaussian random field* approaches discussed in Chapter 7. In general, the Gaussian distribution is appropriate for real-valued counts when distributions are not skewed and have normal kurtosis (e.g. are not heavy-tailed). It is especially useful (as compared to the Poisson) when counts may be significantly overdispersed or underdispersed. When using the Gaussian distribution for integer-valued counts (e.g. as the approximation to a discrete distribution such as the Poisson or negative binomial), the Gaussian is generally a close approximation when counts are sufficiently large.

For our simple models, we typically use either the Poisson or Gaussian distributions, because each of these leads to efficiently computable score functions $F(S)$. In some cases, however, neither of these models may be adequate, and we may wish to sacrifice some computational efficiency for a more accurate and representative model of the data. In these cases, we can use models such as the negative binomial for overdispersed integer counts, the Bernoulli-Poisson model for data with irregularities at some individual locations, and the non-parametric scan statistic for data that is not fit adequately by any known model. The Bernoulli-Poisson and non-parametric scan statistics are discussed in Section 2.4.

For both the Poisson and Gaussian models, we typically assume that the mean of each distribution is proportional to some known baseline b_i, multiplied by an unknown rate parameter q_i, and then we use the observed counts to perform inference on the q_i. In the expectation-based statistic, we assume $q_i = 1$ (count equal to expectation) everywhere under the null, and $q_i > 1$ (count greater than expectation) in the affected region under the alternative hypothesis. In the population-based statistic, we assume q_i to be uniform everywhere under the null, and greater inside the affected region than outside under the alternative hypothesis.

2.3.3 Derivation of the Poisson expectation-based statistic

Let us first consider the simple expectation-based scan statistic discussed above, under the assumption that counts are independently Poisson distributed. In this case, we are given the baseline (or expected count) b_i and the observed count c_i for each spatial location s_i, and our goal is to determine if any spatial region S has counts significantly greater than baselines. Another way of asking this question is, if each count c_i has been drawn from a Poisson distribution with mean proportional to the expectation b_i times the “relative risk” q, is there any region with relative risk greater than 1? Thus we test the null hypothesis H_0 against the set of alternative hypotheses $H_1(S)$, where:

H_0: $c_i \sim \text{Poisson}(b_i)$ for all spatial locations s_i.
$H_1(S)$: $c_i \sim \text{Poisson}(qb_i)$ for all spatial locations s_i in S, and $c_i \sim \text{Poisson}(b_i)$ for all spatial locations s_i outside S, for some constant $q > 1$.

Here, the alternative hypothesis $H_1(S)$ has one parameter, q (the relative risk in region S), and the null hypothesis H_0 has no parameters. Computing the likelihood ratio, and using the maximum likelihood estimate for our parameter q, we obtain the following expression:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \text{Pr}(c_i \sim \text{Poisson}(qb_i)) \prod_{s_i \notin S} \text{Pr}(c_i \sim \text{Poisson}(b_i))}{\prod_{s_i} \text{Pr}(c_i \sim \text{Poisson}(b_i))}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(qb_i))}{\prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(b_i))}$$

Plugging in the equations for the Poisson likelihood, and simplifying, we obtain:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \frac{e^{-qb_i}(qb_i)^{c_i}}{(c_i)!}}{\prod_{s_i \in S} \frac{e^{-b_i}(b_i)^{c_i}}{(c_i)!}}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} e^{-qb_i} q^{c_i}}{\prod_{s_i \in S} e^{-b_i}}$$

$$= \frac{\max_{q>1} e^{-qB} q^C}{e^{-B}}$$

where C and B are the total count $\sum c_i$ and total baseline $\sum b_i$ of region S respectively. We find that the value of q that maximizes the numerator is $q = \max(1, \frac{C}{B})$. Plugging in this value of q, we obtain $F(S) = \left(\frac{C}{B}\right)^C e^{B-C}$, if $C > B$, and $F(S) = 1$ otherwise. Because $F(S)$ is a function only of the sufficient statistics $C(S)$ and $B(S)$, this function is efficiently computable: we can calculate the score of any region S by first calculating the aggregate count and baseline and then applying the function F. This approach can easily be extended to the case where counts are generated from $c_i \sim \text{Poisson}(q_0 b_i)$, for a known constant q_0, under the null hypothesis. In this case, we have $F(S) = \left(\frac{C}{q_0 B}\right)^C e^{q_0 B - C}$ if $C > q_0 B$, and $F(S) = 1$ otherwise.

As noted above, we can find the most significant spatial cluster by finding the region which maximizes $F(S)$. We can then perform statistical significance testing by randomization as discussed above, where each replica dataset has all counts generated under the null hypothesis $c_i \sim \text{Poisson}(b_i)$, or $c_i \sim \text{Poisson}(q_0 b_i)$ in the more general case.

2.3.4 Derivation of the Poisson population-based statistic

Next we consider the derivation of Kulldorff's spatial scan statistic [78]. As discussed in Chapter 1, this is a population-based method commonly used in disease surveillance, which also makes the simplifying assumption of independent, Poisson distributed counts. However, Kulldorff's statistic assumes that the counts (i.e. number of disease cases) are distributed as $c_i \sim \text{Poisson}(qb_i)$, where b_i is the (known) census population of s_i and q is the (unknown) underlying disease rate. We then attempt to discover spatial regions where the underlying disease rate q is significantly higher inside the region than outside. Thus we wish to test the null hypothesis H_0 ("the underlying disease rate is spatially uniform") against the set of alternative hypotheses $H_1(S)$: "the underlying disease rate is higher inside region S than outside S." More precisely, we have the following:

H_0: $c_i \sim \text{Poisson}(q_{all} b_i)$ for all locations s_i, for some constant q_{all}.
$H_1(S)$: $c_i \sim \text{Poisson}(q_{in} b_i)$ for all locations s_i in S, and $c_i \sim \text{Poisson}(q_{out} b_i)$ for all locations s_i outside S, for some constants $q_{in} > q_{out}$.

In this case, the alternative hypothesis has two free parameters (q_{in} and q_{out}) and the null hypothesis has one free parameter (q_{all}). Computing the likelihood ratio, and using maximum likelihood parameter estimates, we obtain:

$$F(S) = \frac{\max_{q_{in} > q_{out}} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(q_{in} b_i)) \prod_{s_i \notin S} \Pr(c_i \sim \text{Poisson}(q_{out} b_i))}{\max_{q_{all}} \prod_{s_i} \Pr(c_i \sim \text{Poisson}(q_{all} b_i))}$$

Plugging in the equations for the Poisson likelihood, and simplifying, we obtain:

$$F(S) = \frac{\max_{q_{in} > q_{out}} \prod_{s_i \in S} \frac{e^{-q_{in} b_i} (q_{in} b_i)^{c_i}}{(c_i)!} \prod_{s_i \notin S} \frac{e^{-q_{out} b_i} (q_{out} b_i)^{c_i}}{(c_i)!}}{\max_{q_{all}} \prod_{s_i} \frac{e^{-q_{all} b_i} (q_{all} b_i)^{c_i}}{(c_i)!}}$$

$$= \frac{\max_{q_{in} > q_{out}} \prod_{s_i \in S} e^{-q_{in} b_i} (q_{in})^{c_i} \prod_{s_i \notin S} e^{-q_{out} b_i} (q_{out})^{c_i}}{\max_{q_{all}} \prod_{s_i} e^{-q_{all} b_i} (q_{all})^{c_i}}$$

$$= \frac{\max_{q_{in} > q_{out}} e^{-q_{in} B_{in}} (q_{in})^{C_{in}} e^{-q_{out} B_{out}} (q_{out})^{C_{out}}}{\max_{q_{all}} e^{-q_{all} B_{all}} (q_{all})^{C_{all}}}$$

where C_{in} and B_{in} are the total count and baseline inside region S, C_{out} and B_{out} are the total count and baseline outside region S, and C_{all} and B_{all} are the total count and baseline everywhere. We can then compute the maximum likelihood estimates $q_{in} = \frac{C_{in}}{B_{in}}$, $q_{out} = \frac{C_{out}}{B_{out}}$, and $q_{all} = \frac{C_{all}}{B_{all}}$, if $\frac{C_{in}}{B_{in}} > \frac{C_{out}}{B_{out}}$, and $q_{in} = q_{out} = q_{all} = \frac{C_{all}}{B_{all}}$ otherwise. Plugging in these maximum likelihood values, we obtain: $F(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}}\right)^{-C_{all}}$, if $\frac{C_{in}}{B_{in}} > \frac{C_{out}}{B_{out}}$, and $F(S) = 1$ otherwise. Again, the score function can be computed efficiently, using the sufficient statistics of region S and the global sufficient statistics C_{all} and B_{all}.

As in the expectation-based approach, we can find the most significant spatial cluster by finding the region that maximizes $F(S)$, and perform statistical significance testing by randomization. In this case, however, each replica dataset has all counts generated under the null hypothesis $c_i \sim \text{Poisson}(q_{all} b_i)$, where we use the maximum likelihood estimate $q_{all} = \frac{C_{all}}{B_{all}}$ from the original dataset.

2.3.5 Derivation of the Gaussian expectation-based scan statistic

We now consider an expectation-based scan statistic with Gaussian-distributed counts. In this case, in addition to the observed counts c_i, we are given the expected count μ_i and the expected standard deviation σ_i for each spatial location s_i. Our goal, as before, is to determine if any spatial region S has counts significantly greater than baselines. Assuming that each count c_i has been drawn from a Gaussian distribution with mean proportional to μ_i times the relative risk q, and with standard deviation σ_i, we must determine whether any region has relative risk greater than 1. Thus we test the null hypothesis H_0 against the set of alternative hypotheses $H_1(S)$, where:

H_0: $c_i \sim \text{Gaussian}(\mu_i, \sigma_i)$ for all spatial locations s_i.
$H_1(S)$: $c_i \sim \text{Gaussian}(q\mu_i, \sigma_i)$ for all spatial locations s_i in S, and $c_i \sim \text{Gaussian}(\mu_i, \sigma_i)$ for all spatial locations s_i outside S, for some constant $q > 1$.

Notice that we have assumed that the variance of counts does not increase inside the cluster S; similar statistics can be easily derived for cases where the variance increases. As before, the alternative hypothesis $H_1(S)$ has one parameter, q (the relative risk in region S), and the null hypothesis

H_0 has no parameters. Computing the likelihood ratio, and using the maximum likelihood estimate for the parameter q, we obtain the following expression:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \Pr(c_i \sim \text{Gaussian}(q\mu_i, \sigma_i)) \prod_{s_i \notin S} \Pr(c_i \sim \text{Gaussian}(\mu_i, \sigma_i))}{\prod_{s_i} \Pr(c_i \sim \text{Gaussian}(\mu_i, \sigma_i))}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \Pr(c_i \sim \text{Gaussian}(q\mu_i, \sigma_i))}{\prod_{s_i \in S} \Pr(c_i \sim \text{Gaussian}(\mu_i, \sigma_i))}$$

Plugging in the equations for the Gaussian likelihood, and simplifying, we obtain:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(c_i - q\mu_i)^2}{2\sigma_i^2}}}{\prod_{s_i \in S} \frac{1}{\sigma_i \sqrt{2\pi}} e^{-\frac{(c_i - \mu_i)^2}{2\sigma_i^2}}}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} e^{-\frac{(c_i - q\mu_i)^2}{2\sigma_i^2}}}{\prod_{s_i \in S} e^{-\frac{(c_i - \mu_i)^2}{2\sigma_i^2}}}$$

$$= \max_{q>1} \prod_{s_i \in S} \exp\left(\frac{(c_i - \mu_i)^2 - (c_i - q\mu_i)^2}{2\sigma_i^2}\right)$$

$$= \max_{q>1} \exp\left(\frac{1-q^2}{2} \sum_{s_i \in S} \frac{\mu_i^2}{\sigma_i^2} + (q-1) \sum_{s_i \in S} \frac{c_i \mu_i}{\sigma_i^2}\right)$$

$$= \max_{q>1} \exp\left(\frac{1-q^2}{2} B' + (q-1)C'\right)$$

where $B' = \sum_{s_i \in S} \frac{\mu_i^2}{\sigma_i^2}$ and $C' = \sum_{s_i \in S} \frac{c_i \mu_i}{\sigma_i^2}$. These sufficient statistics $B'(S)$ and $C'(S)$ can be interpreted as weighted sums of the expectations μ_i and counts c_i respectively, where the weighting of a location s_i is inversely proportional to the coefficient of variation $\frac{\sigma_i^2}{\mu_i}$. We find that the maximum likelihood value of q is $q = \max(1, \frac{C'}{B'})$. Plugging in this value of q, we obtain $F(S) = \exp\left(\frac{(C')^2}{2B'} + \frac{B'}{2} - C'\right)$, if $C' > B'$, and $F(S) = 1$ otherwise. Because $F(S)$ is a function only of the sufficient statistics $C'(S)$ and $B'(S)$, we again have an efficiently computable score function.

As above, we can find the most significant spatial cluster S^* by maximizing $F(S)$, and perform statistical significance testing by randomization. To do so, we generate each replica dataset by drawing all counts from the null hypothesis $c_i \sim \text{Gaussian}(\mu_i, \sigma_i)$.

2.3.6 Derivation of the Gaussian population-based scan statistic

Finally, we consider a population-based scan statistic with Gaussian-distributed counts. Again, we are given the observed count c_i, expected count μ_i, and expected standard deviation σ_i for each location s_i. But now we assume that the counts are distributed as $c_i \sim \text{Gaussian}(q\mu_i, \sigma_i)$, where q is the underlying rate, and search for regions with significantly higher rate inside than outside. Thus we test the null hypothesis H_0 against the set of alternative hypotheses $H_1(S)$, where:

H_0: $c_i \sim$ Gaussian$(q_{all}\mu_i, \sigma_i)$ for all spatial locations s_i, for some constant q_{all}.
$H_1(S)$: $c_i \sim$ Gaussian$(q_{in}\mu_i, \sigma_i)$ for all spatial locations s_i in S, and $c_i \sim$ Gaussian$(q_{out}\mu_i, \sigma_i)$ for all spatial locations s_i outside S, for some constants $q_{in} > q_{out}$.

Computing the likelihood ratio, and using the maximum likelihood estimates of the rate parameters q_{in}, q_{out}, and q_{all}, we obtain the following expression:

$$F(S) = \frac{\max_{q_{in}>q_{out}} \prod_{s_i \in S} \Pr(c_i \sim \text{Gaussian}(q_{in}\mu_i, \sigma_i)) \prod_{s_i \notin S} \Pr(c_i \sim \text{Gaussian}(q_{out}\mu_i, \sigma_i))}{\max_{q_{all}} \prod_{s_i} \Pr(c_i \sim \text{Gaussian}(q_{all}\mu_i, \sigma_i))}$$

Plugging in the equations for the Gaussian likelihood, and simplifying, we obtain:

$$F(S) = \frac{\max_{q_{in}>q_{out}} \prod_{s_i \in S} \exp\left(-\frac{(c_i - q_{in}\mu_i)^2}{2\sigma_i^2}\right) \prod_{s_i \notin S} \exp\left(-\frac{(c_i - q_{out}\mu_i)^2}{2\sigma_i^2}\right)}{\max_{q_{all}} \prod_{s_i} \exp\left(-\frac{(c_i - q_{all}\mu_i)^2}{2\sigma_i^2}\right)}$$

We now let B'_{in}, B'_{out}, and B'_{all} represent the sums $\sum \frac{\mu_i^2}{\sigma_i^2}$ for s_i inside S, for s_i outside S, and for all s_i respectively. Similarly, we let C'_{in}, C'_{out}, and C'_{all} represent the sums $\sum \frac{c_i \mu_i}{\sigma_i^2}$ for s_i inside S, for s_i outside S, and for all s_i respectively. Then this expression simplifies to:

$$F(S) = \frac{\max_{q_{in}>q_{out}} \exp\left(-\frac{1}{2}q_{in}^2 B'_{in} + q_{in}C'_{in} - \frac{1}{2}q_{out}^2 B'_{out} + q_{out}C'_{out}\right)}{\max_{q_{all}} \exp\left(-\frac{1}{2}q_{all}^2 B'_{all} + q_{all}C'_{all}\right)}$$

We can then compute the maximum likelihood estimates $q_{in} = \frac{C'_{in}}{B'_{in}}$, $q_{out} = \frac{C'_{out}}{B'_{out}}$, and $q_{all} = \frac{C'_{all}}{B'_{all}}$, if $\frac{C'_{in}}{B'_{in}} > \frac{C'_{out}}{B'_{out}}$, and $q_{in} = q_{out} = q_{all} = \frac{C'_{all}}{B'_{all}}$ otherwise. Plugging in these maximum likelihood values, we obtain: $F(S) = \exp\left(\frac{(C'_{in})^2}{2B'_{in}} + \frac{(C'_{out})^2}{2B'_{out}} - \frac{(C'_{all})^2}{2B'_{all}}\right)$, if $\frac{C'_{in}}{B'_{in}} > \frac{C'_{out}}{B'_{out}}$, and $F(S) = 1$ otherwise. Again, $F(S)$ is efficiently computable as a function of the sufficient statistics of region S and the global sufficient statistics C'_{all} and B'_{all}.

As above, we can find the most significant spatial cluster S^* by maximizing $F(S)$, and perform statistical significance testing by randomization. To do so, we generate each replica dataset by drawing all counts from the null hypothesis $c_i \sim$ Gaussian$(q_{all}\mu_i, \sigma_i)$, where we use the maximum likelihood estimate $q_{all} = \frac{C'_{all}}{B'_{all}}$ from the original dataset.

2.4 More scan statistics

Many other likelihood ratio scan statistics are possible, including models with simultaneous attacks in multiple regions and models with spatially varying (rather than uniform) rates. We believe that some of these more complex model specifications may have more power to detect relevant and interesting clusters, while excluding those potential clusters which are not relevant to the application domain under consideration. In this section, we briefly discuss three such methods which may be useful for disease surveillance as well as other application domains. Each method deals with one confounding factor which the simple models do not account for: the Bernoulli-Poisson scan statistic is robust to outliers, the thresholded scan statistics are robust to small fluctuations in the underlying rate, and the non-parametric scan statistic is robust to skewed and heavy-tailed count distributions. The cost of this greater flexibility is less computational efficiency: many of these statistics cannot be expressed in terms of the sufficient statistics of a region. The main goal of this overview is to demonstrate the generality and flexibility of our statistical framework, and its ability to be adapted to domains where simpler models are inadequate. A more extensive exploration of these methods is beyond the scope of this thesis, but will be addressed in further work.

2.4.1 The Bernoulli-Poisson scan statistic

In some application domains, use of the simple scan statistic models discussed above results in a large number of false positives due to *outliers*, or individual spatial locations with counts that are much higher than expected. While in some cases we are interested in detecting such very localized increases, it is more often the case that these increases are due to irregularities in the data or other irrelevant causes. For example, in our monitoring of over-the-counter drug sales data, we often see stores with large spikes in sales on a given day, due to bulk purchases, inventory movements, promotional sales, or errors in data collection. We typically want to ignore these outliers (since they are not indicative of a disease outbreak), and only detect clusters that affect multiple locations in an area.

In the Bernoulli-Poisson scan statistic, we assume that all counts c_i are Poisson distributed with means $q_i b_i$, where b_i is the expected count and q_i is the relative risk at location s_i. The difference from the standard expectation-based Poisson scan statistic is that the q_i are drawn from a noisy distribution, with probability ϵ of being equal to some "outlier value" o_i. The value of ϵ must be specified as an input parameter ($0 < \epsilon < \frac{1}{2}$), while the o_i will be selected by maximum likelihood parameter estimation. Thus we compare the null hypothesis H_0 to the set of alternative hypotheses $H_1(S)$, where:

H_0: for all spatial locations s_i, $q_i = 1$ with probability $1 - \epsilon$, and $q_i = o_i$ with probability ϵ.
$H_1(S)$: for all spatial locations s_i in S, $q_i = q$ with probability $1 - \epsilon$, and $q_i = o_i$ with probability ϵ, for some constant $q > 1$. For all spatial locations s_i outside S, $q_i = 1$ with probability $1 - \epsilon$, and $q_i = o_i$ with probability ϵ.

We now compute the likelihood ratio statistic, using maximum likelihood estimates of all free parameters:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \max((1-\epsilon)\Pr(c_i \sim \text{Poisson}(qb_i)), \epsilon \max_{o_i} \Pr(c_i \sim \text{Poisson}(o_i b_i)))}{\prod_{s_i \in S} \max((1-\epsilon)\Pr(c_i \sim \text{Poisson}(b_i)), \epsilon \max_{o_i} \Pr(c_i \sim \text{Poisson}(o_i b_i)))}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \max((1-\epsilon)\Pr(c_i \sim \text{Poisson}(qb_i)), \epsilon \Pr(c_i \sim \text{Poisson}(c_i)))}{\prod_{s_i \in S} \max((1-\epsilon)\Pr(c_i \sim \text{Poisson}(b_i)), \epsilon \Pr(c_i \sim \text{Poisson}(c_i)))}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \max\left(\frac{(1-\epsilon)e^{-qb_i}(qb_i)^{c_i}}{(c_i)!}, \frac{\epsilon e^{-c_i}(c_i)^{c_i}}{(c_i)!}\right)}{\prod_{s_i \in S} \max\left(\frac{(1-\epsilon)e^{-b_i}(b_i)^{c_i}}{(c_i)!}, \frac{\epsilon e^{-c_i}(c_i)^{c_i}}{(c_i)!}\right)}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \max((1-\epsilon)e^{-qb_i}(qb_i)^{c_i}, \epsilon e^{-c_i}(c_i)^{c_i})}{\prod_{s_i \in S} \max((1-\epsilon)e^{-b_i}(b_i)^{c_i}, \epsilon e^{-c_i}(c_i)^{c_i})}$$

The denominator may be calculated easily by computing the maximum of the expressions $(1-\epsilon)e^{-b_i}(b_i)^{c_i}$ and $\epsilon e^{-c_i}(c_i)^{c_i}$ for each spatial location s_i. All locations where the latter expression is larger are outliers under the null hypothesis. The numerator is more difficult to calculate, as we must compute the maximum likelihood value of the relative risk q. To do so, we note that the function $f(q) = (1-\epsilon)e^{-qb_i}(qb_i)^{c_i}$ is concave downward with a maximum at $q = \frac{c_i}{b_i}$. Thus, for each s_i, we can compute the interval (q_{min}, q_{max}) such that $f(q) > \epsilon e^{-c_i}(c_i)^{c_i}$. We obtain $q_{min} = -\frac{c_i}{b_i} W\left(0, -\frac{1}{e}\left(\frac{\epsilon}{1-\epsilon}\right)^{\frac{1}{c_i}}\right)$, and $q_{max} = -\frac{c_i}{b_i} W\left(-1, -\frac{1}{e}\left(\frac{\epsilon}{1-\epsilon}\right)^{\frac{1}{c_i}}\right)$, where $W(\cdot)$ is Lambert's W function. We now form a single sorted list $Z = \langle z_j \rangle$ containing all distinct q_{min} and q_{max} values greater than 1. These represent all the distinct intervals we must consider for q: $q \in [1, z_1]$, $q \in [z_1, z_2]$, …, $q \in [z_{n-1}, z_n]$, $q \in [z_n, \infty]$. For each interval, we compute which locations are and are not outliers, then compute the optimal value of q for that interval: $q = \frac{C}{B}$ restricted to that interval, where C and B are respectively the total count $\sum c_i$ and total baseline $\sum b_i$ of all non-outliers. This allows us to find the optimal value of q for region S, the optimal set of outliers under the alternative hypothesis $H_1(S)$, and the score $F(S)$. As in the simple scan statistic models, the most significant region is the one which maximizes $F(S)$, and we can compute the significance of this region by randomization. We create replica datasets by sampling under the null hypothesis $c_i \sim$ Poisson$(q_i b_i)$, where $q_i = 1$ for all non-outliers and $q_i = \frac{c_i}{b_i}$ for all outliers under the null.

As an example of the computation of the Bernoulli-Poisson statistic, let us consider a region with five locations s_i, with (c_i, b_i) equal to $(12, 10)$, $(100, 3)$, $(9, 11)$, $(17, 10)$, and $(22, 10)$ respectively. Assuming $\epsilon = 0.01$, we find the intervals (q_{min}, q_{max}) equal to $(0.43, 2.58)$, $(24.2, 44.5)$, $(0.24, 1.94)$, $(0.74, 3.27)$, and $(1.07, 3.94)$ respectively. Thus we know that, under the null hypothesis $q = 1$, the second and fifth locations are outliers. To find the optimal value of q, we must search the intervals $[1, 1.07]$, $[1.07, 1.94]$, $[1.94, 2.58]$, $[2.58, 3.27]$, $[3.27, 3.94]$, $[3.94, 24.2]$, $[24.2, 44.5]$, and $[44.5, \infty]$. We obtain the optimal value in the interval $[1.07, 1.94]$, where only the second location is an outlier. In this case, we have $q = \frac{12+9+17+22}{10+11+10+10} = 1.46$, with the resulting score $F(S) = 22.1$. As the probability of outliers ϵ increases, more locations become outliers: for $\epsilon > .0825$, the third location is also an outlier under the alternative hypothesis. Similarly, as ϵ decreases, we have less outliers: the very anomalous location with $c_i = 100$ and $b_i = 3$ is only a non-outlier for $\epsilon < 3 \times 10^{-73}$. Finally, we note that in the limit of $\epsilon \to 0$, the score $F(S)$ converges to that of the simple expectation-based Poisson scan statistic.

2.4.2 Thresholded scan statistics

Another potential source of false positives when simple scan statistics are used is slight variations in the underlying rate parameter q. Our simple statistics attempt to detect any regions where the rate is higher than expected: in the expectation-based approach, these are any regions where the relative risk $q > 1$, and in the population-based approach, these are any regions where the rate is higher inside the region than outside ($q_{in} > q_{out}$). For example, even a 1% increase in rate will be detected if it corresponds to a large enough underlying population or baseline to make that increase significant. However, in many applications, we are only interested in regions where the rate is *substantially* increased, so these slight fluctuations in rate can be thought of as statistically but not practically significant. In practical applications such as disease surveillance, the simple scan statistics often detect slight increases in count corresponding to a large spatial region, but these regions are unlikely to be indicative of a "true cluster" (e.g. disease outbreak). It is more likely that the variations result from other, irrelevant factors such as model misspecification: we may have underestimated the baseline for the given region, or failed to account for some relevant covariates. No matter how complex our model, there will always be some aspects of the real-world data that we fail to account for, and we would like our statistics to be as robust to these as possible.

Our solution is to only detect clusters where the underlying rate is increased by more than some constant $\epsilon \geq 0$. We term such methods *thresholded scan statistics*, and ϵ the *detection threshold*. For the expectation-based statistic, we wish to detect regions where the relative risk $q > 1 + \epsilon$, and for the population-based statistic, we wish to detect regions where the rate $q_{in} > (1 + \epsilon)q_{out}$. For example, $\epsilon = 0.2$ would correspond to detecting regions with more than 20% increases in rate, while $\epsilon = 0$ is equivalent to the simple scan statistics discussed above. A thresholded scan statistic may be defined in a number of ways, depending on our answers to five distinct questions. First, as noted above, we can define either expectation-based or population-based statistics. Second, we must choose what sort of fluctuation in rates is allowable under the null hypothesis of no clusters. This requires us to answer three questions: do we allow rates to fluctuate between $1 - \epsilon$ and $1 + \epsilon$ or between 1 and $1 + \epsilon$, do we allow fluctuations everywhere or only in a single region, and must the amount of fluctuation be constant across locations or different for each location? Finally, we must choose what sort of increase is allowable under the alternative hypothesis $H_1(S)$: we can either assume a constant multiplicative increase, or a different amount of increase for each location in the region. Here we consider one such statistic, which is expectation-based, allows counts to fluctuate everywhere (and differently for each location) between 1 and $1 + \epsilon$, and assumes a constant multiplicative increase under $H_1(S)$. In this case, we compare the null hypothesis H_0 to the set of alternative hypotheses $H_1(S)$, where:

H_0: $c_i \sim$ Poisson$(\epsilon_i b_i)$ everywhere, where $1 \leq \epsilon_i \leq 1 + \epsilon$.
$H_1(S)$: $c_i \sim$ Poisson$(q\epsilon_i b_i)$ inside region S for some constant $q > 1$, and $c_i \sim$ Poisson$(\epsilon_i b_i)$ outside S, where $1 \leq \epsilon_i \leq 1 + \epsilon$.

For these models, we derive the following likelihood ratio statistic:

$$F(S) = \frac{\max_{q>1} \prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} \Pr(c_i \sim \text{Poisson}(q\epsilon_i b_i))}{\prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} \Pr(c_i \sim \text{Poisson}(\epsilon_i b_i))}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} \frac{e^{-q\epsilon_i b_i}(q\epsilon_i b_i)^{c_i}}{(c_i)!}}{\prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} \frac{e^{-\epsilon_i b_i}(\epsilon_i b_i)^{c_i}}{(c_i)!}}$$

$$= \frac{\max_{q>1} \prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} e^{-q\epsilon_i b_i}(q\epsilon_i)^{c_i}}{\prod_{s_i \in S} \max_{1 \leq \epsilon_i \leq 1+\epsilon} e^{-\epsilon_i b_i}(\epsilon_i)^{c_i}}$$

The denominator of this expression can be computed easily by noting that the optimal value of each ϵ_i is $\frac{c_i}{b_i}$ restricted to the interval $[1, 1 + \epsilon]$. The numerator is more difficult to calculate, as we must compute the maximum likelihood value of the parameter q. To do so, we note that each s_i has $\epsilon_i = \frac{c_i}{qb_i}$ restricted to $[1, 1+\epsilon]$. We will have $1 \leq \frac{c_i}{qb_i} \leq 1+\epsilon$ for the interval $q \in [q_{min}, q_{max}]$, where $q_{min} = \frac{c_i}{(1+\epsilon)b_i}$ and $q_{max} = \frac{c_i}{b_i}$. We now form a single sorted list $Z = \langle z_j \rangle$ containing all distinct q_{min} and q_{max} values greater than 1. These represent all the distinct intervals we must consider for q: $q \in [1, z_1]$, $q \in [z_1, z_2]$, …, $q \in [z_{n-1}, z_n]$, $q \in [z_n, \infty]$. For each interval $q \in [z_j, z_{j+1}]$, we can compute the optimal value of q by dividing the locations into three groups: locations with $\frac{c_i}{b_i} \leq z_j$, locations with $\frac{c_i}{b_i} \geq (1+\epsilon)z_{j+1}$, and locations with $z_j < \frac{c_i}{b_i} < (1+\epsilon)z_{j+1}$. For the first set of locations, the optimal value of ϵ_i will be 1 regardless of the value of q. For the second set of locations, the optimal value of ϵ_i will be $(1+\epsilon)$ regardless of the value of q. For the third set of locations, the optimal value of ϵ_i will be $\frac{c_i}{qb_i}$, and each location's contribution to the score is independent of q. Thus, for values of q restricted to $[z_j, z_{j+1}]$, we have:

$$\arg\max_q \prod \max_{1 \leq \epsilon_i \leq 1+\epsilon} e^{-q\epsilon_i b_i}(q\epsilon_i)^{c_i}$$

$$= \arg\max_q \left(\prod_{\frac{c_i}{b_i} \leq z_j} e^{-qb_i} q^{c_i} \right) \left(\prod_{\frac{c_i}{b_i} \geq (1+\epsilon)z_{j+1}} e^{-(1+\epsilon)qb_i}((1+\epsilon)q)^{c_i} \right) \left(\prod_{z_j < \frac{c_i}{b_i} < (1+\epsilon)z_{j+1}} e^{-c_i} \left(\frac{c_i}{b_i} \right)^{c_i} \right)$$

$$= \arg\max_q \left(\prod_{\frac{c_i}{b_i} \leq z_j} e^{-qb_i} q^{c_i} \right) \left(\prod_{\frac{c_i}{b_i} \geq (1+\epsilon)z_{j+1}} e^{-(1+\epsilon)qb_i} q^{c_i} \right)$$

$$= \arg\max_q e^{-q(B_1+(1+\epsilon)B_2)} q^{C_1+C_2} = \frac{C_1+C_2}{B_1+(1+\epsilon)B_2}$$

restricted to the interval $[z_j, z_{j+1}]$. In these equations, C_1 and B_1 are the total count $\sum c_i$ and total baseline $\sum b_i$ for all locations with $\frac{c_i}{b_i} \leq z_j$, and C_2 and B_2 are the total count $\sum c_i$ and total baseline $\sum b_i$ for all locations with $\frac{c_i}{b_i} \geq (1+\epsilon)z_{j+1}$. Given the optimal value of q for the interval, we compute the score $F(S)$ by setting all $\epsilon_i = \frac{c_i}{qb_i}$ restricted to $[1, 1+\epsilon]$. Finally, we choose the maximum score over all intervals to obtain the optimal value of $F(S)$.

As in the simple scan statistic models, the most significant region is the one which maximizes $F(S)$, and we can compute the significance of this region by randomization. We create replica datasets by sampling under the null hypothesis $c_i \sim \text{Poisson}(\epsilon_i b_i)$, where each ϵ_i is equal to $\frac{c_i}{b_i}$ restricted to the interval $[1, 1+\epsilon]$.

In previous work, we proposed a *discriminative* scan statistic model that computes the likelihood ratio of the alternative hypothesis $H_1(S)$ to the null hypothesis $H_0(S)$ for a given region S. One discriminative version of the thresholded scan statistic, which we used in [118], compares the null hypothesis $H_0(S)$: $q_{in} \leq (1+\epsilon)q_{out}$ to the alternative hypothesis $H_1(S)$: $q_{in} > (1+\epsilon)q_{out}$, where q_{in} and q_{out} are the underlying rates inside and outside region S respectively. Thus we have a thresholded statistic that is population-based, allows a constant rate increase in a single region under the null, and assumes a constant multiplicative increase under $H_1(S)$. While this statistic is somewhat different from our generalized scan statistic framework (which assumes a single composite null hypothesis H_0 rather than a separate null $H_0(S)$ for each region tested), the discriminative thresholded scan statistic is efficiently computable, and it was used successfully to detect clusters in multidimensional disease surveillance and brain imaging data in [118]. The thresholded scan statistic results presented in Chapters 3 and 7 rely on this version of the statistic; we also plan to compare these results to the other thresholded scan statistic models discussed above. For the discriminative thresholded scan statistic, we derive the likelihood ratio statistic as follows:

$$F(S) = \frac{\max_{q_{in} > (1+\epsilon)q_{out}} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(q_{in} b_i)) \prod_{s_i \in G-S} \Pr(c_i \sim \text{Poisson}(q_{out} b_i))}{\max_{q_{in} \leq (1+\epsilon)q_{out}} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(q_{in} b_i)) \prod_{s_i \in G-S} \Pr(c_i \sim \text{Poisson}(q_{out} b_i))}$$

$$= \left(\frac{C_{in}}{(1+\epsilon)B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}+\epsilon B_{in}}\right)^{-C_{all}}$$

if $\frac{C_{in}}{B_{in}} > (1+\epsilon)\frac{C_{out}}{B_{out}}$, where the counts and baselines are defined as in the Poisson population-based statistic. Then the most significant region can be obtained by maximizing $F(S)$, and we can compute statistical significance by randomization under the null hypothesis $H_0(S)$.

2.4.3 The non-parametric scan statistic

As discussed above, our expectation-based scan statistics attempt to model the expected distribution of counts for each spatial location under the null hypothesis of no clusters, then find regions where the counts are higher than expected. The simple expectation-based statistics assume that the counts are generated by some parametric model, then learn the parameters of this model, typically from historical data. For example, the Poisson expectation-based statistic learns the baseline (expected count) b_i for each spatial location, while the Gaussian expectation-based statistic learns both the expected count μ_i and the expected variance σ_i. The disadvantage of these model-based approaches is that they rely heavily on our distributional assumptions: for example, the Poisson statistic cannot account for overdispersion or underdispersion of counts, and neither Poisson nor Gaussian statistics can account for heavy-tailed count distributions.

Our solution is a *non-parametric scan statistic* approach, where we make no model assumptions on the distribution of counts, but instead use the empirical distribution of historical counts for each spatial location. Let us assume that we have a count c_i and a time series of past counts z_i^t ($1 \leq t \leq T$) for each spatial location s_i. Furthermore, let us make three simplifying assumptions: that the historical data contains no relevant clusters, that the time series of counts for each location is stationary, and that counts are uncorrelated. Then under the null hypothesis of no clusters, we expect that the current count c_i for each location will be drawn from the same distribution as the historical counts z_i^t for that location. Thus we can define the empirical p-value P_i for each spatial location s_i to be the ratio $\frac{T_{beat}+1}{T+1}$, where T_{beat} is the number of historical counts z_i^t larger than c_i. Under the null hypothesis, and given the simplifying assumptions above, each location's empirical p-value will be asymptotically uniformly distributed on $[0, 1]$. We wish to detect regions S where the counts c_i are higher than expected, and thus where the P_i are lower than expected. In other words, we wish to test the null hypothesis H_0 against the set of alternative hypotheses $H_1(S)$, where:

H_0: $P_i \sim$ Uniform$[0, 1]$ everywhere.
$H_1(S)$: $P_i \sim g(x)$ inside S, and $P_i \sim$ Uniform$[0, 1]$ outside S, for some unknown probability distribution $g(x)$ with cumulative distribution $G(x)$ satisfying $G(0) = 0$, $G(1) = 1$, and $G(\alpha) \geq \alpha$ for all $0 \leq \alpha \leq 1$.

To test this hypothesis, we use the "higher criticism" method of Donoho and Jin [37]. We first compute the empirical p-value P_i for each spatial location. For any constant $0 < \alpha < 1$, we expect each P_i to be less than α with probability α under the null hypothesis, or with probability at least α under any alternative hypothesis $H_1(S)$ such that $s_i \in S$. For a region S, we can define $N(S)$ to be the number of spatial locations in S, and $N_\alpha(S)$ to be the number of spatial locations in S with $P_i < \alpha$. Then we expect $N_\alpha(S)$ to be binomially distributed with mean $\alpha N(S)$ and variance $\alpha(1-\alpha)N(S)$ under H_0, and we expect the values of $N_\alpha(S)$ to be larger than $\alpha N(S)$ under the alternative hypothesis $H_1(S)$. Following [37], we can select a range of α that we are interested in,

$\alpha_{min} < \alpha < \alpha_{max}$, and we can define the non-parametric scan statistic as follows:

$$F(S) = \max_{\alpha_{min} < \alpha < \alpha_{max}} \frac{N_\alpha(S) - \alpha N(S)}{\sqrt{\alpha(1-\alpha)N(S)}}$$

As usual, we can compute the most significant region S^* by finding the maximum value of $F(S)$, and calculate the statistical significance of this region by randomization testing. For the non-parametric statistic, we can generate the empirical p-values of the replica datasets directly, drawing each value from the uniform distribution on $[0, 1]$. An alternative method of significance testing, in keeping with the non-parametric approach, would be to compute the test statistic $F^* = \max_S F(S)$ of the historical data for each time step t, then compute the empirical p-value of the maximum region score using these values.

To use the non-parametric scan statistic in practice, we must consider the simplifying assumptions above. Most importantly, we do not expect the time series of counts to be stationary in most applications, but must adjust for covariates such as seasonal and day-of-week trends. Thus we must apply the statistic to counts that have been adjusted for these and other relevant covariates. Additionally, we can account for correlated counts by examining the pairwise correlations of the empirical p-values: more precisely, we can add the quantity $2\sum_{s_{i_1}, s_{i_2} \in S}(\Pr(P_{i_1} < \alpha, P_{i_2} < \alpha) - \alpha^2)$ to the variance of the binomial distribution for $N_\alpha(S)$, and adjust the score function $F(S)$ accordingly. Randomization testing can be performed by generating correlated counts or by using the empirical p-value of the maximum score, as discussed above.

Chapter 3

Fast algorithms for spatial cluster detection

3.1 Introduction

This chapter focuses on computational methods for rapid and efficient spatial cluster detection. Efficient cluster detection algorithms are necessary for two reasons: first, because we are often searching for clusters in huge spatial datasets, making naïve methods computationally infeasible, and second, because application domains such as disease surveillance require us to detect and respond to clusters as soon as possible. When responding to an emerging outbreak of disease, every hour of earlier detection has the potential to significantly reduce morbidity and mortality rates [149]. There are three sources of "lag time" between the onset of an outbreak and the earliest time we could possibly detect the outbreak: the time it takes patients to generate indicative data (i.e. visiting emergency departments or buying over-the-counter drugs), the time to collect this data (e.g. by the National Retail Data Monitor) and make it available for analysis, and the time it takes to analyze the data and report results. For massive datasets, the lag time resulting from data analysis has the potential to be huge, but faster algorithms will help to reduce this lag time. We typically receive syndromic data on a daily or hourly basis, and thus we want to achieve "near real time" analysis, processing data in minutes or hours rather than in days or weeks. Moreover, our algorithms should be fast enough that multiple such analyses (e.g. different statistical models, or different data streams) can be performed, giving public health officials a better situational awareness. An eventual goal of this work is to enable real-time cluster detection and investigation, allowing public health officials to explore multiple time series and perform multiple spatial scan queries "on the fly." Our work toward this goal is discussed in Chapter 8.

The centerpiece of my discussion of computational methods is the "fast spatial scan," a new multi-resolution search algorithm which allows us to perform the spatial scan hundreds or thousands of times faster without any loss of accuracy. This algorithm relies on a new type of space-partitioning data structure which we call the "overlap-kd tree," and this data structure might also be useful for speeding up other spatial search algorithms. The fast spatial scan is presented in Section 3.3; before presenting this method, I provide an overview of the computational problem and describe the standard "naïve" computational methods in Section 3.2. Finally, Section 3.4 presents results of running the fast spatial scan on various public health and brain imaging datasets, and compares the fast spatial scan to other computational approaches. I also discuss computational methods further in Chapter 4 (space-time scanning) and Chapter 5 (the Bayesian scan statistic).

Much of this chapter has been adapted from our papers in KDD 2004 [112], NIPS 2004 [118], and the 2004 National Syndromic Surveillance Conference [119], as well as our chapters in the *Handbook of Biosurveillance* [115] and *Spatial and Syndromic Surveillance for Public Health* [114]. I wish to thank my co-authors Andrew Moore, Maheshkumar Sabhnani, Francisco Pereira, and Tom Mitchell, as well as editors Mike Wagner and Andrew Lawson, for their contributions. The fast spatial scan was first derived for the two-dimensional case in [112], and extended to the multidimensional case in [118]; it was first presented to the public health community in [114, 119]. We also presented an earlier, approximate version of the fast spatial scan in [111, 110], but we focus here on the exact, more efficient method presented in our later work.

3.1.1 Searching for elongated regions

Most of the previous approaches to cluster detection search for *compact* clusters, such as circles (e.g. Kulldorff [78]) or squares. One exception to this is the work of Kulldorff et al. [86], who search over a subset of the elliptical clusters, but this method is computationally infeasible for even moderately-sized datasets. Our fast spatial scan method, however, allows for rapid and efficient detection of *elongated* clusters as well: we search over the space of rectangular clusters. This extension is extremely important in epidemiological applications because disease clusters are often elongated: airborne pathogens may be blown by wind, creating an ellipsoid “plume,” and waterborne pathogens may be carried along the path of a river. In each of these cases, the resulting clusters have high aspect ratios, and a test for compact clusters will have low power for detecting the affected region. Detection of clusters with high aspect ratios is also important in brain imaging, because of the “folded sheet” structure of the brain, and in astrophysics, because galaxies and other astronomical objects may be elongated in shape.

While our discussion below focuses on finding “axis-aligned” rectangular regions, assuming that data points have been aggregated to a grid, the fast spatial scan can be easily extended to find rectangular regions which are not aligned with the coordinate axes. One simple method of doing this is to examine multiple “rotations” of the data, mapping each to a separate grid and computing the most significant region and its score for each grid. In this case, we must also perform the same rotations on each replica grid, and thus the complexity of the algorithm is multiplied by the number of rotations. However, if we have information about relevant conditions such as wind direction or the flow of a river, we can use this information to align the coordinate axes, reducing or avoiding the need to examine multiple rotations.

3.1.2 Searching for multidimensional regions

In [118], we extended the fast spatial scan to multidimensional datasets, dramatically increasing the scope of problems for which these techniques can be used. In addition to datasets with more than two spatial dimensions (for example, functional magnetic resonance imaging data, which consists of a 3D image of brain activity), we can also examine data with a temporal component, or where we wish to take demographic information such as age and gender into account. For example, for biosurveillance datasets (e.g. over-the-counter drug sales data), we can use *time* as a third, “pseudo-spatial” dimension, in addition to the spatial dimensions of longitude and latitude. Searching for clusters in this three-dimensional space allows us to search for *persistent spatial clusters*: spatial regions where the counts are higher than expected for some length of time. As another example, we can use gender and age decile as pseudo-spatial dimensions in the Emergency Department dataset, and search for clusters in this four-dimensional space. This gives our test higher power to detect outbreaks which affect different patient demographics to different extents. For example, if a disease primarily strikes elderly males, we might find a cluster with gender = male and age decile ≥ 6 in some spatial region, and this cluster may not be detectable from the combined data. This method accounts correctly for the multiple hypothesis testing resulting from testing different combinations of genders and age groups; if we were to instead perform a separate test at level α on each combination of gender and age decile, the overall false positive rate would be much higher than α due to multiple testing.

3.2 Computational issues in spatial scanning

In this section, we return to the question of what set of regions to search over (first considered in the generalized spatial scan framework of Chapter 2), and discuss how to perform this search efficiently. First, we note that the run time of the naïve spatial scan can be approximated by the product of three factors: the number of replications R, the average number of regions searched per replication $|S|$, and the average time to search a region t. The number of replications R is typically fixed in advance, but we can stop early if many replicas beat the original search area (i.e. the maximum region scores F^* of the replicas are higher than the maximum region score F^* of the original). If this happens, it is clear that no significant clusters are present. The other two factors $|S|$ and t depend on both the set of regions to be searched and the algorithm used to search these regions. For a set of M distinct spatial locations in two dimensions, the number of circular or axis-aligned square regions (assuming that the size of the circle or square can vary) is proportional to M^3, while the number of axis-aligned rectangular regions (assuming that both dimensions of the rectangle can vary) is proportional to M^4. For non-axis-aligned squares or rectangles, we must also multiply this number by the number of different orientations searched. However, most algorithms only search a subset of these regions: for example, Kulldorff's algorithm [80] searches only circles centered at one of the M spatial locations, and the number of such regions is proportional to M^2, not M^3. Another possibility is to aggregate the spatial locations to a grid, either uniform or based on the distinct spatial coordinates of the data points. For a two-dimensional, $N \times N$ grid, the number of axis-aligned square regions is proportional to N^3, and the number of axis-aligned rectangular regions is proportional to N^4. Whatever set of regions we choose, the simplest possible implementation of the scan statistic is to search each of these regions by stepping through the M spatial locations, determining which locations are inside and outside the region, computing the aggregate baselines and counts, and applying the score function. Thus in this approach, we have $|S|$ (number of regions searched per replication) equal to the total number of distinct regions, and t (time to search a region) proportional to the number of spatial locations M.

There are several possible ways to improve on the runtime of this naïve approach. First, we can reduce the time to search a region t, making this search time independent of the number of spatial locations M. We consider two possible methods for searching a region in constant time. The first method, which we call "incremental addition," assumes that we want to search over all regions of a given type: for example, in the approach of Kulldorff [80], we want to search all distinct circular regions centered at one of the spatial locations. To do so, we increase the region's size incrementally, such that one new spatial location at a time enters the region; for each new location, we can add that location's count and baseline to the aggregates, and recompute the score function. For example, in Kulldorff's method, for each location s_i we keep a list of the other locations, sorted from closest to furthest away. Then we can search over the M distinct circular regions centered at s_i by adding the other points one at a time in order. Because the sorting only has to be done once (and does not have to be repeated for each replication) this results in constant search time per region. In other words, Kulldorff's method requires time proportional to M^2 to search over all of the M^2 regions. This must be done for each of the R replications, giving total search time proportional to RM^2.

The second method assumes that points have been aggregated to an $N \times N$ grid, and that we are searching over squares or rectangles. We can use the well-known "cumulative counts" technique to search in constant time per region. We first precompute a matrix of the *cumulative counts* $cc_{ij} = \sum_{k=1\ldots i} \sum_{l=1\ldots j} c_{kl}$ in $O(N^2)$ operations, using dynamic programming. We can then compute each region's count by adding/subtracting at most four cumulative counts, and similarly for baselines.[1] Thus we can calculate the score of a region in $O(1)$ by computing the count C and baseline B, then applying the score function $F(C, B)$. As a result, we can perform the scan statistic for gridded square or rectangular regions in time proportional to R times the number of regions, i.e. RN^3 or RN^4 for square or rectangular regions respectively. We also note that the cumulative counts technique can be used in d dimensions: in this case, we must add/subtract a number of counts that scales exponentially with dimension but is still independent of the grid size N. Thus a naïve search requires time $O(RN^{d+1})$ or $O(RN^{2d})$ for d-dimensional hypercubes or d-dimensional hyper-rectangles respectively.

Even if we can search in constant time per region, the spatial scan statistic is still extremely computationally expensive, because of the large number of regions searched. For example, to search over all rectangular regions on a 256×256 grid, and perform randomization testing (assuming $R = 999$ replications), we must search a total of 1.1 trillion regions, which would take 14-45 days on our test systems. This is clearly far too slow for real-time detection of emerging disease outbreaks. While one option is to simply search fewer regions, this reduces our power to detect clusters. A better option is provided by the fast spatial scan algorithms discussed below, which allow us to reduce the number of regions searched, but without losing any accuracy. The idea is that, since we only care about the most significant regions, i.e. those with the highest scores $F(S)$, we do not need to search a region S if we know that it will not have a high score. Thus we start by examining large regions S, and if we can show that none of the smaller regions contained in S can have high scores, we do not need to actually search each of these regions. Thus, we can achieve the same result as if we had searched all possible regions, but by only searching a small fraction of these. Further speedups are gained by the use of multiresolution data structures, which allow us to efficiently move between searching at coarse and fine resolutions. These methods are able to search hundreds or thousands of times faster than an exhaustive search, without any loss of accuracy (i.e. the fast spatial scan finds exactly the same region and p-value as exhaustive search). As a result, these methods have enabled us to perform spatial scans on datasets such as nationwide over-the-counter sales data, from over 20,000 stores in near real-time, searching for disease clusters in minutes or hours rather than days or weeks.

[1] More precisely, we have $\sum_{k=i_1\ldots i_2} \sum_{l=j_1\ldots j_2} c_{kl} = cc_{i_2,j_2} - cc_{i_2,j_1-1} - cc_{i_1-1,j_2} + cc_{i_1-1,j_1-1}$, where $cc_{i,0} = cc_{0,j} = 0$.

3.3 The fast spatial scan algorithm

For the fast spatial scan, we consider the case in which data have been aggregated to a d-dimensional grid. Let G be a d-dimensional grid of cells, with size $N_1 \times N_2 \times \ldots \times N_d$. Each cell $s_i \in G$ (where i is a d-dimensional vector) is associated with a *count* c_i and a *baseline* b_i. As discussed above, the count of a cell might represent the number of disease cases occurring in that geographical area over some time interval, while the baseline of that cell might represent an expected count (estimated from past data) or at-risk population. Given these counts and baselines, our goal is to search over all d-dimensional rectangular regions $S \subseteq G$, and find regions where the total count $C(S) = \sum_S c_i$ is higher than expected, given the baseline $B(S) = \sum_S b_i$. In addition to discovering these high-density regions, we must also perform statistical testing to determine whether these regions are significant. Though we focus on finding the single, most significant region, the method can be iterated (removing each significant cluster once it is found) to find multiple significant regions.

As discussed in the previous chapter, we can find the most significant region and its p-value by deriving a score function $F(S)$ based on the null and alternative hypotheses we wish to compare, finding the region S^* with the maximum value of $F(S)$, and performing randomization testing to calculate significance. We present a fast search algorithm which is applicable to a general function $F(S)$, where $F(S)$ is based on the total count of region S, $C(S) = \sum_S c_i$, and the total baseline of region S, $B(S) = \sum_S b_i$. Thus we will often write $F(C, B)$, where C and B are the count and baseline of the region under consideration. In this discussion, we assume that the score function F satisfies the following three intuitive properties:

1. For a fixed baseline, score increases monotonically with count: $\frac{\partial F}{\partial C}(C, B) \geq 0$ for all (C, B).

2. For a fixed count, score decreases monotonically with baseline: $\frac{\partial F}{\partial B}(C, B) \leq 0$ for all (C, B).

3. For a fixed ratio $\frac{C}{B}$, score increases monotonically with count and baseline: $\frac{\partial F}{\partial B}(C, B) + \frac{C}{B}\frac{\partial F}{\partial C}(C, B) \geq 0$ for all (C, B).

The first two properties state that an overdensity of counts is present when count is large relative to baseline; thus score will be increased by either increasing the count or decreasing the baseline. The third property states, in essence, that an overdensity of counts is more significant when the underlying count and baseline are large. As a simple example, a region where 20% of the population is sick ($\frac{C}{B} = .20$) might be very significant if it represented ten thousand sick people out of fifty thousand, but not so significant if it represented five people, one of whom is sick. More generally, smaller counts and baselines will typically result in higher variance in the ratio $\frac{C}{B}$. For example, assuming that counts are Poisson distributed with means proportional to B, the variance of $\frac{C}{B}$ is proportional to $\frac{B}{B^2} = \frac{1}{B}$. Thus a higher than expected ratio of count to baseline will be increased in significance when count and baseline are large.

Here we present a *fast spatial scan* algorithm which is exact (always finds the correct value of F^* and the corresponding region S^*) and yet is much faster than naïve search. The key intuition is that, since we only care about finding the highest scoring region, we do not need to search over every single rectangular region: in particular, we do not need to search a set of regions if we can prove that none of them can have $F(S) \geq F^*$. As a simple example, if a given region has a very low count, we may be able to conclude that *no* subregion contained in that region can have a score higher than F^*, and thus we do not need to actually compute the score of each subregion. Thus we use a top-down, *branch-and-bound* approach: we maintain the current maximum score F^* of the regions we have searched so far, calculate upper bounds on the scores of subregions contained in a given region, and *prune* regions whose upper bounds are less than the current value of F^*. When searching a replica grid, we care only whether F^* of the replica grid is greater than $F^*(G)$. Thus we can use F^* of the original grid for pruning on the replicas, and can stop searching a replica if we find a region with $F(S) > F^*(G)$.

3.3.1 The overlap-kd tree data structure

Our top-down approach to cluster detection can be thought of as a multiresolution search of the space under consideration: we search first at coarse resolutions (large regions), then at successively finer resolutions (smaller regions) as necessary. This suggests that a hierarchical, space-partitioning data structure such as kd-trees [13, 127], mrkd-trees [34], or quadtrees [48, 133] may be useful in speeding up our search. However, our desire for an exact solution makes it difficult to apply these data structures to our problem. In a kd-tree, each spatial region is recursively partitioned into two disjoint "child" regions, each of which can then be further subdivided. The difficulty, however, is that many subregions of the parent are not contained entirely in either child, but overlap partially with each. Thus, in addition to recursively searching each child region, we must also search over all of these "shared" regions at each level of the tree. For d-dimensional data, there are $O(N^{2d})$ shared regions even at the top level of the tree (i.e. regions partially overlapping both halves of grid G). Thus an exhaustive search over all such regions is too computationally expensive, and a different partitioning approach is necessary. A second option would be to work in bottom-up fashion, attempting to find the two "pieces" of the highest scoring region, one in each child, and then merge the two. This approach fails because of the non-monotonicity of the score function: the highest scoring region may have a *higher* score than either of its two pieces.

Here we improve on the top-down search idea by using a new data structure, the overlap-kd tree, where the children of a region overlap. The idea of overlapping children is common in the literature, and has been used in data structures including R-trees [62]. The difference is that our data structure must be optimized for the task of searching over all regions to find the highest scoring region, rather than dynamic insertion and deletion of spatial data. Since we cannot afford to do individual tree searches for each region, an R-tree is too inefficient for our search task, and instead we consider a new variant of the kd-tree with overlapping children.

An initial step toward the overlap-kd tree data structure can be seen by considering two divisions of a two-dimensional rectangular spatial region S: first, into its left and right halves (which we denote by S_1 and S_2), and second, into its top and bottom halves (which we denote by S_3 and S_4). Assuming that S has size $k_1 \times k_2$, this means that S_1 and S_2 have size $\frac{1}{2}k_1 \times k_2$, and S_3 and S_4 have size $k_1 \times \frac{1}{2}k_2$. Considering these four (overlapping) halves, we can show that any subregion of S either a) is contained entirely in (at least) one of $S_1 \ldots S_4$, or b) contains the centroid of S. Thus one possibility would be to search S by exhaustively searching all regions containing its centroid, then recursing the search on its four "children" $S_1 \ldots S_4$. Again, there are $O(N^{2d})$ "shared" regions at the top level of the tree (i.e. regions containing the centroid of grid G), so an exhaustive search is infeasible.

Our solution is a partitioning approach in which adjacent regions partially overlap, a technique we call "overlap-multiresolution partitioning." Again we consider the division of S into its left, right, top, and bottom "children." However, while in the discussion above each child contained exactly half the area of S, now we let each child contain *more* than half the area. We again assume

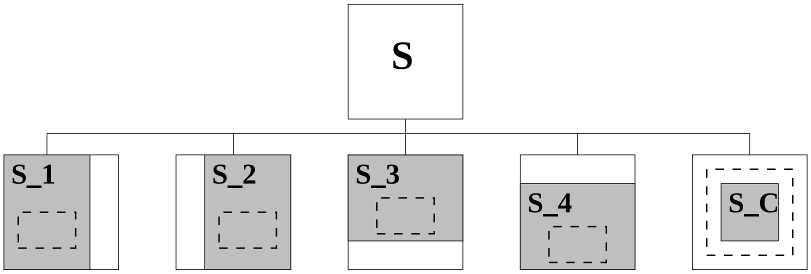


Figure 3.1: Overlap-multiresolution partitioning of region S (for $d = 2$). Any subregion of S either a) is contained in some S_i, $i = 1 \ldots 4$, or b) contains S_C.

that region S has size $k_1 \times k_2$, and we choose fractions $f_1, f_2 > \frac{1}{2}$. Then S_1 and S_2 have size $f_1 k_1 \times k_2$, and S_3 and S_4 have size $k_1 \times f_2 k_2$. This partitioning (for $f_1 = f_2 = \frac{3}{4}$) is illustrated in Figure 3.1. Note that there is a region S_C common to all four children; we call this region the *center* of S. The size of S_C is $((2f_1 - 1)k_1 \times (2f_2 - 1)k_2)$, and thus the center has non-zero area. When we partition region S in this manner, it can be proved that any subregion of S either a) is contained entirely in (at least) one of $S_1 \ldots S_4$, or b) contains the center region S_C. Figure 3.1 illustrates each of these possibilities.

This partitioning approach may be extended to arbitrary dimension, resulting in a novel data structure which we term an *overlap-kd tree*. The overlap-kd tree, like kd-trees and quadtrees, is a hierarchical, space-partitioning data structure. The root node of the tree represents the entire space under consideration (i.e. the entire grid G), and each other node represents a subregion of the grid. Each non-leaf node of a d-dimensional overlap-kd tree has $2d$ children, an “upper” and a “lower” child in each dimension. For example, in three dimensions, a node has six children: upper and lower children in the x, y, and z dimensions. The overlap-kd tree is different from the standard kd-tree and quadtree in that adjacent regions overlap: rather than splitting the region in half along each dimension, instead each child contains *more* than half the area of the parent region. In general, let region S have size $k_1 \times k_2 \times \ldots \times k_d$. Then the two children of S in dimension j (for $j = 1 \ldots d$) have size $k_1 \times \ldots \times k_{j-1} \times f_j k_j \times k_{j+1} \times \ldots \times k_d$, where $\frac{1}{2} < f_j < 1$. Defining the center S_C as the region common to all of these $2d$ children, it can be proved (as in the two-dimensional case) that any subregion of S either a) is contained entirely in at least one of $S_1 \ldots S_{2d}$, or b) contains the center region S_C. A picture of this partitioning in the three-dimensional case is given in Figure 3.2.

Now we can search all subregions of S by recursively searching $S_1 \ldots S_{2d}$, then searching all of the regions contained in S which contain the center S_C. Unfortunately, there may still be a large number of such “outer regions”: at the top level there are $O(N^{2d})$ regions contained in grid G which contain its center G_C. However, since we know that each such region contains the large region G_C, we can place very tight bounds on the score of these regions, often allowing us to prune most or all of them. We discuss how these bounds are calculated in the following subsection. Thus the basic outline of our search procedure (ignoring pruning, for the moment) is:

```
overlap-search(S)
{
  call base-case-search(S)
  define child regions S_1..S_2d, center S_C as above
  call overlap-search(S_i) for i=1..2d
```

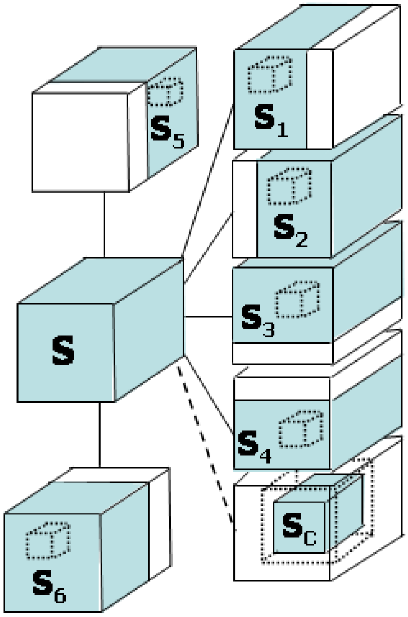


Figure 3.2: Overlap-multiresolution partitioning of region S (for $d = 3$). Any subregion of S either a) is contained in some S_i, $i = 1 \ldots 6$, or b) contains S_C.

```
  for all S' such that S' is contained in S and contains S_C,
    call base-case-search(S')
}
```

Now we consider how to select the fractions f_i for each call of overlap-search, and characterize the resulting set Φ of regions S on which overlap-search(S) is called. Regions $S \in \Phi$ are called *gridded regions*, and regions $S \notin \Phi$ are called *outer regions*. We begin the search by calling overlap-search(G). Then for each recursive call to overlap-search(S), where the size of S is $k_1 \times \ldots \times k_d$, we set each f_i based on the value of k_i: $f_i = \frac{3}{4}$ if $k_i = 2^r$ for some integer r, and $f_i = \frac{2}{3}$ if $k_i = 3 \times 2^r$ for some integer r. For simplicity, we assume that all N_i are either a power of two, or three times a power of two, and thus all region sizes k_i will fall into one of these two cases. For instance, if the original grid G has size 64×64, then the children of G will be of sizes 64×48 and 48×64, and the grandchildren of G will be of sizes 64×32, 48×48, and 32×64. Repeating this partitioning recursively down to regions of size 1 (or larger, if we so choose), we obtain the overlap-kd tree structure. For $d = 2$, the first two levels of the overlap-kd tree are shown in Figure 3.3. Note that even though grid G has four child regions, and each of its child regions has four children, G has only ten (not 16) distinct grandchildren, several of which are the child of multiple regions. The X's on the tree will be discussed later, and can be ignored for now.

The overlap-kd tree has several useful properties, which we present here without proof. First, for every rectangular region $S \subseteq G$, either S is a gridded region (contained in the overlap-kd tree), or there exists a unique gridded region S' such that S is an outer region of S' (i.e. S is contained in S', and contains the center region of S'). This means that, if overlap-search is called exactly once for each gridded region, and no pruning is done, then base-case-search will be called exactly once for every rectangular region $S \subseteq G$. In practice, we will prune many regions, so base-case-search

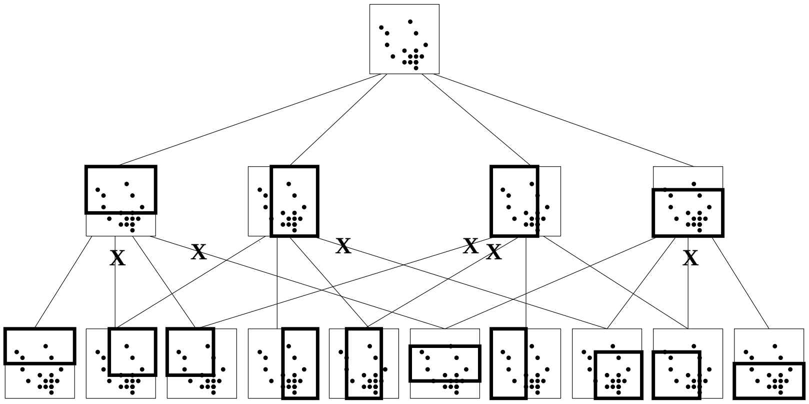


Figure 3.3: The first two levels of the two-dimensional overlap-kd tree. Each node represents a gridded region (denoted by a thick rectangle) of the entire dataset (thin square and dots).

will be called *at most once* for every rectangular region, and every region will be either searched or pruned. The second nice property of our overlap-kd tree is that the total number of gridded regions $|\Phi|$ is $O((N \log N)^d)$ rather than $O(N^{2d})$. This implies that, if we are able to prune (almost) all outer regions, we can find the most significant region in $O((N \log N)^d)$ time. In fact, we may not even need to search all gridded regions, so in many cases the search will be even faster.

Before we consider how to calculate score bounds and use them for pruning, we must first deal with an essential issue in searching overlap-kd trees. Since a child region may have multiple parents, how do we ensure that each gridded region is examined only once, rather than being called recursively by each parent? One simple answer is to keep a hash table of the regions we have examined, and only call overlap-search(S) if region S has not already been examined. The disadvantage of this approach is that it requires space proportional to the number of gridded regions, $O((N \log N)^d)$, and spends a substantial amount of time doing hash queries and updates. A more elegant solution is what we call *lazy expansion*: rather than calling overlap-search(S_i) on all the children of a region S, we selectively expand only certain children at each stage, in such a way that there is exactly one path from the root of the overlap-kd tree to any node of the tree. One such scheme is shown in Figure 3.3: if the path between a parent and child is marked with an X, lazy expansion does not make that recursive call. No extra space is needed by this method; instead, a simple set of rules is used to decide which children of a node to expand. A child is expanded if it has no other parents, or if the parent node has the highest *priority* of all the child's parents. We give parents with lower aspect ratios priority over parents with higher aspect ratios: for example, a 48×48 parent would have priority over a 64×32 parent if the two share a 48×32 child. This rule allows us to perform variants of the search where regions with very high aspect ratios are not included; an extreme case would be to only search for squares, as in our earlier fast spatial scan work [111]. Within an aspect ratio, we fix an arbitrary priority ordering. Since we maintain the property that every node is accessible from the root, the correctness of our algorithm is maintained: every gridded region will be examined (if no pruning is done), and thus every region $S \subseteq G$ will be either searched or pruned.

3.3.2 Score bounds

We now consider which regions can be *pruned* (discarded without searching) during our multiresolution search procedure. First, given some region S, we must calculate an upper bound on the scores $F(S')$ for regions $S' \subset S$. More precisely, we are interested in two upper bounds: a bound on the score of *all* subregions $S' \subset S$, and a bound on the score of the *outer* subregions of S (those regions contained in S and containing its center S_C). We compare these to the maximum region score $F^* = \max F(S)$ that we have found so far in our search. If the first bound is less than or equal to the current value of F^*, we can prune region S completely; we do not need to search any (gridded or outer) subregion of S. If only the second bound is less than or equal to the current value of F^*, we do not need to search the outer subregions of S, but we must recursively call overlap-search on the gridded children of S. If both bounds are greater than the current value of F^*, we must both recursively call overlap-search and search the outer regions.

Score bounds are calculated based on various pieces of information about the subregions of S, including: upper and lower bounds b_{max}, b_{min} on the baseline of subregions S'; an upper bound d_{max} on the ratio $\frac{C}{B}$ of S'; an upper bound d_{inc} on the ratio $\frac{C}{B}$ of $S' - S_C$; and a lower bound d_{min} on the ratio $\frac{C}{B}$ of $S - S'$. We also know the count C and baseline B of region S, and the count c_{center} and baseline b_{center} of region S_C.

We will focus here on finding an upper bound on the scores of all subregions of S containing the center of S. (We can also upper bound the scores of *all* subregions of S as a special case, where the baseline, count, and area of the center are zero.) To compute this bound, let c_{in} and b_{in} be the count and baseline of S'. To find an upper bound on $F(S')$, we must calculate the values of c_{in} and b_{in} which maximize F subject to the given constraints:

1. $\frac{c_{in} - c_{center}}{b_{in} - b_{center}} \leq d_{inc}$
2. $\frac{c_{in}}{b_{in}} \leq d_{max}$
3. $\frac{C - c_{in}}{B - b_{in}} \geq d_{min}$
4. $b_{min} \leq b_{in} \leq b_{max}$

While we could use convex programming to solve this maximization problem in the general case, the properties of the score function make this task significantly easier, allowing us to calculate the optimal values of c_{in} and b_{in}. Since $\frac{\partial F}{\partial C} \geq 0$, we know that the maximum value of F for a given b_{in} occurs when c_{in} is maximized subject to the constraints. We solve the first three constraints for c_{in}, giving us $c_{in} = \min(C_1, C_2, C_3)$, where:

$$C_1 = d_{inc} b_{in} - (d_{inc} b_{center} - c_{center}) = d_{inc} b_{in} - Z_1$$

$$C_2 = d_{max} b_{in}$$

$$C_3 = d_{min} b_{in} + (C - d_{min} B) = d_{min} b_{in} + Z_3$$

In the typical case,[2] we have $d_{min} \leq d_{max} \leq d_{inc}$, $Z_1 > 0$, and $Z_3 > 0$: this means that $c_{in} = C_1$ for small b_{in}, $c_{in} = C_2$ for moderate b_{in}, and $c_{in} = C_3$ for large b_{in}, as illustrated in Figure 3.4.

[2] We must also handle a variety of special cases where one or more of these inequalities are violated, and some constraints may not be relevant. We omit the details of this case-by-case analysis.

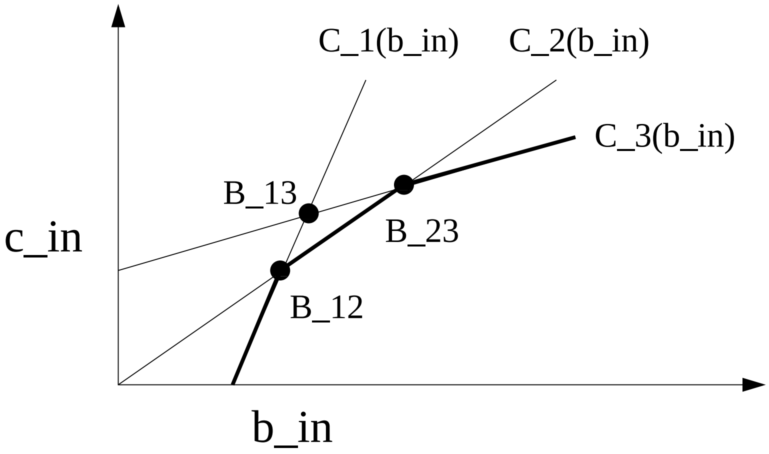


Figure 3.4: Maximizing count c_{in} for a given baseline b_{in}. Count must be less than $C_1(b_{in})$, $C_2(b_{in})$, and $C_3(b_{in})$.

Thus we can solve for the intersection points B_{12}, B_{13}, and B_{23}, where $C_i \leq C_j$ for $b_{in} \leq B_{ij}$, and we use these quantities to find the maximum allowable count c_{in} for a given b_{in}. Solving the equations, we find that $B_{12} = \frac{Z_1}{d_{inc}-d_{max}}$, $B_{13} = \frac{Z_1+Z_3}{d_{inc}-d_{min}}$, and $B_{23} = \frac{Z_3}{d_{max}-d_{min}}$. In the typical case,[3] we have $0 < B_{12} \leq B_{13} \leq B_{23} < \infty$. In this case, we use the values of B_{12} and B_{23}, and the value B_{13} is not needed. Then the count c_{in} = $c_{center} + d_{inc}(b_{in} - b_{center})$ for $b_{center} \leq b_{in} \leq B_{12}$, c_{in} = $d_{max}b_{in}$ for $B_{12} \leq b_{in} \leq B_{23}$, and c_{in} = $d_{max}B_{23} + d_{min}(b_{in} - B_{23})$ for $b_{in} \geq B_{23}$. This is illustrated by Figure 3.5: the region S is separated into four "layers" of differing rates (ratio of counts to baseline). Starting from the inside, we have the center (with a known baseline b_{center} and count c_{center}), a layer of high rate d_{inc}, a layer of moderate rate d_{max}, and a layer of low rate d_{min}.

Now we can write c_{in} as a function of b_{in}, and thus the score F becomes a function of the single variable b_{in}. Where does the maximum of this function occur? Again we rely on properties of the function $F(C, B)$, and a case-by-case analysis is necessary. In the typical case $d_{inc} > d_{max} > \frac{c_{center}}{b_{center}}$, we know that the score increases with baseline in the "high rate" and "moderate rate" layers. This follows from two properties of our score function: $\frac{\partial F}{\partial C} \geq 0$ and $\frac{\partial F}{\partial B} + \frac{C}{B}\frac{\partial F}{\partial C} \geq 0$. In the high rate layer, the ratio of counts to baselines for S' $\left(\frac{c_{in}}{b_{in}}\right)$ increases from $\frac{c_{center}}{b_{center}}$ to d_{max} as we add more baselines, so the score F is monotonically increasing with baseline. In the moderate rate layer, the ratio of counts to baselines for S' stays constant (at d_{max}) as baseline increases, so again F is monotonically increasing. In the low rate layer, the ratio of counts to baselines for S' *decreases* as baseline increases: in this case, since count and baseline are both increasing, the score may increase or decrease. We assume that the score function F has no local maxima in the interval (B_{23}, B), and thus that the maximum occurs either at $(c_{in}, b_{in}) = (d_{max}B_{23}, B_{23})$ or at $(c_{in}, b_{in}) = (C, B)$.[4] We are only interested in finding subregions with scores *higher* than the parent, so we can ignore the

[3] See previous note.

[4] Formally, we assume the following constraint on the first and second partials of F: $F_B^2 F_{CC} + F_C^2 F_{BB} - 2F_C F_B F_{CB} \geq 0$. This is true for a large class of functions, including Kulldorff's statistic. If this constraint is violated, we must also calculate $F(C, B)$ at each local maximum, which is not difficult if the number of maxima is small and each maximum is easy to calculate.

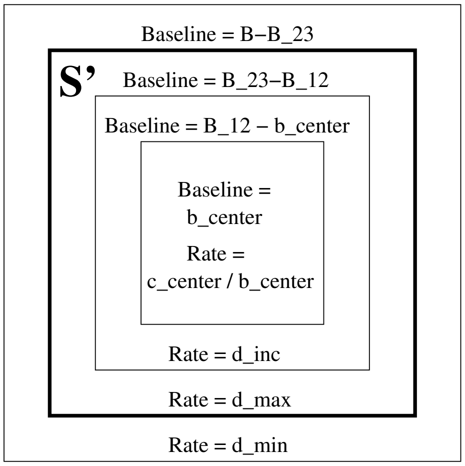


Figure 3.5: Division of region S into layers of differing rate. In the typical case, subregion S' includes all but the outer layer.

latter case. Thus our upper bound on $F(S')$ is $F(c_{in}, b_{in})$, where $b_{in} = B_{23}$ and $c_{in} = d_{max}b_{in}$. The various special cases, where one or more of the inequalities above are violated, are handled similarly using the intersection points B_{12}, B_{13}, and B_{23} as necessary. We also must adjust our value of b_{in} if it violates the inequality $b_{min} \leq b_{in} \leq b_{max}$, adjusting c_{in} accordingly given the rate of the layers being added or subtracted.

3.3.3 Calculating bounds by quartering

We now consider how the bounds on baselines and rates (ratios of count to baseline) are obtained. The simplest method of doing so is to use global values: first, we precompute the minimum and maximum baselines B and ratios $\frac{C}{B}$ of all "small" regions S in the grid, requiring time $O(N^d)$. To do this, we first precompute the minimum and maximum baseline and rate of all single cells s_i in the grid. We can also use the minimum and maximum baselines of a grid cell, together with the minimum and maximum area of a region, to obtain bounds b_{min} and b_{max}. Slightly less conservative bounds can be obtained using the assumption of a minimum region size k_{min}, and these can be used rather than the single square bounds when allowable. This gives us usable (though very conservative) values for d_{min}, d_{max}, d_{inc}, b_{min}, and b_{max}. These global bounds are inexpensive to compute (we need only compute them once per grid), but result in very conservative estimates of region scores.

Thus we use these bounds in our algorithm as a first pass which prunes many regions but also leaves many unpruned. If a region survives this round of pruning, we compute much tighter bounds on region scores in a second pass, which is also more computationally expensive. To do so, we obtain tighter bounds on the baselines and rates using a novel technique we term *quartering*, then use these constraints to bound $F(S')$ as above. We explain the quartering method for the two-dimensional case ($d = 2$) but note that we have generalized this procedure to arbitrary dimension.

Given a region S of size $k_1 \times k_2$, with a (non-zero) center region S_C, the first step of quartering is to divide S into its four (non-overlapping) quadrants $S_1 \ldots S_4$, as in Figure 3.6. We now consider each S_i separately, together with the quarter of the center (S_{Ci}) which overlaps that quadrant. For

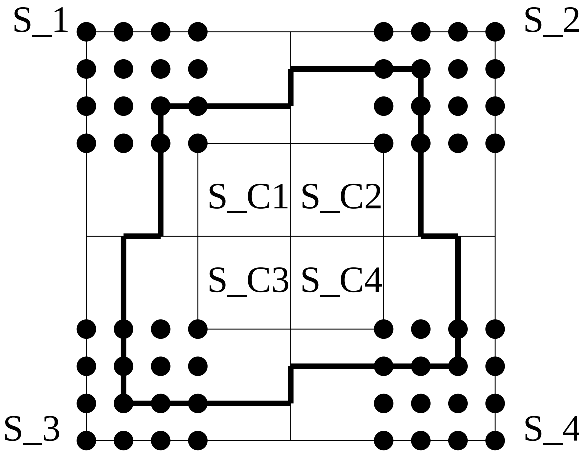


Figure 3.6: Quartering of region S

each quadrant, we consider all rectangles S'_i with one corner at the centroid of S, and one corner outside S_{Ci} (i.e. on one of the dots in Figure 3.6). Note that there are $O(k_1 k_2)$ such rectangles, and thus we can search over all of these regions S'_i in quadratic time, as opposed to $O(k_1^2 k_2^2)$ for naïve search of all $S' \subset S$ containing S_C.

Our search procedure is very simple: given a region S'_i, let b_{in}, c_{in}, and A_{in} denote its baseline, count, and area; let b_{out}, c_{out}, and A_{out} denote the baseline, count, and area of $S_i - S'_i$; and let b_{dif}, c_{dif}, and A_{dif} denote the baseline, count, and area of $S'_i - S_{Ci}$. We then calculate the rate (ratio of count to baseline) d and the average baseline per cell b_s for each of S'_i, $S_i - S'_i$, and $S'_i - S_{Ci}$: $d_{in} = \frac{c_{in}}{b_{in}}$, $b_{s,in} = \frac{b_{in}}{A_{in}}$, and the other quantities (d_{out}, d_{dif}, $b_{s,out}$, $b_{s,dif}$) are defined similarly. We then set d_{max} equal to the maximum of all d_{in}, d_{inc} equal to the maximum of all d_{dif}, and d_{min} equal to the minimum of all d_{out}. Similarly, we take the minimum and maximum values of $b_{s,in}$, $b_{s,out}$, and $b_{s,dif}$; we can use these to calculate bounds b_{min} and b_{max} once we are given the minimum and maximum area of S'. Then ratio of count to baseline is monotonic, so we know that the ratio of the entire region S' is bounded by the maximum of the max-ratios and the minimum of the min-ratios computed for all regions S'_i. Baseline per square is also monotonic, so an identical argument applies.

In essence, what are doing is bounding the baselines and rates for the piece of region S' contained in each quadrant. Then we use the maximum and minimum values of these quantities to bound the baselines and rates for all regions S'. Another way to think of this is that we are calculating bounds on baselines and rates for all the irregular (but rectangle-like) regions containing the center S_C and consisting of one rectangle in each quadrant, as drawn in Figure 3.6; then these quantities are also bounds on the baselines and rates of all *rectangles* which contain S_C. We do not provide a formal proof here, but we note that the bounds on baselines and ratios derived by quartering are exact (i.e. no rectangle $S' \subset S$, such that $S_C \subseteq S'$, can have baseline or rate outside these bounds) and that they are much tighter than the global bounds, allowing many more regions to be pruned. However, as noted above, quartering is significantly more computationally expensive than using the global bounds, taking time proportional to the volume of region S, and thus $O(N^d)$ per region for large regions. This is why we first use the global bounds for pruning outer regions, and only use quartering on regions that this initial pruning does not eliminate.

3.3.4 The algorithm

We now possess all of the algorithmic and statistical tools needed to present our algorithm in full. The basic structure is similar to the top-down "overlap-search" routine presented above, with several important differences. First, we use a best-first search (implemented using a pair of priority queues q_1 and q_2) rather than a recursive depth-first search. Our algorithm has two stages: in the first stage we examine only gridded regions, and in the second stage we search outer regions if necessary. In both stages, we prune regions whenever possible, calculating increasingly tight bounds on subregions' baselines and rates, and using these to calculate upper bounds F_{bound} on $F(S')$ as above. For the original grid, regions are pruned whenever they can be proven to have a score less than the highest value of $F(S)$ found so far; for the replica grids, regions are pruned whenever they can be proven to have a score less than the maximum score of the original grid (and also, we can stop searching a replica immediately if we find a region with score higher than the maximum of the original grid). We can also do this for the case where we are interested in finding the k-best regions of the original grid; in this case, we can simply use the current value of the kth highest score $F(S)$ for pruning. For simplicity, we focus on searching the original grid, and finding the 1-best region, in our presentation of the algorithm below. The first stage of our algorithm proceeds as follows, using the (loose) global bounds on baselines and rates to calculate F_{bound}:

```
Add G to q_1.
While q_1 not empty:
  Get region S with highest F(S) from q_1.
  If F(S) > F*, set S* = S and F* = F(S).
  If F_bound(S’ in S) > F*,
    add gridded children of S to q_1 (using "lazy expansion").
  If F_bound(S’ in S containing S_C) > F*, add S to q_2.
```

Thus, after the first stage of our algorithm, we have searched or pruned all gridded regions (requiring at most $O((N \log N)^d)$ time), and the current S^* is the gridded region with highest $F(S)$. q_2 now contains the subset of gridded regions whose outer regions have not yet been pruned, prioritized by their upper bounds F_{bound}. The second stage of our algorithm proceeds as follows:

```
While q_2 not empty (and some S on q_2 has F_bound(S) > F*):
  Get region S with highest F_bound(S) from q_2.
  Use quartering to calculate tighter bounds on B(S) and C(S)/B(S).
  Recalculate F_bound(S) using these bounds.
  If F_bound(S) > F*, then search-outer-regions(S).
```

Now the only question left is how to perform the search-outer-regions procedure. We first note that a hyper-rectangular region requires $2d$ coordinates for specification: the minimum value x_i and size k_i in each dimension i. Thus a naïve search of the outer regions of S could be done using $2d$ nested loops, stepping over each legal combination of these coordinates (i.e. such that the resulting region S' is in S and contains S_C). Our procedure is similar to this, except that we take several more opportunities for pruning. Once we have fixed the values of $k_i(S')$ and $x_i(S')$ for a given i, we can obtain a tighter bound on $F(S')$ by *expanding* the center region S_C and *contracting* the parent region S such that $k_i(S) = k_i(S_C) = k_i(S')$ and $x_i(S) = x_i(S_C) = x_i(S')$. We then recalculate bounds on the baselines and rates for the new S and S_C using quartering, and finally recompute F_{bound} for the new parent and center. Only if the new value of F_{bound} is greater than F^* do we need to loop over k_{i+1} and x_{i+1} for that combination of k_i and x_i.

Thus the second stage of our algorithm can be seen as a series of “screens” that an outer region must pass through if it is to be searched. The first screen is whether the parent region is taken off q_2 and examined, the second screen is whether the parent region passes the quartering test, the third screen is whether the new parent region (formed after k_1 and x_1 are fixed) passes the quartering test, etc. We can show that the complexity of this procedure is $O(N^{2d})$ if all our screens fail, and better than $O(N^{2d})$ otherwise. Typically well over 90% of regions are eliminated at each screen, and thus we search only a small fraction of possible regions.

We now examine the complexity of this procedure in the two-dimensional case, given a large parent region (i.e. one containing $O(N^4)$ outer regions S'). If the parent region does not pass the first screen, we have spent only $O(1)$ to search these $O(N^4)$ regions; if the parent does not pass the second screen, we have spent only the $O(N^2)$ time required by quartering. If the parent passes the second screen, but none of the new parent regions pass through the third and fourth screens, we have spent only $O(N^2) \times O(N)$ (for quartering, given each k_1 and x_1) + $O(N^3)$ (for bounding scores, given each k_1, x_1, and k_2) $= O(N^3)$ time. Thus only if all four screens fail will the algorithm have $O(N^4)$ complexity.

3.4 Results

In our fast spatial scan work [112, 118], we have demonstrated that our fast spatial scan algorithm achieves huge speedups over the naïve approach both on real and simulated datasets, without any loss of accuracy (i.e. our algorithm finds exactly the same region and p-value as the naïve approach). These results include:

- 450-4700x speedups on 2D Emergency Department datasets, for grid resolutions ranging from 128×128 to 512×512 [112].
- 96-739x speedups on 2D OTC sales datasets, for grid resolutions ranging from 128×128 to 256×256 [112].
- 7-148x speedups on 3D fMRI imaging datasets, for grid resolution of $64 \times 64 \times 14$ voxels [118].
- 235-325x speedups on 4D Emergency Department datasets, using patient’s gender (2 values) and age decile (8 values) as pseudo-spatial dimensions and thus searching a $128 \times 128 \times 2 \times 8$ grid [118].
- 48-1400x speedups on 3D OTC sales datasets, searching for persistent spatial clusters (i.e. using date of sale as a pseudo-spatial dimension with 8 values) and thus searching a $128 \times 128 \times 8$ grid [118].

We have also done preliminary work comparing our algorithm to Kulldorff’s SaTScan software [87] (the current state of the art for detection of disease clusters) using Emergency Department data. This comparison suggests that we can detect elongated disease clusters 10-100x faster than

Table 3.1: Performance of algorithm, simulated datasets, $N = 256$. For each dataset, we give the time in seconds to search the original grid and each replica grid, as well as the number of regions searched. The speedup is the ratio of runtimes of the naïve and fast approaches.

test	method	sec/orig	speedup	sec/rep	speedup	regions (orig)	regions (rep)
all	naïve	3864	x1	3864	x1	1.03B	1.03B
7x9, 0.01	fast	5.47	x706	1.68	x2300	100K	1.20K
11x5, 0.002	fast	21.72	x178	12.43	x311	1.03M	196K
4x3, 0.002	fast	42.96	x90	40.57	x95	2.59M	1.87M
no region	fast	189.68	x20	110.25	x35	27.4M	12.7M

SaTScan can detect circular clusters, despite needing to search over a much larger space of possible clusters [114]. These results show that our method is sufficiently fast to be useful for the detection of significant spatial clusters, even in cases where the datasets are too large for other approaches to be feasible. We now present these results in detail in the following subsections.

3.4.1 Results for two-dimensional scan

We first describe results with artificially generated grids and then real-world case data. An artificial grid is generated from a set of parameters (N, k_1, k_2, μ, σ, q', q'') as follows. The grid generator first creates an $N \times N$ grid, and randomly selects a $k_1 \times k_2$ "test region." Then the baseline b_i of each grid cell (representing at-risk population) is chosen randomly from a normal distribution with mean μ and standard deviation σ (baselines less than zero are set to zero). Finally, the count of each grid cell is chosen randomly from a Poisson distribution with parameter qb_i, where the disease rate $q = q'$ inside the test region and $q = q''$ outside the test region.

For all our simulated tests, we used grid size $N = 256$, and a background disease rate of $q'' = .001$. We tested for three different combinations of test region parameters ($k_1 \times k_2$, q'): (7×9, .01), (11×5, .002), and (4×3, .002). These represent the cases of an extremely dense disease cluster, and large and small disease clusters which are significant but not extremely dense. We also ran a fourth test where no disease cluster was present, and thus $q = .001$ everywhere.

We used three different population distributions for testing: the "standard" distribution ($\mu = 10^4$, $\sigma = 10^3$), and two types of "highly varying" populations. For the "city" distribution, we randomly selected a 10×10 "city region": populations were generated with $\mu = 5 \times 10^4$ and $\sigma = 5 \times 10^3$ inside the city, and $\mu = 10^4$ and $\sigma = 10^3$ outside the city. For the "high-σ" distribution, we generated all populations with $\mu = 10^4$ and $\sigma = 5 \times 10^3$. For each combination of test region parameters and population distribution, run times were averaged over 20 random trials. We also ran an additional 90 trials (for a total of 110) to test accuracy, confirming that the algorithm found the highest scoring region in all cases. We also recorded the average number of regions examined; for our algorithm, this includes calculation of score bounds as well as scores of individual regions. Separate results are presented for the original grid and for each replica; for a large number of random replications ($R = 999$) the results per replica dominate, since total run time is $t_{orig} + R(t_{rep})$ to search the original grid and perform randomization testing. See Table 3.1 for results.

Our first observation was that the run time and number of regions searched were not significantly affected by the underlying population distribution; typically the three results differed by only 5-10%, and in many cases test regions were found *faster* for the highly varying distributions than the

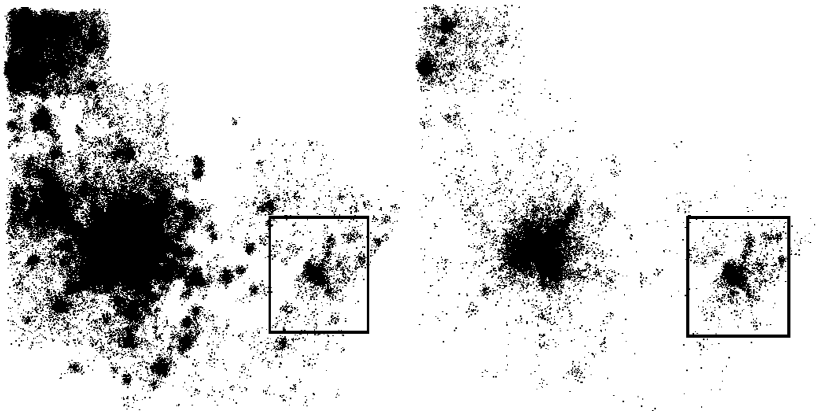


Figure 3.7: Emergency Department dataset. The left picture shows the baseline (population) distribution and the right picture shows the counts. The most significant region is shown as a rectangle.

standard distribution. Thus Table 3.1, rather than presenting separate results for each population distribution, presents the average performance over all three population distributions for each test. This result demonstrates the robustness of the algorithm to highly non-uniform baselines; this is very different than our previous work [111], where the algorithm was severely slowed by highly varying baselines. The algorithm achieved average speedups ranging from 35x (for no test region), to 2300x (for an extremely dense test region) as compared to the naïve approach. We note that, for the case of no test region, it is typically not necessary to run more than 10-20 randomizations before concluding with high probability that the discovered region is not significant. For example, if four or more of the first ten replicas beat the original grid, we know that this result will only occur 0.1% of the time if the region is significant, so we can safely assume that the region is not significant. Thus our true average "worst-case" results will be closer to the 95x speedup on small, significant (but not extremely dense) test regions. Since the naïve approach requires approximately 45 days for a 256×256 grid with $R = 999$, this suggests that our algorithm can complete the same task in less than 12 hours.

We now discuss the performance of the algorithm on various real-world datasets. Our first test set was a database of anonymized Emergency Department data collected from Western Pennsylvania hospitals in the period 1999-2002. This dataset contained a total of 630,000 records, each representing a single ED visit and giving the latitude and longitude of the patient's home location to the nearest 0.005 degrees ($\sim \frac{1}{3}$ mile). These locations were mapped to three grid sizes: $N = 128$, 256, and 512. For each grid, we tested for spatial clustering of "recent" disease cases: the count of a grid cell was the number of ED visits in that area in the last two months, and the baseline of a cell was the total number of ED visits in that cell in the entire four years of data. See Figure 3.7 for a picture of this dataset, including the highest scoring region. For each of these grids, our fast algorithm found the same, statistically significant region (p-value 1/1000) as the naïve approach. The major difference, of course, was in runtime and number of regions searched (see Table 3.2). Our algorithm found the most significant region of the original grids 22-24x faster than the naïve approach; however, much faster performance was achieved when searching the replica grids. The algorithm achieved speedups increasing from 450x to 4700x as grid size increased from 128 to 512.

Table 3.2: Performance of algorithm, real-world datasets. For each dataset, we give the time in seconds to search the original grid and each replica grid, as well as the number of regions searched. The speedup is the ratio of runtimes of the naïve and fast approaches.

test	method	sec/orig	speedup	sec/rep	speedup	regions (orig)	regions (rep)
ED	naïve	72	x1	68	x1	62.0M	62.0M
($N = 128$)	fast	3	x24	0.15	x453	5.12M	15.9K
ED	naïve	1207	x1	1185	x1	1.03B	1.03B
($N = 256$)	fast	55	x22	1.2	x988	95.9M	74.7K
ED	naïve	19146	x1	18921	x1	16.8B	16.8B
($N = 512$)	fast	854	x22	4.0	x4730	1.51B	120K
national OTC	naïve	71	x1	77	x1	62.0M	62.0M
($N = 128$)	fast	2	x36	0.8	x96	682K	200K
national OTC	naïve	1166	x1	1232	x1	1.03B	1.03B
($N = 256$)	fast	14	x96	2.8	x440	3.24M	497K
regional OTC	naïve	78	x1	79	x1	62.0M	62.0M
($N = 128$)	fast	2	x39	0.6	x132	783K	101K
regional OTC	naïve	1334	x1	1330	x1	1.03B	1.03B
($N = 256$)	fast	13	x103	1.8	x739	3.10M	168K

Our second test set was a nationwide database of retail sales of over-the-counter cough and cold medication. Sales figures were reported by zip code; the data covered 5000 zip codes across the U.S., with highest coverage in the Northeast. In this case, our goal was to see if the spatial distribution of sales on a given day (2/14/2004) was significantly different than the spatial distribution of sales a week before (2/7/2004), and to identify a significant cluster of increased sales if one exists. Note that the population-based statistic used in this test adjusts for increases or decreases in the total number of sales; clusters are only detected if there is spatial variation in the amount of increase or decrease. Thus we used the sales on 2/7 as our underlying baseline distribution, and the sales on 2/14 as our count distribution. We created four grids from this data, two using all of the national data, and two using only data from the Northeast (where a greater proportion of zip codes report sales data). For both "national" and "regional" over-the-counter data, we created grids of sizes $N = 128$ and $N = 256$, converting each zip code's centroid to a latitude and longitude. For each of these four grids, our algorithm found the same statistically significant region (p-value 1/1000) as the naïve approach, and achieved speedups of 96-132x on the 128×128 grids and 440-739x on the 256×256 grids.

Thus the algorithm found the most significant region in all of our simulated and real-world trials, while achieving speedups of at least 20x (and typically much larger) as compared to the naïve approach. This speedup is extremely important for the real-time detection of disease outbreaks: if a system is able to detect an outbreak in minutes rather than days, preventive measures or treatments can be administered earlier, decreasing rates of morbidity and mortality. We believe that our algorithm will be useful for rapid detection of significant spatial clusters in a variety of other applications as well.

3.4.2 Comparison to SaTScan

It is difficult to evaluate the computational speed of an algorithm in isolation, and thus a comparison to other techniques in the literature is necessary. We note, however, that none of the prior algorithmic work on scan statistics allows for the efficient detection of *elongated* clusters; the detection of compact clusters (e.g. circles or squares) is a significantly easier computational task, since there is one less degree of freedom to search over. Thus the most accurate comparison is to the obvious technique of naïvely searching all rectangles; this comparison was done in the previous section. However, since no available software actually uses this "naïve rectangles" approach, we feel that a comparison to other techniques (though inexact at best) will be useful.

In particular, we focus on Martin Kulldorff's SaTScan software [87]. SaTScan represents the current state-of-the-art in cluster detection, and is widely used in the epidemiological community. We emphasize that this is not an "apples to apples" comparison: because of the inexactness of this comparison and the inherent differences between the two methods of cluster detection, it is difficult to draw general conclusions. In particular, there are three main differences between the methods. First, as noted above, our algorithm searches for elongated clusters (in particular, axis-aligned rectangles) while SaTScan searches for compact clusters (in particular, circles). Thus (assuming that M is the number of distinct spatial locations) our algorithm must search over the $O(M^4)$ possible rectangles, while SaTScan must search over the $O(M^3)$ possible circles. Second, neither our algorithm nor SaTScan actually searches over "all" of the regions of the given type (rectangles or circles). SaTScan searches only circles centered at one of the data points, reducing the search space to $O(M^2)$ regions. Our method, on the other hand, aggregates the data points to a uniform $N \times N$ grid, and searches over the $O(N^4)$ gridded rectangular regions. Thus our method's runtime is a function of the grid resolution N, while SaTScan's runtime is a function of the number of spatially distinct data points M. If each data point truly represents cases occurring at that precise spatial location, we are losing some precision by aggregating points to a grid; however, this loss of precision is minimal for high grid resolutions N. Also, in cases where data points are derived from regions (e.g. representing a census tract or zip code by a point mass at the center of that region) then the assumption of discrete data points is itself somewhat inexact. Finally, both our method and SaTScan use clever computational techniques to speed up performance: our pruning method allows us to search only a small subset of the $O(N^4)$ gridded rectangular regions, while obtaining the same results as if we had searched all of these regions. SaTScan, though it does not use pruning to speed up the search (and thus, must actually search over all of the $O(M^2)$ regions), uses an "incremental addition" technique which allows searching in constant time per region. We also achieve constant search time per region, using the "cumulative counts" trick noted above.

As a simple comparison, we ran both our method and SaTScan on the Emergency Department dataset discussed above. This dataset consisted of 630,000 records, of which the last 60,000 records (recent data) were used as "counts" and the entire dataset was used as baselines. Since many records corresponded to identical spatial locations, this gave us approximately $M = 17,000$ distinct spatial locations. We ran both our method and SaTScan on this dataset, using the same system (Pentium 4, 1800 MHz processor, 1 GB RAM) for each. For all runs, we used 999 Monte Carlo replications. Our system found the most significant rectangular region in 11 minutes for a 128×128 grid and 81 minutes for a 256×256 grid, computing a p-value of 1/1000 in each case. SaTScan ran out of memory and thus was unable to find the most significant circular region for this dataset; in comparison, our method requires very little memory (< 50 MB for grid sizes up to 256×256). Thus we instead ran SaTScan on one tenth of the data (60,000 records, 10,000 used as "count"), containing $M = 8,400$ distinct spatial locations. In this case, SaTScan found the most significant circular region in 4 hours; this suggests that (given sufficient memory) it would find the most significant circular region for the entire dataset in approximately 16.5 hours.[5]

We note that, for the smaller dataset, both methods found very similar spatial regions. SaTScan found a circle with center coordinates (40.34 N latitude, 79.82 W longitude) and diameter 18.58 km, with $C = 2458$, $B = 8443$, and a score (log-likelihood ratio) of 413.56. For a 128×128 grid size, our method found a rectangle with almost the same centroid (40.32 N latitude, 79.82 W longitude), and size 23.6×17.2 km. This slightly larger region had $C = 2599$, $B = 9013$, and a score of 429.85. In this case, the most significant rectangular region has a low aspect ratio, so as expected, the region and score are similar to that found by SaTScan. If, on the other hand, the most significant rectangular region has a high aspect ratio, we would expect our algorithm to find a region with a significantly higher score.

We emphasize again that this comparison between our method and SaTScan is both preliminary (testing only on a small sample of datasets) as well as inexact (because of the differences between the algorithms discussed above). Thus we do not attempt to draw any general conclusions about the relative speeds of the two methods; we note only that our "fast spatial scan" is able to find elongated clusters in times comparable to (and in at least some cases, significantly faster than) the detection of compact clusters by SaTScan. Since SaTScan is in wide use in the epidemiological community, this demonstrates that the runtime of our method is sufficiently fast to be useful for the detection of significant spatial clusters.

Finally, we note another recently developed method that allows fast approximate computation of spatial scan statistics. Agarwal et al. [3] present a method of approximately computing Kulldorff's spatial scan statistic as a sum of linear discrepancy functions. This allows the scan statistic to be computed (within additive error ϵ) for axis-aligned rectangles in time $O(\frac{1}{\epsilon}N^3 \log N)$. It is likely that this method will also allow fast cluster detection, but at some cost in accuracy. In particular, we are not guaranteed to find the most significant region, and also, the significance results obtained by randomization will be less reliable. How much impact these factors have on the overall reliability of cluster detection using their method has not yet been determined.

3.4.3 Results for multi-dimensional fast spatial scan

We now describe results of our fast spatial scan algorithm on three sets of multi-dimensional real-world data: two sets of epidemiological data (from emergency department visits and over-the-counter drug sales), and one set of fMRI brain imaging data. Before presenting these results, we wish to emphasize three main points. First, the extension of scan statistics from two-dimensional to d-dimensional datasets dramatically increases the scope of problems for which these techniques can be used. As discussed above, in addition to datasets with more than two spatial dimensions (for example, the fMRI data), we can also examine data with a temporal component (as in the OTC dataset), or where we wish to take demographic information into account (as in the ED dataset). Second, in all of these datasets, the use of thresholded scan statistics (discussed in Chapter 2) instead of the classical scan statistic allows us to focus our search on smaller, denser regions rather than slight (but statistically significant) increases over a large area. Third, as our results here will

[5] We ran the default version of SaTScan. This uses unique data locations of which there were 17,000 in the full dataset, as candidate region centers. It is also possible to run SaTScan on a user-specified grid of candidate region centers. Perhaps that mode might be faster? In fact, the number of unique centers in the experiments reported above is approximately equal to the number of centers on a 128 by 128 grid, and considerably less than that in a 256 by 256 grid. Thus SaTScan would not be accelerated by switching to a grid approach.

demonstrate, the fast spatial scan gains huge performance improvements over the naïve approach, making the use of the scan statistic feasible in these large, real-world datasets.

Our first test set was the database of anonymized Emergency Department data collected from Western Pennsylvania hospitals in the period 1999-2002, as discussed above. This dataset contains a total of 630,000 records, each representing a single ED visit and giving the latitude and longitude of the patient's home location to the nearest $\frac{1}{3}$ mile. Additionally, a record contains information about the patient's gender and age decile. Thus we map records into a four-dimensional grid, consisting of two spatial dimensions (longitude, latitude) and two "pseudo-spatial" dimensions (patient gender and age decile). This has several advantages over the traditional (two-dimensional) spatial scan. First, our test has higher power to detect syndromes which affect differing patient demographics to different extents. For example, if a disease primarily strikes male infants, we might find a cluster with gender = male and age decile = 0 in some spatial region, and this cluster may not be detectable from the combined data. Second, our method accounts correctly for multiple hypothesis testing. If we were to instead perform a separate test at level α on each combination of gender and age decile, the overall false positive rate would be much higher than α. We mapped the ED dataset to a $128 \times 128 \times 2 \times 8$ grid, with the first two coordinates corresponding to longitude and latitude, the third coordinate corresponding to the patient's gender, and the fourth coordinate corresponding to the patient's age decile. We tested for spatial clustering of "recent" disease cases: the count of a cell was the number of ED visits in that spatial region, for patients of that age and gender, in 2002, and the baseline was the total number of ED visits in that spatial region, for patients of that age and gender, over the entire temporal period 1999-2002. To find such clusters, we used the discriminative thresholded scan statistic discussed in Chapter 2, with values of the threshold parameter ϵ ranging from 0 to 1.0. For the classical scan statistic ($\epsilon = 0$), we found a region of size $35 \times 34 \times 2 \times 8$; thus the most significant region was spatially localized but cut across all genders and age groups. The region had $C = 3570$ and $B = 6409$, as compared to $\frac{C}{B} = 0.05$ outside the region, and thus this is clearly an overdensity of counts. This was confirmed by the algorithm, which found the region statistically significant (p-value 1/101). With the three other values of ϵ, the algorithm found almost the same region ($35 \times 33 \times 2 \times 8$, $C = 3566$, $B = 6390$) and again found it statistically significant (p-value 1/101). For all values of ϵ, the fast scan statistic found the most significant region hundreds of times faster than the naïve spatial scan (see Table 3.3): while the naïve approach required approximately 12 hours per replication, the fast scan searched each replica in approximately 2 minutes, plus 100 minutes to search the original grid. Thus the fast algorithm achieved speedups of 235-325x over the naïve approach for the entire run (i.e. searching the original grid and 100 replicas) on the ED dataset.

Our second test set was a nationwide database of retail sales of over-the-counter cough and cold medication. Sales figures were reported by zip code; the data covered 5000 zip codes across the U.S. In this case, our goal was to see if the spatial distribution of sales in a given week (February 7-14, 2004) was significantly different than the spatial distribution of sales during the previous week, and to identify a significant cluster of increased sales if one exists. Since we wanted to detect clusters even if they were only present for part of the week, we used the date (Feb. 7-14) as a third dimension. This is similar to the retrospective space-time scan statistic of [82], which also uses time as a third dimension. However, that algorithm searches over cylinders rather than hyper-rectangles, and thus cannot detect spatially elongated clusters. The count of a cell was taken to be the number of sales in that spatial region on that day; to adjust for day-of-week effects, the baseline of a cell was taken to be the number of sales in that spatial region on the day one week prior (Jan. 31-Feb. 7). Thus

Table 3.3: Performance of algorithm, multi-dimensional real-world datasets

test	ϵ	sec/orig	sec/rep	speedup	regions (orig)	regions (rep)
ED	0	6140	126	x235	358M	622K
$(128 \times 128 \times 2 \times 8)$	0.25	6035	100	x275	352M	339K
(7.35B regions)	0.5	5994	102	x272	348M	362K
	1.0	5607	79.6	x325	334M	336K
OTC	0	4453	195	x48	302M	2.46M
$(128 \times 128 \times 8)$	0.25	429	123	x90	12.2M	1.39M
(2.45B regions)	0.5	334	51	x210	8.65M	350K
	1.0	229	5.9	x1400	4.40M	< 10
fMRI	0	880	384	x7	39.9M	14.0M
$(64 \times 64 \times 14)$	0.01	597	285	x9	35.2M	10.4M
(588M regions)	0.02	558	188	x14	33.1M	6.65M
	0.03	547	97.3	x27	32.3M	3.93M
	0.04	538	30.0	x77	31.9M	1.44M
	0.05	538	13.1	x148	31.7M	310K

we created a $128 \times 128 \times 8$ grid, where the first two coordinates were derived from the longitude and latitude of that zip code, and the third coordinate was temporal, based on the date. For this dataset, the classical scan statistic ($\epsilon = 0$) found a region of size 123×76 from February 7-11. Unfortunately, since the rate $\frac{C}{B}$ was only 0.99 inside the region (as compared to 0.96 outside) this region would not be interesting to an epidemiologist. Nevertheless, the region was found to be significant (p-value 1/101) because of the large total baseline. Thus, in this case, the classical scan statistic finds a large region of very slight overdensity rather than a smaller, denser region, and thus is not as useful for detecting epidemics. For $\epsilon = 0.25$ and $\epsilon = 0.5$, the scan statistic found a much more interesting region: a 4×1 region on February 9 where $C = 882$ and $B = 240$. In this region, the number of sales of cough medication was 3.7x its expected value; the region's p-value was computed to be 1/101, so this is a significant overdensity. For $\epsilon = 1$, the region found was almost the same, consisting of three of these four cells, with $C = 825$ and $B = 190$. Again the region was found to be significant (p-value 1/101). For this dataset, the naïve approach took approximately three hours per replication. The fast scan statistic took between six seconds and four minutes per replication, plus ten minutes to search the original grid, thus obtaining speedups of 48-1400x on the OTC dataset. We note that higher values of the threshold ϵ, in addition to focusing our search on more relevant regions, also allow the fast spatial scan to do more pruning, thus achieving significantly faster run times.

Our third and final test set was a set of fMRI data, consisting of two "snapshots" of a subject's brain under null and experimental conditions respectively. The experimental condition was from a test by Mitchell et al. [103] where the subject is given words, one at a time; he must read these words and identify them as verbs or nouns. The null condition is the subject's average brain activity while fixating on a cursor, before any words are presented. Each snapshot consists of a $64 \times 64 \times 14$ grid of voxels, with a reading of fMRI activation for the subset of the voxels where brain activity is occurring. In this case, the count of a cell is the fMRI activation for that voxel under the experimental condition, and the baseline is the activation for that voxel under the null condition. For voxels with no brain activity, we have $c_i = b_i = 0$. For the fMRI dataset, the amount of change between activated and non-activated regions is small, and thus we used values of ϵ ranging from 0 to 0.05 as suggested by the fMRI literature.

For the classical scan statistic ($\epsilon = 0$) our algorithm found a $23 \times 20 \times 11$ region, and again found this region significant (p-value 1/101). However, this is another example where the classical scan statistic finds a region which is large ($\frac{1}{4}$ of the entire brain) and only slightly increased in count: $\frac{C}{B} = 1.007$ inside the region and $\frac{C}{B} = 1.002$ outside the region. For $\epsilon = 0.01$, we find a more interesting cluster: a $5 \times 10 \times 1$ region in the visual cortex containing four non-zero voxels. For this region $\frac{C}{B} = 1.052$, a large increase, and the region is significant at $\alpha = 0.1$ (p-value 10/101) though not at $\alpha = 0.05$. For $\epsilon = 0.02$, we find the same region, but conclude that it is not significant (p-value 32/101). For $\epsilon = 0.03$ and $\epsilon = 0.04$, we find a $3 \times 2 \times 1$ region with $\frac{C}{B} = 1.065$, but this region is not significant (p-values 61/101 and 89/101 respectively). Similarly, for $\epsilon = 0.05$, we find a single voxel with $\frac{C}{B} = 1.075$, but again it is not significant (p-value 91/101). For this dataset, the naïve approach took approximately 45 minutes per replication. The fast scan statistic took between 13 seconds and six minutes per replication, thus obtaining speedups of 7-148x on the fMRI dataset.

Thus we have demonstrated (through tests on a variety of real-world datasets) that the fast multidimensional spatial scan statistic has significant performance advantages over the naïve approach, resulting in speedups up to 1400x without any loss of accuracy. This makes it feasible to apply scan statistics in a variety of application domains, including the spatial and spatio-temporal detection of disease epidemics (taking demographic information into account), as well as the detection of regions of increased brain activity in fMRI data.

Chapter 4

Methods for space-time cluster detection

4.1 Introduction

This chapter extends our spatial cluster detection framework to the space-time case. While most of the prior work on cluster detection is purely spatial in nature (e.g. [4, 78, 49]), it is clear that the time dimension is an essential component of most cluster detection problems. Typically, we are interested in detecting clusters that are *emerging* in time, and our goal is to detect these emerging clusters as early as possible. For example, in the public health domain, our goal may be to detect emerging clusters of disease cases, which may be indicative of a naturally occurring disease outbreak (e.g. influenza), a bioterrorist attack (e.g. anthrax release), or an environmental hazard (e.g. radiation leak). In any case, early detection of such disease clusters can lead to earlier public health response, potentially saving many lives. In medical imaging, we may attempt to detect tumors or other hazardous growths, and early detection of such tumors may increase the patient's chance of survival. Finally, in military reconnaissance, the goal may be to monitor the strength and activity of enemy forces, and we may want to detect a buildup of troops that is indicative of an impending attack.

Kulldorff et al. [82] first proposed a variant of the spatial scan statistic for detection of space-time clusters, and applied scan statistics for prospective disease surveillance in [81]. The goal of the space-time scan statistic is a straightforward extension of the purely spatial scan: to detect regions of space-time where the counts are significantly higher than expected. Let us assume that we have a discrete set of time steps $t = 1 \ldots T$ (e.g. daily observations for T days), and for each spatial location s_i, we have counts c_i^t and baselines b_i^t representing the observed and expected number of cases in the given area on each time step. Then there are two very simple ways of extending the spatial scan to space-time: to run a separate spatial scan for each time step t, or to treat time as an extra dimension and thus run a single multidimensional spatial scan in space-time (for example, we could search over three-dimensional "hyper-rectangles" which represent a given rectangular region of space during a given time interval). The problem with the first method is that, by only examining one day of data at a time, we may fail to detect more slowly emerging clusters. The problem with the second method is that we tend to find less relevant clusters: for prospective surveillance, we want to detect newly emerging clusters, not those that have persisted for a long time. Thus, in order to achieve better methods for space-time cluster detection, we must consider the question, "How is the time dimension different from space?" We argue that there are three main distinctions:

1. The concept of "now". In the time dimension, the present is an important point of reference. For example, in disease surveillance, we are typically only interested in clusters that are still "active"

at the present time, and that have emerged within the recent past (e.g. within a few days or a week). We do not want to detect clusters that have persisted for months or years, and we are also not interested in those clusters which have already come and gone. The exception to this, of course, is if we are performing a retrospective analysis, attempting to detect all space-time clusters regardless of how long ago they occurred. The retrospective statistic searches over time intervals $t_{min} \ldots t_{max}$, where $1 \leq t_{min} \leq t_{max} \leq T$, while the prospective statistic searches over time intervals $t_{min} \ldots T$, where $1 \leq t_{min} \leq T$, adjusting correctly for multiple hypothesis testing in each case. We focus here on prospective analysis, since this is more relevant for our typical disease surveillance task.

2. "Learning from the past." In the spatial cluster detection framework given in Chapter 2, we typically assume that we have some baseline denominator data, such as an at-risk population, given in advance. In the space-time framework, on the other hand, we must infer the expected counts b_i^t of recent days from the time series of previous counts c_i^t, then use the expectation-based scan statistic (discussed in Chapter 2) to find space-time clusters where the counts are higher than expected. Thus the first major contribution of this chapter is an expectation-based space-time scan statistic approach. Inferring expectations from previous counts has several advantages over the standard method of relying on at-risk population: we can account for spatial variation in disease rate (due to factors such as age and health of population and environmental hazards) as well as the variation of disease rate over time (due to factors such as day of week and seasonality), and thus reduce the number of false positives due to these sources of variation in the baseline rate.

3. The "arrow of time." Time has a fixed directionality, moving from the past, through the present, to the future. We are often interested in clusters which *emerge* over time: for example, a disease may start out having only minor impact on the affected population, then increase its impact (and thus the observed symptom counts) either gradually or rapidly until it peaks. Based on this observation, the second major contribution of this chapter is a variant of the space-time scan statistic designed for more rapid detection of emerging outbreaks. The idea is that rather than assuming (as in the standard, "persistent" space-time scan statistic) that the disease rate q remains constant over the course of an epidemic, we expect the disease rate to increase over time, and thus we fit a model which assumes a monotonically increasing sequence of disease rates q_t at each affected time step t in the affected region.[1] We will show that this "emerging cluster" space-time scan statistic often outperforms the standard "persistent cluster" approach.

Taking these factors into account, the prospective space-time scan statistic has two main parts: inferring (based on past counts) what we expect the recent counts to be, and finding regions where the observed recent counts are significantly higher than expected. More precisely, given a "temporal window size" W, we wish to know whether any space-time cluster within the last W days has counts c_i^t higher than expected. To do so, we first infer the expected counts $b_i^t = E\left[c_i^t\right]$ for all spatial locations on each recent day t, $T - W < t \leq T$, then use a space-time scan statistic to find space-time clusters with higher than expected counts. These steps are described in detail below.

In the remainder of this chapter, I present our statistical and computational methods for the detection of space-time clusters. Section 4.2 describes our framework for space-time cluster detection, and how the generalized spatial scan framework can be adapted to the space-time case. Section 4.3 presents several variants of the space-time scan statistic, including methods for detecting persistent clusters, emerging clusters, and parametrized clusters. Section 4.4 discusses the inference of baseline values by time series analysis, Section 4.5 discusses computational considerations, and Section

[1] It is also possible to consider models where the disease rate varies not only in time but in space. Examples of models with spatially varying disease rate are given in Chapter 2.

4.6 discusses related work. Finally, I present results and discussion of our space-time methods in Sections 4.7 and 4.8.

Parts of this chapter have been adapted from our papers in KDD 2005 [120] and the 2005 National Syndromic Surveillance Conference [121], as well as a CMU technical report [109]. I wish to thank my co-authors Andrew Moore, Maheshkumar Sabhnani, and Kenny Daniel for their contributions to this work. Additionally, parts of this chapter have been adapted from our chapter in the *Handbook of Biosurveillance* [115]; I wish to thank my co-author Andrew Moore and editor Michael Wagner for their contributions.

4.2 Space-time cluster detection

In the general case, we have data collected at a set of discrete time steps $t = 1 \ldots T$ (where time T represents the present) at a set of discrete spatial locations s_i. For each s_i at each time step t, we are given a *count* c_i^t, and our goal is to find if there is any region S (set of locations s_i) and time interval ($t = t_{min} \ldots t_{max}$) for which the counts are significantly higher than expected. Thus we must first decide on the set of spatial regions S, and the time intervals $t_{min} \ldots t_{max}$, that we are interested in searching. In the scan statistics framework discussed below, we typically search over the set of all spatial regions of some given shape, and variable size. For simplicity, we assume here that the spatial locations s_i are aggregated to a uniform, two-dimensional, $N \times N$ grid G, and we search over the set of all axis-aligned rectangular regions $S \subseteq G$.[2] This allows us to detect both compact and elongated clusters, which is important since disease clusters may be elongated due to dispersal of pathogens by wind, water, or other factors. For prospective surveillance, as is our focus here, we care only about those clusters which are still present at the current time T, and thus we search over time intervals with $t_{max} = T$; if we were performing a retrospective analysis, on the other hand, we would search over all $t_{max} \leq T$. We must also choose the size of the "temporal window" W: we assume that we are only interested in detecting clusters that have emerged within the last W days (and are still present), and thus we search over time intervals $t_{min} \ldots T$ for all $T - W < t_{min} \leq T$.

In the disease detection framework, we assume that the count (number of cases) in each spatial region s_i on each day t is Poisson distributed, $c_i^t \sim \text{Poisson}(\lambda_i^t)$ with some unknown parameter λ_i^t. Thus our method consists of two parts: time series analysis for calculating the expected number of cases (or "baseline") $b_i^t = E[c_i^t]$ for each spatial region on each day, and space-time scan statistics for determining whether the actual numbers of cases c_i^t in some region S are significantly higher than expected (given b_i^t) in the last W days. The choice of temporal window size W impacts both parts of our method: we calculate the baselines b_i^t for the "current" days $T - W < t \leq T$ by time series analysis, based on the "past" days $1 \leq t \leq T - W$, and then determine whether there are any emerging space-time clusters in the last W days.

Our space-time scan statistic method is much like the purely spatial method: we choose the models H_0 and $H_1(S, t_{min})$, where the null hypothesis H_0 assumes no clusters and the alternative hypothesis $H_1(S, t_{min})$ represents a cluster in spatial region S starting at time t_{min} and continuing to the present time T. From our models, we can derive the corresponding score function $F(S, t_{min})$ using the likelihood ratio statistic, and then find the space-time cluster (S^*, t^*_{min}) which maximizes the score function F. Finally, we can compute the statistical significance (p-value) of this space-time cluster by randomization testing, as in the purely spatial approach. The performance of our

[2] Non-axis-aligned rectangles can be detected by examining multiple rotations of the data, as in [112].

space-time scan statistic method is affected by four main considerations: the size of the temporal window W, the type of space-time scan statistic used, the level on which the data is aggregated, and the method of time series analysis. We discuss these considerations in detail below.

4.3 Space-time scan statistics

One of the most important statistical tools for cluster detection is the *spatial scan statistic* [88, 78, 80]. This method searches over a given set of spatial regions, finding those regions which maximize a likelihood ratio statistic and thus are most likely to be generated under the alternative hypothesis of clustering rather than under the null hypothesis of no clustering. Randomization testing is used to compute the p-value of each detected region, correctly adjusting for multiple hypothesis testing, and thus we can both identify potential clusters and determine whether they are significant. The standard spatial scan algorithm [80] has two primary drawbacks: it is extremely computationally intensive, making it infeasible to use for massive real-world datasets, and only compact (circular) clusters are detected. In Chapter 3, we have addressed both of these problems by proposing the "fast spatial scan" algorithm [112, 118], which can rapidly search for elongated clusters (hyper-rectangles) in large multi-dimensional datasets. As noted above, we choose here to search over rectangular regions, using a space-time variant of the fast spatial scan as necessary to speed up our search.

In its original, population-based formulation [88, 78], the spatial scan statistic does not take time into account. Instead, it assumes a single count c_i (e.g. number of disease cases) for each spatial location s_i, as well as a given baseline b_i (e.g. at-risk population). Then the goal of the scan statistic is to find regions where the *rate* (or expected ratio of count to baseline) is higher inside the region than outside. The statistic used for this is the likelihood ratio $F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$, where the null hypothesis H_0 represents no clustering, and each alternative hypothesis $H_1(S)$ represents clustering in some region S. More precisely, under H_0 we assume a uniform disease rate q_{all}, such that $c_i \sim \text{Poisson}(q_{all}b_i)$ for all locations s_i. Under $H_1(S)$, we assume that $c_i \sim \text{Poisson}(q_{in}b_i)$ for all locations $s_i \in S$, and $c_i \sim \text{Poisson}(q_{out}b_i)$ for all locations $s_i \in G - S$, for some constants $q_{in} > q_{out}$. From this, we can derive an expression for $F(S)$ using the maximum likelihood estimates of q_{in}, q_{out}, and q_{all}: $F(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}}\right)^{-C_{all}}$, if $\frac{C_{in}}{B_{in}} > \frac{C_{out}}{B_{out}}$, and $F(S) = 1$ otherwise, where "in," "out," and "all" are the sums of counts and baselines for S, $G - S$, and G respectively. Then the most significant spatial region S is the one with the highest score $F(S)$; we denote this region by S^*, and its score by F^*. Once we have found this region by searching over the space of possible regions S, we must still determine its statistical significance, i.e. whether S^* is a significant spatial cluster. To adjust correctly for multiple hypothesis testing, we find the region's p-value by randomization: we randomly create a large number R of replica grids under the null hypothesis $c_i \sim \text{Poisson}(q_{all}b_i)$, and find the highest scoring region and its score for each replica grid. Then the p-value can be computed as $\frac{R_{beat}+1}{R+1}$, where R_{beat} is the number of replica grids with F^* higher than the original grid. If this p-value is less than some constant α (here $\alpha = .05$), we can conclude that the discovered region is unlikely to have occurred by chance, and is thus a significant spatial cluster; we can then search for secondary clusters. Otherwise, no significant clusters exist.

The formulation of the scan statistic that we use here is somewhat different, because we are interested not in detecting regions with higher rates inside than outside, but regions with higher *counts* than *expected*. This "expectation-based" framework is presented in Chapter 2, and we briefly review the approach here. Let us assume that baselines b_i represent the expected values of each count c_i; we discuss how to obtain these baselines below. Then we wish to test the null hypothesis H_0: all counts c_i are generated by $c_i \sim \text{Poisson}(b_i)$, against the set of alternative hypotheses $H_1(S)$: for spatial locations $s_i \in S$, all counts c_i are generated by $c_i \sim \text{Poisson}(qb_i)$, for some constant $q > 1$, and for all other spatial locations $s_i \in G - S$, all counts $c_i \sim \text{Poisson}(b_i)$. We then compute the likelihood ratio:

$$F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)} = \frac{\max_{q \geq 1} \prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(qb_i))}{\prod_{s_i \in S} \Pr(c_i \sim \text{Poisson}(b_i))}$$

$$= \frac{\max_{q \geq 1} \prod_{s_i \in S} (qb_i)^{c_i} e^{-qb_i}}{\prod_{s_i \in S} b_i^{c_i} e^{-b_i}} = \frac{\max_{q \geq 1} q^{C_{in}} e^{-qB_{in}}}{e^{-B_{in}}}$$

Using the maximum likelihood estimate of the parameter $q = \max\left(1, \frac{C_{in}}{B_{in}}\right)$, we obtain the score function $F(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} e^{B_{in} - C_{in}}$, if $C_{in} > B_{in}$, and $F(S) = 1$ otherwise. As before, we search over all spatial regions S to find the highest scoring region S^*. Then the statistical significance (p-value) of S^* can be found by randomization testing as before, where the replica grids are generated under the null hypothesis $c_i \sim \text{Poisson}(b_i)$.

4.3.1 The 1-day space-time scan statistic

To extend this spatial scan statistic to the prospective space-time case, the simplest method is to use a 1-day temporal window ($W = 1$), searching for clusters on only the present day $t = T$. Thus we wish to know whether there is any spatial region S with higher than expected counts on day T, given the actual counts c_i^T and expected counts b_i^T for each spatial location s_i. To do so, we compare the null hypothesis H_0: $c_i^T \sim \text{Poisson}(b_i^T)$ for all s_i, to the set of alternative hypotheses $H_1(S)$: $c_i^T \sim \text{Poisson}(qb_i^T)$ for all $s_i \in S$, for some constant $q > 1$, and $c_i^T \sim \text{Poisson}(b_i^T)$ elsewhere. Thus the statistic takes the same form as the purely spatial scan statistic, and we obtain: $F(S) = \left(\frac{C}{B}\right)^C e^{B-C}$, if $C > B$, and $F(S) = 1$ otherwise, where $C = \sum_{s_i \in S} c_i^T$ and $B = \sum_{s_i \in S} b_i^T$ denote the total count and total baseline of region S on time step T. Again, we search over all spatial regions S to find the highest scoring region S^* and its score F^*. To compute the p-value, we perform randomization testing as before, where each replica grid has counts c_i^T generated from $\text{Poisson}(b_i^T)$ and all other counts c_i^t ($t \neq T$) copied from the original grid.

4.3.2 Multi-day space-time scan statistics

While the 1-day prospective space-time scan statistic is very useful for detecting rapidly growing outbreaks, it may have difficulty detecting more slowly growing outbreaks, as noted above. For the multi-day prospective space-time scan statistics, we have some temporal window $W > 1$, and must determine whether any outbreaks have emerged within the most recent W days (and are still present). In other words, we wish to find whether there is any spatial region S with higher than expected counts on days $t_{min} \ldots T$, for some $T - W < t_{min} \leq T$. To do so, we first compute the expected counts b_i^t and the actual counts c_i^t for each spatial location s_i on each day $T - W < t \leq T$; we discuss how the baselines b_i^t are calculated in the following section. We then search over all spatial regions $S \subseteq G$, and all allowable values of t_{min}, finding the highest value of the spatio-temporal score function $F(S, t_{min})$. The calculation of this function depends on whether we are searching for "persistent" or "emerging" clusters, as we discuss below. In any case, once we have found the highest scoring region (S^*, t^*_{min}) and its score F^*, we can compute the p-value of this region by performing randomization testing as before, where each replica grid has counts c_i^t generated from Poisson(b_i^t) for $T - W < t \leq T$, and all other counts c_i^t copied from the original grid.

Now we must consider how to compute the function $F(S, t_{min})$. The standard population-based method for computing the space-time scan statistic, proposed for the retrospective case by [82] and for the prospective case by [81], builds on the Kulldorff spatial scan statistic [78] given above. As in the purely spatial scan, this method assumes that baselines b_i^t are given in advance (e.g. population in each location for each time interval), and that counts c_i^t are generated from Poisson distributions with means proportional to b_i^t. Then the goal is to find space-time clusters (S, t_{min}) where the rate (ratio of count to baseline) is significantly higher inside the region than outside. As in the purely spatial case, this can be adapted to our expectation-based framework, in which the goal is to find space-time clusters where the observed counts c_i^t are higher than the expected counts b_i^t. For the "persistent cluster" case, we maintain the other major assumption of the standard model: that the multiplicative increase in counts ("relative risk") in an affected region remains constant through the temporal duration of the cluster. For the "emerging cluster" case, we instead make the assumption that the relative risk increases monotonically through the cluster's duration. It is also possible to assume a parametric form for the increase in relative risk over time (e.g. exponential or linear increase), as we discuss below.

4.3.3 Persistent clusters

The test for persistent clusters assumes that the relative risk of a cluster remains constant over time; as a result, the score function is very similar to the 1-day statistic, with sums taken over the entire duration of a cluster rather than only a single day.

As noted above, we must search over all spatial regions S and all values of t_{min} (where $T - W < t_{min} \leq T$), finding the maximum score $F(S, t_{min})$. For a given region S and value t_{min}, we compare the null hypothesis H_0: $c_i^t \sim$ Poisson(b_i^t) for all spatial locations s_i and all $T - W < t \leq T$, to the set of alternative hypotheses $H_1(S, t_{min})$: $c_i^t \sim$ Poisson(qb_i^t) for $s_i \in S$ and $t = t_{min} \ldots T$, for some constant $q > 1$, and $c_i^t \sim$ Poisson(b_i^t) elsewhere. Thus we can compute the likelihood ratio:

$$F(S, t_{min}) = \frac{\max_{q \geq 1} \prod \Pr(c_i^t \sim \text{Poisson}(qb_i^t))}{\prod \Pr(c_i^t \sim \text{Poisson}(b_i^t))} = \frac{\max_{q \geq 1} \prod (qb_i^t)^{c_i^t} e^{-qb_i^t}}{\prod (b_i^t)^{c_i^t} e^{-b_i^t}}$$

where the products are taken over $s_i \in S$ and $t_{min} \leq t \leq T$. This simplifies to $\max_{q \geq 1} \frac{q^C e^{-qB}}{e^{-B}}$, where C and B are the total count $\sum_{s_i \in S} \sum_{t_{min} \leq t \leq T} c_i^t$ and total baseline $\sum_{s_i \in S} \sum_{t_{min} \leq t \leq T} b_i^t$ respectively. Finally, using the maximum likelihood estimate $q = \max\left(1, \frac{C}{B}\right)$, we obtain $F(S, t_{min}) = \left(\frac{C}{B}\right)^C e^{B-C}$ if $C > B$, and $F = 1$ otherwise.

4.3.4 Emerging clusters

While the space-time scan statistic for persistent clusters assumes that relative risk of a cluster remains constant through its duration, this is typically not true in disease surveillance. When a disease outbreak occurs, the disease rate will typically rise continually over the duration of the outbreak until the outbreak reaches its peak, at which point it will level off or decrease. Our main goal in the epidemiological domain is to detect emerging outbreaks (i.e. those that have not yet reached their peak), so we focus on finding clusters where the relative risk is monotonically increasing over the duration of the cluster. Again, we must search over all spatial regions S and all values of t_{min} (where $T - W < t_{min} \leq T$), finding the maximum score $F(S, t_{min})$. For a given region S and value t_{min}, we compare the null hypothesis H_0: $c_i^t \sim$ Poisson(b_i^t) for all spatial locations s_i and all $T - W < t \leq T$, to the set of alternative hypotheses $H_1(S, t_{min})$: $c_i^t \sim$ Poisson$(q_t b_i^t)$ for $s_i \in S$ and $t = t_{min} \dots T$, for some monotonically increasing sequence of constants $1 \leq q_{t_{min}} \leq \dots \leq q_T$, and $c_i^t \sim$ Poisson(b_i^t) elsewhere. Thus we can compute the likelihood ratio:

$$F(S, t_{min}) = \frac{\max_{1 \leq q_{t_{min}} \leq \dots \leq q_T} \prod \text{Pr}(c_i^t \sim \text{Poisson}(q_t b_i^t))}{\prod \text{Pr}(c_i^t \sim \text{Poisson}(b_i^t))}$$

$$= \frac{\max_{1 \leq q_{t_{min}} \leq \dots \leq q_T} \prod (q_t b_i^t)^{c_i^t} e^{-q_t b_i^t}}{\prod (b_i^t)^{c_i^t} e^{-b_i^t}}$$

$$= \frac{\max_{1 \leq q_{t_{min}} \leq \dots \leq q_T} \prod_{t = t_{min} \dots T} q_t^{C_t} e^{-q_t B_t}}{e^{-B}}$$

where C_t and B_t are the total count $\sum_{s_i \in S} c_i^t$ and the total baseline $\sum_{s_i \in S} b_i^t$ on day t, and B is the total baseline $\sum_{s_i \in S} \sum_{t_{min} \leq t \leq T} b_i^t$ as above.

Now, we must maximize the numerator subject to the monotonicity constraints on the q_t. To do so, let $E = E_1 \dots E_p$ be a partitioning of $t_{min} \dots T$ into sets of consecutive integers, such that for all $t_1, t_2 \in E_j$, $q_{t_1} = q_{t_2} = Q_j$, and for all E_{j_1} and E_{j_2}, where $j_1 < j_2$, $Q_{j_1} < Q_{j_2}$. In other words, the E_j define a partitioning of $t_{min} \dots T$ into time periods where the relative risk is constant. Note that the q_t are uniquely defined by the partitions E_j and the rates Q_j. We can then write:

$$F(S, t_{min}) = \frac{\max_{E_1 \dots E_p} \max_{1 \leq Q_1 < \dots < Q_p} \prod_{E_j} (Q_j)^{C_j} e^{-Q_j B_j}}{e^{-B}}$$

where $B_j = \sum_{s_i \in S} \sum_{t \in E_j} b_i^t$ and $C_j = \sum_{s_i \in S} \sum_{t \in E_j} c_i^t$. We now prove several lemmas which will help us to simplify this expression.

Lemma 4.3.1 *A necessary condition for (E, Q) to maximize $F(S, t_{min})$ is that $Q_j = \frac{C_j}{B_j}$ for all j.*

Proof Let us assume a fixed partitioning $E = \{E_j\}$, with strictly increasing Q_j, and ask whether the Q_j are optimal for those E_j. We note that, in the absence of constraints on the Q_j, each expression $e^{-Q_j B_j} (Q_j)^{C_j}$ is maximized at $Q_j = \frac{C_j}{B_j}$. Moreover, the score is convex with respect to Q_j. Thus, if some $Q_j < \frac{C_j}{B_j}$, we can increase the score by raising that Q_j slightly (without changing the ordering of Q_j), so the given Q_j are not optimal. Similarly, if some $Q_j > \frac{C_j}{B_j}$, we can increase the score by lowering that Q_j slightly (without changing the ordering of Q_j), so the given Q_j are not optimal. Thus for the Q_j to be optimal, we must have $Q_j = \frac{C_j}{B_j}$ for all j. ▮

Lemma 4.3.2 *A necessary condition for (E, Q) to maximize $F(S, t_{min})$ is that for all $j_1 < j_2$, $\frac{C_{j_1}}{B_{j_1}} < \frac{C_{j_2}}{B_{j_2}}$.*

Proof Otherwise either $Q_{j_1} \neq \frac{C_{j_1}}{B_{j_1}}$, or $Q_{j_2} \neq \frac{C_{j_2}}{B_{j_2}}$, or $Q_{j_1} \geq Q_{j_2}$. In the first two cases, the condition of Lemma 4.3.1 is violated, so the Q_j are not optimal. In the third case, the restriction that the Q_j are strictly increasing is violated, so the Q_j are not legal. $\blacksquare$

Thus we can write:

$$F(S, t_{min}) = \frac{\max_{E_1 \dots E_p} \prod_{E_j} e^{-C_j} \left(\frac{C_j}{B_j}\right)^{C_j}}{e^{-B}} = e^{B-C} \max_{E_1 \dots E_p} \prod_{E_j} \left(\frac{C_j}{B_j}\right)^{C_j}$$

where the partitioning $E = \{E_j\}$ must satisfy the condition of Lemma 4.3.2, i.e. the ratios $\frac{C_j}{B_j}$ are strictly increasing with j.

Finally, we give an algorithm which produces the optimal partitioning $E = \{E_j\}$. This method uses a stack data structure, where each element of the stack represents a partition E_j by a 5-tuple $(t_{start}, t_{end}, C_j, B_j, Q_j)$. The algorithm starts by pushing the 5-tuple $\left(T, T, C_T, B_T, \max\left(1, \frac{C_T}{B_T}\right)\right)$ onto the stack. Then for each t, from $T - 1$ down to t_{min}, we do the following:

```
temp = (t, t, C_t, B_t, max(1, C_t / B_t))
while (temp.Q >= stack.top.Q)
  temp2 = stack.pop
  temp = (temp.start, temp2.end, temp.C+temp2.C, temp.B +
    temp2.B, max(1, (temp.C+temp2.C) / (temp.B+temp2.B)))
stack.push(temp)
```

We now prove that this method produces the unique optimal partitioning E and rates Q, and thus the values of q_t that maximize the score subject to the monotonicity constraints above.

Lemma 4.3.3 *A necessary condition for the partitioning E to maximize $F(S, t_{min})$ is that for each $E_j = t_1 \dots t_2$, for all t such that $t_1 \leq t < t_2$, we have $\frac{\sum_{k=t_1 \dots t} C_k}{\sum_{k=t_1 \dots t} B_k} \geq \frac{\sum_{k=t+1 \dots t_2} C_k}{\sum_{k=t+1 \dots t_2} B_k}$.*

Proof Otherwise there exists some $E_j = t_1 \dots t_2$, and some t such that $t_1 \leq t < t_2$, where $\frac{\sum_{k=t_1 \dots t} C_k}{\sum_{k=t_1 \dots t} B_k} < Q_j < \frac{\sum_{k=t+1 \dots t_2} C_k}{\sum_{k=t+1 \dots t_2} B_k}$ (note that Q_j is a weighted average of the two ratios). We can now increase the score by separating E_j into two partitions $E_{j_1} = t_1 \dots t$ and $E_{j_2} = t + 1 \dots t_2$, where Q_{j_1} is slightly less than Q_j, and Q_{j_2} is slightly more than Q_j (without otherwise changing the order of Q_j). Thus E is not optimal. $\blacksquare$

Lemma 4.3.4 *A partitioning E satisfying the conditions of Lemmas 4.3.2 and 4.3.3 is unique, and thus that partitioning is optimal.*

Proof Assume two partitionings E^1 and E^2 satisfying the conditions of Lemmas 4.3.2 and 4.3.3. Consider the first j such that $E^1_j \neq E^2_j$. Let $E^1_j = t_0 \dots t_1$ and $E^2_j = t_0 \dots t_2$, assuming without loss of generality that $t_1 > t_2$. Now consider the first $k > j$ such that $E^2_k = t_3 \dots t_4$ and $t_4 \geq t_1$. Thus we have $t_0 \leq t_2 < t_3 \leq t_1 \leq t_4$. Let us write $\mu(t_0 \dots t_2) = \frac{\sum_{k=t_0 \dots t_2} C_k}{\sum_{k=t_0 \dots t_2} B_k}$ and define the other $\mu(\cdot)$ similarly. Applying the condition of Lemma 4.3.2 to E^2, we know $\mu(t_0 \dots t_2) < \mu(t_3 \dots t_4)$. Also, if $t_2+1 < t_3$, we know $\mu(t_0 \dots t_2) < \mu(t_2+1 \dots t_3-1) < \mu(t_3 \dots t_4)$. Applying the condition of Lemma 4.3.3 to E^2, we know that if $t_1 < t_4$, we have $\mu(t_3 \dots t_1) \geq \mu(t_3 \dots t_4) \geq \mu(t_1+1 \dots t_4)$. From these inequalities, we know $\mu(t_3 \dots t_1) > \mu(t_0 \dots t_3-1)$. But applying the condition of Lemma 4.3.3 to E^1, we know $\mu(t_0 \dots t_3-1) \geq \mu(t_3 \dots t_1)$, which is a contradiction. Thus the partitioning satisfying the conditions of Lemmas 4.3.2 and 4.3.3 is unique. Since these are necessary conditions for optimality, and a unique partitioning satisfies these conditions, we know that the partitioning is optimal. ∎

Theorem 4.3.5 *The method presented above maximizes $F(S, t_{min})$ subject to the monotonicity constraints.*

Proof We first note that the method satisfies the conditions of Lemma 4.3.1 (since $Q_j = \frac{C_j}{B_j}$ for each partition E_j), and Lemma 4.3.2 (since the while loop ensures the ordering of Q_j). To show that the method satisfies the condition of Lemma 4.3.3, we show that each new partition created by the "merge step" temp = (temp.start, temp2.end, ...) maintains this condition as an invariant. Let $E_{temp} = t_0 \dots t_1$, and $E_{temp2} = t_1+1 \dots t_2$. We know that E_{temp} and E_{temp2} satisfy the condition of Lemma 4.3.3, and we must show that the merged partition E_{new} also satisfies this condition. In other words, we are given $\mu(t_0 \dots j) \geq \mu(j+1 \dots t_1)$ for all j ($t_0 \leq j < t_1$), and $\mu(t_1+1 \dots j) \geq \mu(j+1 \dots t_2)$ for all j ($t_1+1 \leq j < t_2$). We also know that temp.Q is at least temp2.Q, since the merge step only takes place if this condition holds, so $\mu(t_0 \dots t_1) \geq \mu(t_1+1 \dots t_2)$. To show that the merged partition satisfies the condition of Lemma 4.3.3, we must show that $\mu(t_0 \dots j) \geq \mu(j+1 \dots t_2)$ for all j ($t_0 \leq j < t_2$). We know this is true for $j = t_1$, but must also prove it for $j < t_1$ and $j > t_1$. For $j < t_1$, we have $\mu(t_0 \dots j) \geq \mu(t_0 \dots t_1) \geq \mu(j+1 \dots t_1)$ and $\mu(t_0 \dots t_1) \geq \mu(t_1+1 \dots t_2)$. Thus $\mu(t_0 \dots j) \geq \mu(j+1 \dots t_1)$ and $\mu(t_0 \dots j) \geq \mu(t_1+1 \dots t_2)$, so $\mu(t_0 \dots j) \geq \mu(j+1 \dots t_2)$ as desired. For $j > t_1$, we have $\mu(t_1+1 \dots j) \geq \mu(t_1+1 \dots t_2) \geq \mu(j+1 \dots t_2)$ and $\mu(t_0 \dots t_1) \geq \mu(t_1+1 \dots t_2)$. Thus $\mu(t_0 \dots t_1) \geq \mu(j+1 \dots t_2)$ and $\mu(t_1+1 \dots j) \geq \mu(j+1 \dots t_2)$, so $\mu(t_0 \dots j) \geq \mu(j+1 \dots t_2)$ as desired. ∎

4.3.5 Parametrized clusters

Here we assume that the rate increases over the duration of the cluster according to some known, parametrized distribution. We focus here on the case where the rate is exponentially increasing (multiplied by ϕ on every time step). Similar expressions may be derived for the case of a linear increase in rate (i.e. rate is increased by Δ on every time step).

In this case, we compare the null hypothesis H_0: the rate equals 1 over all locations and times, to the set of alternative hypotheses $H_1(S)$: the rate is $\phi^{t-t_{min}+1}$ at times $t = t_{min} \dots T$ in region S, and equals 1 over all other locations and times. The likelihood ratio is:

$$F(S, t_{min}) = \frac{\max_{\phi \geq 1} \prod \Pr(c_i^t \sim \text{Poisson}(b_i^t \phi^{t-t_{min}+1}))}{\prod \Pr(c_i^t \sim \text{Poisson}(b_i^t))}$$

$$= \frac{\max_{\phi \geq 1} \prod e^{-\phi^{t-t_{min}+1} b_i^t} (\phi^{t-t_{min}+1})^{c_i^t}}{\prod e^{-b_i^t}}$$

$$= \frac{\max_{\phi \geq 1} \prod_{t=t_{min}}^{T} e^{-\phi^{t-t_{min}+1} B_t} (\phi^{t-t_{min}+1})^{C_t}}{\prod_{t=t_{min}}^{T} e^{-B_t}}$$

$$= \max_{\phi \geq 1} \prod_{t=t_{min}}^{T} e^{(1-\phi^{t-t_{min}+1})B_t} \phi^{(t-t_{min}+1)C_t}$$

where $B_t = \sum_{s_i^t \in S \times t} b_i^t$ and $C_t = \sum_{s_i^t \in S \times t} c_i^t$. Maximizing with respect to ϕ requires finding the root of a polynomial of degree $T - t_{min} + 1$; approximate (gradient) methods may also be used.

4.4 Inferring baseline values

In order to infer the baselines b_i^t for the "current" days $T - W < t \leq T$, we must consider two distinct questions: on what level to *aggregate* the data for time series analysis, and what method of time series analysis to use. We consider three different levels of spatial aggregation, which we term "building-aggregated time series" (BATS), "cell-aggregated time series" (CATS), and "region-aggregated time series" (RATS) respectively. For the BATS method, we consider the time series for each spatial location independently; for example, we may have a separate time series for each store or hospital, or counts may be already aggregated at some level (e.g. zip code). For each of these locations s_i, we independently compute the baselines b_i^t ($T - W < t \leq T$) from the past counts c_i^t ($1 \leq t \leq T - W$), using one of the time series analysis methods below. Then whenever we calculate $F(S, t_{min})$ for a region, we use the baselines b_i^t and counts c_i^t for each location in the region. The CATS method first computes the aggregate count c_i^t for each cell of the grid $s_i \in G$ on each day t, by summing counts of all spatial locations in that cell. Then the baselines b_i^t are computed independently for each grid cell $s_i \in G$, and whenever we calculate $F(S, t_{min})$ for a region, it is the *cell* counts and baselines that we use to compute the score. Finally, the RATS method, whenever it searches a region S, aggregates the time series of counts $C_t(S)$ "on the fly" by summing counts of all spatial locations in that region, computes baselines $B_t(S)$ for the "current" days $T - W < t \leq T$, and applies the score function $F(S, t_{min})$ to the resulting counts and baselines. This allows us to account for spatial correlations between cells, because the resulting "region" time series is formed by aggregating these correlated counts. However, the lack of a separate baseline per cell makes it more difficult to perform significance testing, as discussed below.

Randomization testing must also be performed differently for each of the three levels of aggregation. To generate a replica grid for BATS, we independently draw a count for each spatial location s_i for each current day t, using its baseline b_i^t. To generate a replica grid for CATS, we independently draw a count for each *cell* of the grid $s_i \in G$ for each current day t, using the cell baseline b_i^t. Finally, randomization testing for RATS is somewhat more difficult than for the other methods, since we must produce cell counts from a correlated distribution. This can be done by Gibbs sampling [55] or possibly generalized Monte Carlo significance testing [14], but the need to perform sampling makes randomization much more computationally expensive. Another alternative would be to bound False Discovery Rate [12] or some other criterion rather than computing statistical significance.

We also note that missing data is a potentially serious problem for all of these methods. For BATS, we may use time series approaches which adjust for the presence of missing data; for CATS and RATS, we must infer these missing values before aggregating data at the cell or region level. For the over-the-counter drug sales data, our current best approach is an exponentially weighted regression approach, applied to day-of-week adjusted counts; the adjustment is made by estimating the proportion of weekly counts falling on each day, and normalizing by these factors.

4.4.1 Time series analysis methods

For a given location, cell, or region s_i, our goal is to estimate the expected values of the "current" counts, $b_i^t = E[c_i^t]$, $T - W < t \leq T$, from the time series of "past" counts c_i^t, $1 \leq t \leq T - W$. A variety of univariate time series methods may be used to infer these baselines, depending on how we wish to deal with three questions: day of week effects, seasonal trends, and bias.

Many epidemiological quantities (for example, OTC drug sales) exhibit strong day of week and seasonal trends. Here we consider three methods of dealing with day of week effects: we can ignore them, *stratify* by day of week (i.e. perform a separate time series calculation for each day of the week), or *adjust* for day of week. To adjust for day of week, we assume that the observed count on a given day is the product of an "actual" count and a constant dependent on the day of week. Thus we compute the proportion of counts β_i on each day of the week ($i = 1 \ldots 7$). Then we transform each past day's observed count by dividing by $7\beta_i$, do a single time series calculation on the transformed past counts to predict the transformed current counts, and finally multiply by $7\beta_i$ to obtain the predicted count for each current day. By adjusting instead of stratifying, more data is used to predict each day's count (potentially reducing the variance of our estimates), but the success of this approach depends on the assumption of a constant and multiplicative day-of-week effect.

We also consider three methods of adjusting for seasonal trends: to use only the most recent counts (e.g. the past four weeks) for prediction, to use all counts but weight the most recent counts more (as is done in our exponentially weighted moving average and exponentially weighted linear regression methods), and to use regression techniques to extrapolate seasonal trends to the current data. For datasets with little or no seasonal trend, simple mean or moving average methods can be sufficient, but for datasets with strong seasonality, these methods will lag behind the seasonal trend, resulting in numerous false positives for increasing trends (e.g. sales of cough and cold medication at the start of winter) or false negatives for decreasing trends (e.g. cough and cold sales at the end of winter). To account for these trends, we recommend the use of regression methods (either weighted linear regression or non-linear regression depending on the data) to extrapolate the current counts. Finally, we consider both methods which attempt to give an unbiased estimate of the current count (e.g. mean of past counts), and methods which attempt to give a positively biased estimate of the current count (e.g. maximum of past counts). As we show, the unbiased methods typically have better detection power, but the conservatively biased methods have the advantage of reducing the number of false positives to a more manageable level.

Here we consider a total of 10 time series analysis methods, including "all_max" (b_i^t = maximum count of last 28 days), "all_mean" (b_i^t = mean count of last 28 days), "strat_max" (b_i^t = maximum count of same day of week, 1-4 weeks ago), "strat_mean" (b_i^t = mean count of same day of week, 1-4 weeks ago), two exponentially weighted moving average methods ("strat_EWMA" stratified by day of week, "adj_EWMA" adjusted for day of week), and two exponentially weighted linear regression methods ("strat_EWLR" stratified by day of week, "adj_EWLR" adjusted for day of week). Our final two methods are inspired by the recent work of Kulldorff et al. [85] on the "space-time permutation scan statistic," so we call them "strat_Kull" (stratified by day of week) and "all_Kull" (ignoring day of week effects). In this framework, the baseline b_i^t is computed as $\frac{\sum_t c_i^t \sum_i c_i^t}{\sum_t \sum_i c_i^t}$, i.e. space and time are assumed to be independent, so the expected fraction of all cases occurring in location s_i on day t can be computed as the product of the fraction of all cases occurring in location s_i and the fraction of all cases occurring on day t. The problem with this method is that the current day's counts are used for prediction of the current day's *expected* counts. As a result, if there is a cluster on the current day, the baselines for the current day will also be higher, reducing our power to detect the cluster. Nevertheless, the strat_Kull and all_Kull methods do extremely well when detecting localized clusters (where the increase in counts is noticeable for a small region, but the region is small enough that the total count for the day is essentially unaffected).

We also note an interesting interaction between the level of aggregation and the method of time series analysis. If the expected counts b_i^t ($T - W < t \leq T$) are calculated as a linear combination of past counts c_i^t ($1 \leq t \leq T - W$), and the weights for each past day t are constant from location to location, then we will calculate the same baselines (and thus, the same scores) regardless of whether we aggregate on the building, cell, or region level. This turns our to be true for most of the methods we investigate: all_mean, strat_mean, strat_EWMA, strat_EWLR, all_Kull, and strat_Kull. On the other hand, if we choose different weights for each location (as is the case when we adjust for day of week, as in adj_EWMA and adj_EWLR), we will calculate different baselines (and thus, different scores) depending on our level of aggregation. Finally, we have very different results for the "max" methods (strat_max and all_max) depending on the level of aggregation, because the maximum is not a linear operator. Since the sum of the maximum counts of each location ($\sum_{s_i \in S} \max_t c_i^t$) is higher than the maximum of the sum ($\max_t \sum_{s_i \in S} c_i^t$), we always expect BATS to predict the highest baselines, and RATS to predict the lowest baselines. For the results given below, we only distinguish between BATS, CATS, and RATS aggregation for those methods where the distinction is relevant (all_max, strat_max, adj_EWMA, and adj_EWLR).

4.5 Computational considerations

We begin by making two important observations. First, for any of the time series analysis methods given above, the baselines b_i^t ($T - W < t \leq T$) can be inferred from the past counts c_i^t ($1 \leq t \leq T - W$) in $O(T)$. Second, we can compute the score function $F(S, t_{min})$, for a given spatial region S and for all $T - W < t_{min} \leq T$, in total time $O(W)$, regardless of whether the persistent or emerging scan statistic is used. This is obvious for the persistent statistic since we can simply proceed backward in time, adding the cumulative count C_t and cumulative baseline B_t for each day t, and recomputing the score. (We can accumulate these counts and baselines in $O(W)$ by using the "cumulative counts" trick discussed in Chapter 3 for each of the W current days.) The $O(W)$ complexity is less obvious for the emerging statistic, since adding any new day t may result in up to $O(W)$ pops from the stack. But each day is *pushed* onto the stack at most once, and thus the total number of *pops* for the W days is at most W, giving total complexity $O(W)$, not $O(W^2)$.

For the BATS method, our computation may be divided into three steps: first, we compute baselines for each spatial location, requiring total time $O(N_s T)$, where N_s is the number of locations. Second, we aggregate "current" store baselines and counts to the grid, requiring time $O(N^2 W)$ where N is the grid size. Third, we search over all spatio-temporal regions (S, t_{min}): for each such region, we must compute the aggregate counts and baselines, and apply the score function F. As noted above, we can do this in $O(W)$ per region, but since a naïve search requires us to examine all $O(N^4)$ gridded rectangular regions, the total search time is $O(N^4 W)$, bringing the total complexity to $O(N_s T + N^4 W)$. For CATS, we first aggregate all store baselines and counts to the grid, requiring time $O(N_s T + N^2 T)$. Then we calculate baselines for each of the N^2 grid cells, requiring total time $O(N^2 T)$. Finally, we search over all spatio-temporal regions; as in BATS, this requires time $O(N^4 W)$, bringing the total complexity to $O(N_s T + N^2 T + N^4 W)$. For RATS, we first aggregate all store baselines and counts to the grid (as in CATS), requiring time $O(N_s T + N^2 T)$. Then for each of the N^4 regions we search, we must calculate the baselines for "current" days on the fly, requiring time $O(T)$, and compute the score function using the counts and baselines for current days, requiring time $O(W)$. Thus the total complexity is $O(N_s T + N^4 T)$.

For large grid sizes N, the $O(N^4)$ complexity of searching over all spatial regions makes a naïve search over all such regions computationally infeasible. However, we can apply the *fast spatial scan* discussed in Chapter 3, allowing us to find the highest scoring region and its p-value while searching only a small fraction of possible regions. In the purely spatial case, the fast spatial scan works by using a multi-resolution, branch-and-bound search to *prune* sets of regions that can be proven to have lower scores than the best region score found so far. We can easily extend this method to the space-time case: given a spatial region S, we must upper bound the scores $F(S', t_{min})$ for all regions $S' \subseteq S$ and $T - W < t_{min} \leq T$. The simplest way of doing so is to compute separate bounds on baselines and counts of S' for each time step t, using the methods given in Chapter 3, then use these bounds to compute an upper bound on the score. It might also be possible to achieve tighter bounds (and thus, better pruning) by enforcing *consistency* constraints across multiple days, i.e. ensuring that S' has the same spatial dimensions on each time step.

4.6 Related work

In general, spatio-temporal methods can be divided into three classes: spatial modeling techniques such as "disease mapping," where observed values are spatially smoothed to infer the distribution of values in space-time [28, 16]; tests for a general tendency of the data to cluster [77, 101]; and tests which attempt to infer the location of clusters [82, 81, 85]. We focus on the latter class of methods, since these are the only methods which allow us to both answer whether any significant clusters exist, and if so, identify these clusters. Three spatio-temporal cluster detection approaches have been proposed by Kulldorff et al.: the retrospective and prospective space-time scan statistics [82, 81], and the space-time permutation scan statistic [85]. The first two approaches attempt to detect persistent clusters, assuming that baselines are given based on census population estimates. The retrospective statistic searches over all space-time intervals, while the prospective statistic searches over those intervals ending at the present time. As noted above, these formulations make sense for the case of explicitly given denominator data, and counts *proportional* to these baselines (e.g. we expect a population of 10000 to have twice as many cases as a population of 5000, but do not know how many cases we expect to see). They are not appropriate for the case where we infer the *expected values* of counts from the time series of past counts (e.g. based on past data, we expect to see 40 cases in the first population and 15 cases in the second). Even if accurate denominator data is provided, the retrospective and prospective statistics may pick up purely spatial clusters resulting from spatial variation in the underlying rate (e.g. different parts of the country have different disease rates), or purely temporal clusters based on temporal fluctuations in rate (seasonal effects or long-term trends), and thus the detected clusters tend to be less useful for prospective detection of emerging outbreaks.

The recently proposed "space-time permutation scan statistic" [85] attempts to remedy these problems; like the present work, it allows baseline data to be inferred from the time series of past counts. As noted above, baselines are calculated by assuming that cases are independently distributed in space and time, and a variant of the test for persistent clusters is used (searching for regions with higher rate inside than outside). Then randomization testing is done by permuting the dates and locations of cases. This method focuses on detecting *space-time interaction*, and explicitly avoids detecting purely spatial or purely temporal clusters. The disadvantages of this are twofold. First, it loses power to detect spatially large clusters, because (as noted above) the current day's counts are used to estimate what the current day's counts should be. In the most extreme case, a spatially uniform multiplicative increase in disease rate over the entire search area would be completely ignored by this method, and thus it is unsafe to use for surveillance except in combination with other methods. The second disadvantage is that if the count decreases in one spatial region and remains constant elsewhere, this is detected as a spatio-temporal cluster. This results in false positives in cases where stores in one area are closed and stores in a different area remain open: the open stores are flagged as a cluster even if their counts have actually decreased.

All of the previously proposed space-time scan statistics are population-based: they often start from census data, which gives an unadjusted population p_i corresponding to each spatial location s_i. This population can then be adjusted for covariates such as the distribution of patient age and gender, giving an estimated "at-risk population" for each spatial location. In a recent paper, Kleinman et al. [75] suggest two additional, model-based adjustments to the population estimates. First, they present a method for temporal adjustment (accounting for day of week, month of the year, and holidays), making the populations larger on days when more visits are likely (e.g. Mondays during influenza season) and smaller on days when fewer visits are likely (e.g. Sundays and holidays). Second, they apply a "generalized linear mixed models" (GLMM) approach, first presented in Kleinman et al. [76], to adjust for the differing baseline risk in each census tract. This makes the adjusted population larger in tracts which have a larger baseline risk, which makes sense since a given number of observed cases should not be as significant if the observed counts in that region are consistently high. These baseline risks are computed from historical data, i.e. the time series of past counts in each census tract, using the GLMM version of logistic regression to fit the model.

Another possibility for inferring baselines is to make the assumption of independence of space and time, as in [85]; this means that the expected count in a given region is equal to the total count of the entire area under surveillance, multiplied by the historical proportion of counts in that region. This approach is successful in detecting very localized outbreaks, but loses power to detect more widespread outbreaks [120]. The reason for this is that a widespread outbreak will increase the total count significantly, thus increasing the expected count in the outbreak region, and hence making the observed increase in counts seem less significant. In the worst scenario, a massive outbreak which causes a constant, multiplicative increase in counts across the entire area under surveillance would be totally ignored by this approach; this is also true for many of the population-based methods, since they only detect spatial variation in disease rate, not an overall increase in counts. If these methods are used, we recommend using a purely temporal method in parallel to ensure that large-scale outbreaks (as well as localized outbreaks) can be detected. Either way, the accurate inference of expected counts from historical data is still an open problem, with different methods performing well for different datasets and outbreak types.

Several other spatio-temporal cluster detection methods have also been proposed. Iyengar [74] searches over "truncated rectangular pyramid" shapes in space-time, thus allowing detection of clusters which move and grow (or shrink) linearly over time; the disadvantage is that this much larger set of possible space-time regions can only be searched approximately. Assuncao et al. [9] assume a spatio-temporal Poisson point process: the exact location of each point in time and space is given, rather than aggregating points to discrete locations and intervals. A test statistic similar to the space-time permutation scan statistic is derived, assuming a Poisson intensity function that is separable in space and time.

4.7 Results

We evaluated our methods on two types of simulated outbreaks, injected into real Emergency Department and over-the-counter drug sale data for Allegheny County, Pennsylvania.[3] First, we considered aerosol releases of inhalational anthrax (e.g. from a bioterrorist attack), produced by the BARD ("Bayesian Aerosol Release Detector") simulator of Hogan et al. [70]. The BARD simulator takes in a "baseline dataset" consisting of one year's worth of Emergency Department records, and the quantity of anthrax released. It then produces multiple simulated attacks, each with a random attack location and environmental conditions (e.g. wind direction), and uses a Bayesian network model to determine the number of spores inhaled by members of the affected population, the resulting number and severity of anthrax cases, and the resulting number of respiratory Emergency Department cases on each day of the outbreak in each affected zip code. Each simulated outbreak can then be injected into the baseline ED dataset, and our methods' detection performance can be evaluated using the testing framework below.

Second, we considered a "Fictional Linear Onset Outbreak" (or "FLOO"), with a linear increase in cases over the duration of the outbreak. A FLOO outbreak is a simple simulated outbreak defined by a set of zip codes, a duration T_{floo}, and a value Δ. The FLOO simulator then produces an outbreak lasting T_{floo} days, with $t\Delta$ respiratory cases in each of the zip codes on day t, $0 < t \leq T_{floo}/2$, and $T_{floo}\Delta/2$ cases on day t, $T_{floo}/2 \leq t < T_{floo}$. Thus we have an outbreak where the number of cases ramps up linearly for some period of time, then levels off. While this is clearly a less realistic model than the BARD-simulated anthrax attack, it does have several advantages. It allows us to precisely control the parameters of the outbreak curve (number of cases on each day), allowing us to test the effects of these parameters on our methods' detection performance. Also, it allows us to perform experiments using over-the-counter drug sale data as well as Emergency Department data, while the BARD simulator only simulates ED cases.

We note that simulation of outbreaks is an active area of ongoing research in biosurveillance. The creation of realistic outbreak scenarios is important because of the difficulty of obtaining sufficient labeled data from real outbreaks, but is also very challenging. State-of-the-art outbreak simulations such as those of Buckeridge et al. [23], and Wallstrom et al. [154] combine disease trends observed from past outbreaks with information about the current background data into which the outbreak is being injected, as well as allowing the user to adjust parameters such as outbreak duration and severity.

We now discuss our basic semi-synthetic testing framework, followed by a discussion of the performance of our methods on each of the three main experiments (anthrax outbreaks in ED data, FLOO outbreaks in ED data, and FLOO outbreaks in OTC data).

4.7.1 Semi-synthetic testing

Our basic goal in the semi-synthetic testing framework is to evaluate detection performance: what proportion of outbreaks a method can detect, and how long it takes to detect these outbreaks. Clearly these numbers are dependent on how often the method is allowed to "sound the alarm," and thus we have a tradeoff between sensitivity (i.e. ability to detect true outbreaks) and detection time on the

[3] All data was aggregated to the zip code level to ensure anonymity, giving 88 distinct spatial locations (zip code centroids). The ED data contained an average of 40 respiratory cases/day, while the OTC data averaged 4000 sales of cough and cold medication/day.

Table 4.1: Summary of performance. Detection rate and average days to detect, 1 false positive/month, all datasets.

	best		median		temporal		spatial		
dataset	rate	days	rate	days	rate	days	rate	days	best method
BARD (0.125)	1.000	1.600	1.000	1.800	1.000	1.900	1.000	2.317	1-day, all_mean
BARD (0.015625)	0.883	3.679	0.883	3.906	0.867	4.250	0.883	5.094	1-day, all_mean
FLOO_ED (1,20)	1.000	4.484	1.000	5.066	0.988	6.119	1.000	7.289	3-day emerging, strat_EWMA
FLOO_ED (2,20)	1.000	2.898	1.000	3.211	1.000	4.551	1.000	4.074	3-day emerging, strat_EWMA
FLOO_ED (4,14)	1.000	1.748	1.000	2.076	1.000	3.103	1.000	2.290	1-day, all_mean
FLOO_OTC (20,20)	1.000	3.891	0.595	7.621	0.315	7.358	0.260	8.910	1-day, strat_Kull
FLOO_OTC (40,14)	1.000	2.319	0.981	4.609	0.240	4.667	0.232	6.082	1-day, strat_Kull
FLOO_OTC (all1,14)	0.475	5.424	0.179	3.340	0.274	5.000	0.213	6.036	1-day, strat_EWLR

one hand, and specificity (i.e. frequency of false positives) on the other. More precisely, our semi-synthetic framework consists of the following components. First, given one year of baseline data (assumed to contain no outbreaks), we run the space-time scan statistic for each day of the last nine months of the year (the first three months are used to provide baseline data only; no outbreaks in this time are considered). We thus obtain the highest scoring region S^*, and its score $F^* = F(S^*)$, for each of these days. Then for each "attack" that we wish to test, we do the following. First, we inject that outbreak into the data, incrementing the number of cases as above. Then for each day of the attack, we compute the highest scoring *relevant* region S^* and its score F^*, where a relevant region is defined as one which contains the centroid of all the cases injected that day. The reason that we only allow the algorithm to search over relevant regions is because we do not want to reward it for triggering an alarm and pinpointing a region which has nothing to do with the outbreak. We then compute, for each day $t = 0 \ldots T_{outbreak}$ (where $T_{outbreak}$ is the length of the attack), the fraction of baseline days (excluding the attacked interval) with scores higher than the maximum score of all relevant regions on days 0 to t. This is the proportion of false positives we would have to accept in order to have detected that outbreak by day t. By repeating this procedure on a number of outbreaks, we can obtain summary statistics about the detection performance of each method: we compute its averaged AMOC curve [47] (average proportion of false positives needed for detection on day t of an outbreak), and for a fixed level of false positives (e.g. 1 false positive/month), we compute the proportion of outbreaks detected and the average number of days to detection.

Note that this basic framework does not perform randomization testing, but only compares *scores* of attack and baseline days. There are several disadvantages to this method: first, since the baselines b_i^t for each day are different, the distribution of scores for each day's replica grids will be different, and thus the highest scoring regions may not correspond exactly to those with the lowest p-values. A second disadvantage is that it does not tell us how to perform *calibration*: setting threshold p-values in order to obtain a fixed false positive rate in real data. This is discussed in more detail below.

We tested a total of 150 methods: each combination of the three aggregation levels (BATS, CATS, RATS), five space-time scan statistics (1-day, 3-day emerging, 3-day persistent, 7-day emerging, 7-day persistent) and the ten methods of time series analysis listed above. We compared these methods against two simple "straw men": a purely spatial scan statistic (assuming uniform underlying at-risk population, and thus setting the baseline of a region proportional to its area), and a purely temporal scan statistic (analyzing the single time series formed by aggregating together all spatial locations, using 1-day all_mean). Since both the ED and OTC datasets were relatively small

Table 4.2: Comparison of methods. Average days to detect, 1 false positive/month, all datasets.

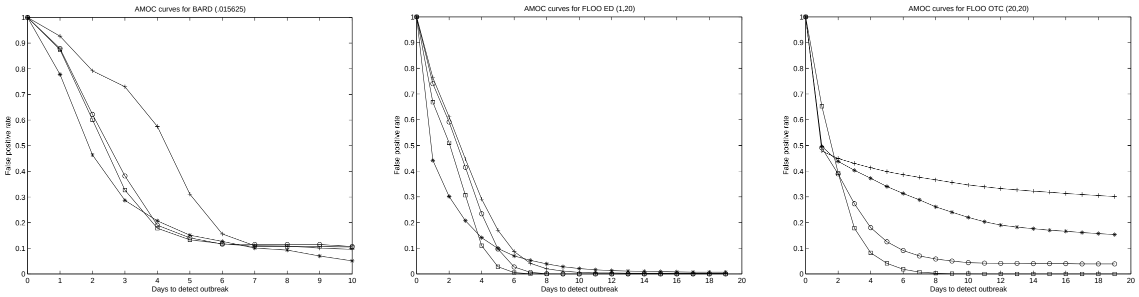
method	BARD (0.125)	BARD (0.016)	FLOO_ED (1,20)	FLOO_ED (2,20)	FLOO_ED (4,14)	FLOO_OTC (20,20)	FLOO_OTC (40,14)	FLOO_OTC (all1,14)
1-day	**1.60**	**4.53**	5.62	3.05	**1.75**	**3.89**	**2.32**	**9.92**
3-day persistent	1.75	4.58	4.53	2.93	1.94	4.02	2.61	11.61
3-day emerging	1.75	4.55	**4.48**	**2.90**	1.92	3.96	2.53	11.57
7-day persistent	1.80	4.67	4.73	3.06	2.01	4.35	2.83	11.89
7-day emerging	1.77	4.67	4.71	3.09	2.00	4.29	2.78	11.73
all_max_BATS	1.98	5.03	6.34	3.61	2.16	6.58	3.30	10.80
all_max_CATS	1.97	4.92	5.75	3.18	2.03	6.58	3.46	10.80
all_max_RATS	1.72	4.65	5.06	3.32	2.03	10.15	5.11	11.02
all_mean	**1.60**	**4.53**	4.79	3.04	**1.75**	15.34	6.67	11.78
strat_max_BATS	1.87	4.83	5.25	3.38	2.17	7.11	3.69	11.73
strat_max_CATS	1.87	4.82	5.25	3.23	2.10	7.21	3.75	11.82
strat_max_RATS	1.73	4.68	5.20	3.21	2.08	12.34	4.57	11.54
strat_mean	1.75	4.63	4.68	3.04	1.99	15.92	6.46	11.67
strat_EWMA	1.75	4.58	**4.48**	**2.90**	1.92	16.88	11.49	12.19
adj_EWMA	1.68	4.55	4.65	2.92	1.89	16.58	7.56	11.84
strat_EWLR	1.83	4.82	5.17	3.42	2.29	10.84	5.23	**9.92**
adj_EWLR	1.75	4.67	5.24	3.12	2.03	10.19	4.36	10.78
all_Kull	1.80	4.65	4.69	2.96	1.95	4.25	2.59	11.63
strat_Kull	1.75	4.68	4.53	2.92	1.94	**3.89**	**2.32**	10.89

in spatial extent (containing only records from Allegheny County), we used a small grid ($N = 16$, maximum cluster size = 8), and thus it was not necessary to use the fast spatial scan. For larger datasets, such as nationwide OTC data, a much larger grid size (e.g. $N = 256$) is necessary to achieve adequate spatial resolution, and thus the fast spatial scan will be an important component of our nationwide disease surveillance system.

For each outbreak type, we compared the detection performance of our methods to the two straw men, and also determined which of our methods was most successful (Table 4.1). Performance was evaluated based on detection rate (proportion of outbreaks detected) at 1 false positive/month, with ties broken based on average number of days to detect; we list both the performance of our "best" spatio-temporal method according to this criterion, as well as a representative "median" method (i.e. the 75th best method out of 150). We compare the methods in more detail in Table 4.2, giving each method's average number of days to detection at 1 false positive/month, assuming that undetected outbreaks were detected on day $T_{outbreak}$. For each of the five scan statistics, we report performance assuming its best combination of time series analysis method and aggregation level; for each of the ten time series analysis methods, we report performance assuming its best scan statistic. Level of aggregation only made a significant difference for the all_max and strat_max methods, so we report these results separately for BATS, CATS, and RATS. For each outbreak, we also construct AMOC curves of the "best," "median," purely temporal, and purely spatial methods; we present three of these curves (one for each outbreak type) in Figure 4.1. We also discuss each outbreak type in more detail below.

4.7.2 Anthrax outbreaks, ED data

For the anthrax outbreaks, we began with real baseline data for respiratory Emergency Department visits in Allegheny County in 2002. We used this data to simulate epidemics using BARD at two different levels of anthrax release: 0.125 (high) and 0.015625 (low). For each release amount, 60



BARD (0.015625) FLOO_ED (1,20) FLOO_OTC (20,20)

Figure 4.1: AMOC curves for three of the eight datasets. The four curves are for the best spatio-temporal method (□), the median spatio-temporal method (○), the purely temporal method ($*$), and the purely spatial method ($+$). Note that the purely temporal method, unlike the others, is not required to pinpoint the region location, so its AMOC will be lower at the start of an attack (before there are a sufficient number of cases to detect); this is purely a function of the testing methodology, and does not imply better performance.

simulated epidemics were created. Separately for the high and low levels, we tested all methods, forming an average AMOC curve for each over all simulated epidemics, and measuring detection rate and average days to detect.

For the high release dataset, all of the methods tested were able to rapidly detect all 60 outbreaks. For a fixed false positive rate of 1/month, every method detected all outbreaks (100% detection rate), with average time to detection ranging from 1.6 to 2.067 days. The top method (1.6 days to detect) was the 1-day statistic using all_mean, and half of all methods detected in 1.8 days or fewer. Since the average delay from release to the first reported case was 1.18 days, these times were close to ideal detection performance. All methods except all_max outperformed the purely temporal scan statistic (100% detection rate, 1.9 days to detect), and all methods outperformed the purely spatial scan statistic (100% detection rate, 2.317 days to detect). For this dataset, there was very little difference between the best and worst performing methods, and thus it is hard to draw definitive conclusions. Nevertheless, we observed that shorter temporal windows performed better (1-day was best, 7-day was worst), and there were no significant differences between emerging and persistent scan statistics. Looking at the outbreak curve for this epidemic, it is clear why this is the case: all outbreaks have huge spikes in the number of cases starting on day 1 or 2, so there is no advantage to having a longer window; and since there is essentially no "ramp-up" in the number of cases (just the large spike, at which point the outbreak is obvious to any method) there is no advantage to the emerging over persistent statistics. For time series analysis, the all_mean method performed best, followed by adj_EWMA. This result is somewhat surprising, suggesting that the ED baseline data has very little day of week or seasonal trends.

Results on the low release dataset were similar, except for two differences resulting from the amount of release. First, 7 of the 60 outbreaks were missed by all methods; these outbreaks consisted of a very small number of cases (less than 5 in total), and as a result there was essentially no signal to detect. The other 53 outbreaks typically produced a large and obvious spike in cases (again, with very little ramp-up prior to the spike), though the delay between release and spike was longer on average (2.6 days from release to first reported case). Again, the 1-day window was best, though the 3-day statistics performed almost as well, and all_mean and adj_EWMA were the top two methods. Our spatio-temporal methods again outperformed the straw men, requiring 3.679 days to detect (best) and 3.906 days to detect (median) at 1 false positive/month. This was substantially better than the purely temporal and purely spatial methods, which required 4.250 and 5.094 days respectively.

4.7.3 FLOO outbreaks, ED data

For the FLOO_ED outbreaks, we again began with the 2002 Allegheny County ED dataset. We injected three types of FLOO attacks, assuming that only zip code 15213 (Pittsburgh) was affected: $(\Delta = 4, T_{floo} = 14)$, $(\Delta = 2, T_{floo} = 20)$, and $(\Delta = 1, T_{floo} = 20)$. Thus the first attack has the fastest-growing outbreak curve ($4t$ cases on day t), and the third has the slowest-growing outbreak curve (t cases on day t). For each outbreak type, we simulated outbreaks for all possible start dates in April-December 2002, and computed each method's average performance over all such outbreaks. All the spatio-temporal methods were able to detect all injected outbreaks at a rate of 1 false positive/month; not surprisingly, median number of days to detect increased from 2.076 for the fastest growing outbreak, to 5.066 for the slowest growing outbreak. All of these detection times were more than one full day faster than the purely spatial and purely temporal methods, with one exception (0.22 days faster than purely spatial for $\Delta = 4$). Again, the all_mean method performed well (1-day all_mean was the winner for $\Delta = 4$, with a detection time of 1.748 days), as did adj_EWMA and strat_EWMA (3-day emerging strat_EWMA was the winner for $\Delta = 2$ and $\Delta = 1$, with detection times of 2.898 and 4.484 days respectively). Our most interesting result was the effect of the temporal window size W: for the fastest growing outbreak, the 1-day method detected outbreaks 0.2 days faster than the 3-day and 7-day methods, but for the slowest growing outbreak, both 3-day and 7-day methods detected outbreaks a full day faster than the 1-day method. Emerging methods outperformed persistent methods for approximately 80% of our trials, though the difference in detection time was typically fairly small (0.02-0.10 days, depending on the time series analysis method). We also observed that higher aggregation typically performed better for the all_max and strat_max methods (i.e. RATS performed best, and BATS worst).

4.7.4 FLOO outbreaks, OTC data

For the FLOO_OTC outbreaks, we began with one year's worth of data for retail sales of over-the-counter cough and cold medication in Allegheny County, collected from 2/13/04-2/12/05. We injected three types of FLOO attacks: for the first two, we again assumed that only zip code 15213 was affected, but (since the overall numbers of OTC sales were much higher than the overall numbers of ED visits) we injected larger numbers of counts, $(\Delta = 40, T_{floo} = 14)$ and $(\Delta = 20, T_{floo} = 20)$. For the third attack, we assumed that *all* zip codes in Allegheny County were affected, using $(\Delta = 1, T_{floo} = 14)$ for each. For each outbreak type, we simulated outbreaks for all possible start dates over the last nine months of our data, and computed each method's average performance over all such outbreaks. Our first observation was that these attacks were substantially harder to detect than in the ED data: for the two localized attacks, our median methods only detected 98.1% and 59.5% of outbreaks for the faster-growing ($\Delta = 40$) and slower-growing ($\Delta = 20$) outbreaks respectively. It appears that the main reason for this was the difficulty in accurately predicting the OTC counts for the baseline days, as we observed huge differences in performance between the various time series analysis methods. The data contained significant seasonal and day of week trends, as well as other irregularities (e.g. large spikes in sales in single stores, probably resulting from promotions), and most of our methods were not entirely successful in accounting for these; nevertheless, they performed much better than the purely spatial and purely temporal methods, which only detected 23-32% of these outbreaks. Our second observation was that the strat_Kull method performed remarkably well in predicting the localized outbreaks, detecting with 100% accuracy in 2.32 and 3.89 days for $\Delta = 40$ and $\Delta = 20$ respectively; strat_Kull and all_Kull detected the $\Delta = 20$ outbreaks over two days faster than any other methods. This suggests that those methods were able to predict baselines for the non-attack days much more accurately than any of the other time series analysis methods: using the current day's counts to predict the current day's baselines allows accurate adjustment for seasonal trends, and *if the attack is sufficiently localized*, only slightly reduces detection power. Clearly it would be better to have a method which correctly predicts the trends *without* using the current day's counts, but none of the methods discussed here were able to do this. For the non-localized attack (cases added to every zip code), the power of strat_Kull was substantially reduced, and it was only able to detect 36% of outbreaks, while our best-performing method (strat_EWLR) detected 48%. And this is far from the worst case for strat_Kull: since different zip codes have different average sales, adding the same number of counts to each creates a large amount of space-time interaction. If we had instead *multiplied* counts in each zip code by the same factor, strat_Kull would have *no* power to detect this. We also note that the 1-day statistics performed best for all three outbreak types on the OTC data, though the 3-day emerging statistics performed almost as well. Again, emerging methods consistently outperformed persistent methods, and the difference in detection time was larger than on the ED data (typically 0.05-0.20 days). Finally, we note that the lower levels of aggregation (BATS and CATS) outperformed RATS for the "max" methods; this is the opposite result from what we observed on the ED data.

Based on these conflicting results, it is difficult to recommend a single method for use on all datasets and outbreak types. As shown above, the optimal temporal window size depends on how fast the number of cases increases, with longer temporal windows appropriate for more slowly growing outbreaks. The optimal temporal window is also affected by our desired tradeoff between number of false positives and detection time: a lower acceptable false positive rate (and thus, longer acceptable detection time) increases the optimal window size. For example, for the FLOO_ED (1,20) outbreak, the 3-day emerging statistic has the fastest time to detection at a rate of 1 false positive/month, while the 7-day emerging statistic has the fastest time to detection at a rate of 1 false positive/year. As noted above, the emerging statistics consistently outperform the corresponding persistent statistics, and while the amount of difference is not that large (0.02-0.20 days across all outbreaks and methods), even slightly earlier detection may make a substantial difference in the efficacy of outbreak response. It appears that the 3-day emerging statistic is a reasonable compromise solution, at least for the set of outbreaks tested. It may also be a good idea to run emerging statistics with different window sizes in parallel, for better detection of both fast-growing and slow-growing outbreaks; optimal combination of detectors is an interesting and open research question. It is clear that the best time series analysis method depends on the characteristics of the dataset, as well as whether the outbreak is spatially localized or occupies a large spatial region: the strat_Kull method is excellent for localized outbreaks, but should be used only in parallel with another method that can detect large-scale outbreaks. For datasets with little seasonal trend, such as the ED data used here, very simple mean and moving average methods are sufficient, but it is still an open question to find a method which can accurately predict baseline counts for OTC data without using the current day's counts to predict the current day's expectations.

Table 4.3: Comparison of expectation-based and population-based scan statistics. Days to detect and proportion of outbreaks detected, 1 false positive/month.

	FLOO_ED	FLOO_ED	FLOO_ED	BARD_ED	BARD_ED	FLOO_OTC	FLOO_OTC
method	(4,14)	(2,20)	(1,20)	(.125)	(.016)	(40,14)	(25,20)
population-based	1.859 (100%)	3.324 (100%)	6.122 (96%)	1.733 (100%)	3.925 (88%)	**3.582** **(100%)**	**5.393** **(100%)**
expectation-based	**1.729** **(100%)**	**3.035** **(100%)**	**5.545** **(99.6%)**	**1.600** **(100%)**	**3.679** **(88%)**	5.679 (61.6%)	7.513 (44.0%)

4.7.5 Comparison of expectation-based and population-based approaches

We have shown that the expectation-based space-time scan statistic is able to rapidly and accurately detect disease outbreaks, and that this approach outperforms both purely temporal and purely spatial scan statistics. We now compare the expectation-based and population-based scan statistics on the ED and OTC datasets, using the same method of estimating baselines for each (all_mean, CATS, no time series correction) and a 1-day temporal window. Based on our preliminary results (running both methods on synthetic, purely spatial data), we expect that for a given, unbiased estimate of the expected count, the expectation-based statistic will outperform the population-based statistic. On the other hand, the population-based statistic will be more robust to a consistent global bias in estimation (overestimating or underestimating the total count for each day). To see which method performs better on the ED and OTC datasets, we compare the two methods for seven experiments, as shown in Table 4.3. From these results, we can see that the expectation-based statistic outperforms the population-based statistic on all five runs for the ED dataset, by an average of 0.369 days (approximately nine hours). On the other hand, the population-based statistic outperforms the expectation-based statistic by a large margin on the OTC dataset, detecting almost twice as many outbreaks and two days faster. These results demonstrate that the expectation-based statistic does well when we have accurate estimates of the expected counts, but poorly when the estimates are not accurate. As we know from the above discussion, the all_mean method does not account well for seasonal trends, resulting in poor estimates of the expected counts for OTC data. The population-based method is more robust to estimation errors than the expectation-based method, but even better performance can be achieved by using the expectation-based approach with time series analysis methods that account for seasonal trends, or by using the Bayesian cluster detection methods of Chapter 5.

4.7.6 Effects of time series correction

As noted above, an exponentially weighted linear regression method is typically used to correct the time series data before applying our space-time scan statistics. One factor that complicates the use of this method, however, is that our input data streams (ED and OTC data) do not indicate whether a given location's data is missing, but simply return a zero count. Thus for the ED data, we do not perform time series correction since we are unable to tell zero values from missing values. For the OTC data, on the other hand, average counts are much larger, especially for cough and cold sales. Thus we use a simple heuristic: if sales of all types are zero for a given store on a given day, we assume that the data is missing and perform time series correction. As we show in Table 4.4, time series correction makes a large difference for both population-based and expectation-based

Table 4.4: Comparison of corrected and uncorrected time series methods. Days to detect and proportion of outbreaks detected, 1 false positive/month.

method	FLOO_OTC (40,14)	FLOO_OTC (25,20)
population-based (corrected)	**2.745** **(100%)**	**3.977** **(100%)**
population-based (uncorrected)	3.582 (100%)	5.393 (100%)
expectation-based (corrected)	5.627 (86.7%)	7.797 (55.6%)
expectation-based (uncorrected)	5.679 (61.6%)	7.513 (44.0%)

methods: the expectation-based method improves from 53% to 71% of outbreaks detected, and the population-based method can detect outbreaks over 1 day faster when time series correction is performed.

4.7.7 Calibration

As noted above, our testing framework simply compares scores of the highest scoring regions on each day, and computes AMOC curves; no randomization testing is done, and thus we do not actually compute the p-value of discovered regions. Because our detection performance is high, it is clear that the attacked regions would have lower p-values than the highest scoring regions on non-attacked days. But this does not answer the question of calibration: at what threshold p-value should we trigger an alarm? If non-attacked days were actually generated under the null hypothesis, we could choose some level α and be guaranteed that we will only trigger false alarms that proportion of the time (e.g. once every 20 days for $\alpha = .05$). However, our null hypothesis, that each count c_i^t is generated by a Poisson distribution with mean b_i^t, is clearly false, since b_i^t is only an estimate of what we expect c_i^t to be, assuming that no outbreak is present. If this estimate were unbiased and exactly precise (zero variance), then we would achieve a false positive rate of α. In practice, however, this estimate can be both biased and highly imprecise. For any method of calculating baselines that is approximately unbiased, but has non-zero variance (i.e. all of our time series analysis methods except all_max and strat_max), we expect the proportion of false positives to be greater than α, since the scan statistic picks out any regions where b_i^t is an underestimate of c_i^t. The all_max and strat_max methods, on the other hand, are conservatively biased (predicting values of b_i^t which overestimate c_i^t on average) but also have non-zero variances; thus they may result in proportions of false positives either higher or lower than α. To examine the calibration of our methods, we calculated the p-value for each day in both the ED and OTC datasets (with no injected attacks). We used a 3-day emerging scan statistic, BATS aggregation, with four different time series analysis methods: two unbiased methods (adj_EWLR and all_mean) and two conservative methods (all_max and strat_max). $R = 100$ randomizations were performed, and we counted the proportion of false positives at $\alpha = 0.01$ and $\alpha = 0.05$ for each method on each dataset. See Table 4.5 for results.

As expected, we observe a large number of false positives in both datasets for the unbiased methods. For the OTC dataset, we also have high false positive rates even for the conservative methods. What conclusions can we draw from this? Because of the variance in our predictions,

Table 4.5: Proportion of false positives

	ED dataset		OTC dataset	
method	$\alpha = .01$	$\alpha = .05$	$\alpha = .01$	$\alpha = .05$
adj_EWLR	0.171	0.393	0.725	0.808
all_mean	0.091	0.240	0.789	0.840
strat_max	0.000	0.025	0.275	0.344
all_max	0.000	0.000	0.058	0.072

the baseline data, especially the OTC data, is not fit well by the null hypothesis. Nevertheless, the likelihood ratio statistic (which serves as a sort of distance away from the null hypothesis) is very successful at distinguishing between attacks and non-attacked days. So how can we calibrate the statistic? One option would be to use an unbiased method with a much lower threshold α, but the problem with this is that it would require a huge number of randomizations to determine whether the p-value is less than α. Another option would be to use a conservative method, but the problem is that these methods not only record fewer false positives, but also are less able to detect a true positive. In fact, as our results above demonstrate, the conservative methods typically have much less power to distinguish attacks from non-attacked days for a given level of false positives, so this is clearly not a good idea. A better option is to trigger alarms for a given threshold on the *score* rather than on the p-value, with that threshold learned from previous data (e.g. the year of ED and OTC data used here). An even better solution might be to account for the uncertainty of our baseline estimates b_i^t, as discussed below, and thus make our null hypothesis more accurately describe the real data.

4.8 Conclusions

We have presented a new class of space-time scan statistics designed for the rapid detection of emerging clusters, and demonstrated that these methods are highly successful on the task of rapidly and accurately detecting emerging disease epidemics. We are currently working to extend this framework in a number of ways. Perhaps the most important of these extensions is to account for the imprecision in our baseline estimates b_i^t, using methods of time series analysis which not only predict the expected values of the "current" counts but also estimate the variance in these estimates. Our current difficulty is that we are testing the null hypothesis that all counts c_i^t are generated from the estimated values b_i^t, but since these values are only estimates, the null hypothesis is clearly false. As a result, as we demonstrated in the previous section, the standard randomization testing framework results in large numbers of false positives, i.e. on most non-attack days we still observe a p-value less than 0.05. The combination of time series methods which account for imprecision of estimates, and scan statistics which use distributions that can account for mean and variance separately (e.g. Gaussian or negative binomial distributions) should allow us to correct these problems. This will also make the distinction between building-aggregated, cell-aggregated, and region-aggregated time series methods more relevant, as the variance computations will be very different depending on the level of aggregation. A second (and related) extension is accounting for factors such as overdispersion and spatial correlation between neighboring counts. Our current methods assume that each spatial location, cell, or region has an independent time series of counts, and thus infer baselines independently for each such time series. When we extend the model to distributions that model mean and variance separately, we should be able to calculate correlations between time series of neighboring spatial locations, and adjust for these correlations.

Finally, we are in the process of applying our spatio-temporal scan statistics to nationwide over-the-counter drug sales, searching for emerging disease outbreaks on a daily basis. Scaling up the system to national data creates both computational issues (the use of the fast spatial scan is essential for searching large grids) as well as statistical issues (dealing with irregularities in the data, such as missing data, and increased sales resulting from product promotions). We are currently working with state and local public health officials to ensure that the clusters we report correspond to relevant potential outbreaks, thus rapidly and accurately identifying emerging outbreaks while keeping the number of false positives low.

Chapter 5

Bayesian spatial cluster detection

5.1 Introduction

Spatial cluster detection has two main goals: to identify the locations, shapes, and sizes of potential clusters, and to determine whether each potential cluster is more likely to be a "true" cluster or simply a chance occurrence. Thus we must compare the null hypothesis H_0 of no clusters against some set of alternative hypotheses $H_1(S)$, each representing a cluster in some region or regions S. In the standard frequentist setting, we compare these hypotheses by significance testing, computing the p-values of potential clusters by randomization; here we propose a Bayesian framework, in which we compute posterior probabilities of each potential cluster. Our primary motivating application is *prospective disease surveillance*: detecting spatial clusters of disease cases resulting from a disease outbreak. We perform surveillance on a daily basis, with the goal of finding emerging epidemics as quickly as possible, while keeping the number of false positives low.

In this chapter, I present a new Bayesian approach to spatial cluster detection, the "Bayesian spatial scan statistic," and demonstrate that this method has several advantages over the standard (frequentist) method. First, the Bayesian method allows us to incorporate prior information about the size and shape of an cluster, and the impact of the cluster on the data stream being monitored. Second, because randomization testing is unnecessary within the Bayesian framework, we can compute the Bayesian scan approximately 1000x faster than the frequentist approach. Other advantages of the Bayesian method include higher detection power and easier calibration, visualization, and interpretation of results. Additionally, the method can be extended to a "multivariate Bayesian scan statistic," enabling us to combine inputs from multiple data streams and to differentiate between different types of clusters (e.g. different types of outbreak in the disease surveillance case).

In Section 5.2, I review the frequentist spatial scan statistic and discuss some of its limitations, and in Section 5.3, I present the new Bayesian spatial scan statistic. Sections 5.4 and 5.5 compare the frequentist and Bayesian approaches with respect to detection power and computation time, and Section 5.6 details some other advantages of the Bayesian approach. Finally, in Section 5.7, I discuss extension of the Bayesian method to the multivariate case, and some of the potential advantages of the multivariate framework.

Much of this chapter has been adapted from our papers in NIPS 2005 [116] and the 2005 National Syndromic Surveillance Conference [117]. I wish to thank my co-authors Andrew Moore and Gregory Cooper for their contributions to this work. Thanks also to Andrew Lawson, Mike Wagner, and Artur Dubrawski for helpful feedback on the univariate and multivariate Bayesian approaches.

Finally, I wish to thank Gauri Datta and David Banks for suggesting the unbiased (UBayes) approach to prior selection, described in Section 5.3.

5.2 Review of the frequentist scan statistic

In the spatial surveillance setting, each day we have data collected for a set of discrete spatial locations s_i. For each location s_i, we have a *count* c_i (e.g. number of disease cases), and an underlying *baseline* b_i. The baseline may correspond to the underlying *population* at risk, or may be an estimate of the expected value of the count (e.g. derived from the time series of previous count data). Our goal, then, is to find if there is any spatial region S (set of locations s_i) for which the counts are significantly higher than expected, given the baselines. For simplicity, we assume here that the locations s_i are aggregated to a uniform, two-dimensional, $N \times N$ grid G, and we search over the set of rectangular regions $S \subseteq G$. This allows us to search both compact and elongated regions, allowing detection of elongated disease clusters resulting from dispersal of pathogens by wind or water.

One of the most important statistical tools for cluster detection is Kulldorff's *spatial scan statistic* [88, 78]. This method, described in detail in Chapters 1 and 2, searches over a given set of spatial regions, finding those regions which maximize a likelihood ratio statistic and thus are most likely to be generated under the alternative hypothesis of clustering rather than the null hypothesis of no clustering. Randomization testing is used to compute the p-value of each detected region, correctly adjusting for multiple hypothesis testing, and thus we can both identify potential clusters and determine whether they are significant. Kulldorff's framework assumes that counts c_i are Poisson distributed with $c_i \sim \text{Poisson}(qb_i)$, where b_i represents the (known) census population of cell s_i and q is the (unknown) underlying disease rate. Then the goal of the scan statistic is to find regions where the disease rate is higher inside the region than outside. The statistic used for this is the likelihood ratio $F(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$, where the null hypothesis H_0 assumes a uniform disease rate $q = q_{all}$. Under $H_1(S)$, we assume that $q = q_{in}$ for all $s_i \in S$, and $q = q_{out}$ for all $s_i \in G - S$, for some constants $q_{in} > q_{out}$.

Once we have found the highest scoring region $S^* = \arg\max_S F(S)$ of grid G, and its score $F^* = F(S^*)$, we must still determine the statistical significance of this region by randomization testing. To do so, we randomly create a large number R of replica grids by sampling under the null hypothesis, and find the highest scoring region and its score for each replica grid. Then the p-value of S^* is $\frac{R_{beat}+1}{R+1}$, where R_{beat} is the number of replicas G' with F^* higher than the original grid.

The frequentist scan statistic is a useful tool for cluster detection, and is commonly used in the public health community for detection of disease outbreaks. However, there are three main disadvantages to this approach. First, it is difficult to make use of any prior information that we may have, for example, our prior beliefs about the size of a potential outbreak and its impact on disease rate. Second, the accuracy of this technique is highly dependent on the correctness of our maximum likelihood parameter estimates. As a result, the model is prone to parameter overfitting, and may lose detection power in practice because of model misspecification. Finally, the frequentist scan statistic is very time consuming, and may be computationally infeasible for large datasets. A naïve approach requires searching over all rectangular regions, both for the original grid and for each replica grid. Since there are $O(N^4)$ rectangles to search for an $N \times N$ grid, the total computation time is $O(RN^4)$, where $R = 999$ is a typical number of replications. In Chapter 3, we show

how to reduce computation time by a factor of 20-2000x using the “fast spatial scan” algorithm; nevertheless, we must still perform this faster search both for the original grid and for each replica.

We propose to remedy these problems through the use of a Bayesian spatial scan statistic. First, our Bayesian model makes use of prior information about the likelihood, size, and impact of an outbreak. If these priors are chosen well, we should achieve better detection power than the frequentist approach. Second, the Bayesian method uses a *marginal likelihood* approach, averaging over possible values of the model parameters q_{in}, q_{out}, and q_{all}, rather than relying on maximum likelihood estimates of these parameters. This makes the model more flexible and less prone to overfitting, and reduces the potential impact of model misspecification. Finally, under the Bayesian model there is no need for randomization testing, and (since we need only to search the original grid) even a naïve search can be performed relatively quickly. We now present the Bayesian spatial scan statistic, and then compare it to the frequentist approach on the task of detecting simulated disease epidemics.

5.3 The Bayesian scan statistic

Here we consider the natural Bayesian extension of Kulldorff’s scan statistic, moving from a Poisson to a conjugate Gamma-Poisson model. Bayesian Gamma-Poisson models are a common representation for count data in epidemiology, and have been used in disease mapping by Clayton and Kaldor [28], Mollié [105], and others. In disease mapping, the effect of the Gamma prior is to produce a spatially smoothed map of disease rates; here we instead focus on computing the posterior probabilities, allowing us to determine the likelihood that an outbreak has occurred, and to estimate the location and size of potential outbreaks.

For the Bayesian spatial scan, as in the frequentist approach, we wish to compare the null hypothesis H_0 of no clusters to the set of alternative hypotheses $H_1(S)$, each representing a cluster in some region S. We assume that the hypotheses are mutually exclusive: $\text{Pr}(H_0) + \sum_S \text{Pr}(H_1(S)) = 1$, where the sum is taken over a given set of regions S. As before, we assume Poisson likelihoods, $c_i \sim \text{Poisson}(qb_i)$. The difference is that we assume a hierarchical Bayesian model where the disease rates q_{in}, q_{out}, and q_{all} are themselves drawn from Gamma distributions. Thus, under the null hypothesis H_0, we have $q = q_{all}$ for all $s_i \in G$, where $q_{all} \sim \text{Gamma}(\alpha_{all}, \beta_{all})$. Under the alternative hypothesis $H_1(S)$, we have $q = q_{in}$ for all $s_i \in S$ and $q = q_{out}$ for all $s_i \in G - S$, where we independently draw $q_{in} \sim \text{Gamma}(\alpha_{in}, \beta_{in})$ and $q_{out} \sim \text{Gamma}(\alpha_{out}, \beta_{out})$. We discuss how the α and β priors are chosen below.

From this model, we can compute the posterior probabilities $\text{Pr}(H_1(S) \mid D)$ of an outbreak in each region S, and the probability $\text{Pr}(H_0 \mid D)$ that no outbreak has occurred, given dataset D: $\text{Pr}(H_0 \mid D) = \frac{\text{Pr}(D \mid H_0)\text{Pr}(H_0)}{\text{Pr}(D)}$ and $\text{Pr}(H_1(S) \mid D) = \frac{\text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))}{\text{Pr}(D)}$, where $\text{Pr}(D) = \text{Pr}(D \mid H_0)\text{Pr}(H_0) + \sum_S \text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))$. We discuss the choice of prior probabilities $\text{Pr}(H_0)$ and $\text{Pr}(H_1(S))$ below. To compute the marginal likelihood of the data given each hypothesis, we must integrate over all possible values of the parameters (q_{in}, q_{out}, q_{all}) weighted by their respective probabilities. Since we have chosen a conjugate prior, we can easily obtain a closed-form solution:

$$\text{Pr}(D \mid H_0) = \int \text{Pr}(q_{all} \sim \text{Gamma}(\alpha_{all}, \beta_{all})) \prod_{s_i \in G} \text{Pr}(c_i \sim \text{Poisson}(q_{all} b_i))\, dq_{all}$$

$$\text{Pr}(D \mid H_1(S)) = \int \text{Pr}(q_{in} \sim \text{Gamma}(\alpha_{in}, \beta_{in})) \prod_{s_i \in S} \text{Pr}(c_i \sim \text{Poisson}(q_{in} b_i))\, dq_{in}$$
$$\times \int \text{Pr}(q_{out} \sim \text{Gamma}(\alpha_{out}, \beta_{out})) \prod_{s_i \in G-S} \text{Pr}(c_i \sim \text{Poisson}(q_{out} b_i))\, dq_{out}$$

Since we have Poisson-distributed counts and a Gamma prior, the marginal likelihood is negative binomial. Computing the integral, and letting $C = \sum c_i$ and $B = \sum b_i$, we obtain:

$$\int \text{Pr}(q \sim \text{Gamma}(\alpha, \beta)) \prod_{s_i} \text{Pr}(c_i \sim \text{Poisson}(q b_i))\, dq = \int \frac{\beta^\alpha}{\Gamma(\alpha)} q^{\alpha-1} e^{-\beta q} \prod_{s_i} \frac{(q b_i)^{c_i} e^{-q b_i}}{(c_i)!}\, dq$$

$$\propto \frac{\beta^\alpha}{\Gamma(\alpha)} \int q^{\alpha-1} e^{-\beta q} q^{\sum c_i} e^{-q \sum b_i}\, dq = \frac{\beta^\alpha}{\Gamma(\alpha)} \int q^{\alpha+C-1} e^{-(\beta+B)q}\, dq = \frac{\beta^\alpha\, \Gamma(\alpha + C)}{(\beta + B)^{\alpha+C}\, \Gamma(\alpha)}$$

Thus we have the following expressions for the marginal likelihoods:

$$\text{Pr}(D \mid H_0) \propto \frac{(\beta_{all})^{\alpha_{all}}\, \Gamma(\alpha_{all} + C_{all})}{(\beta_{all} + B_{all})^{\alpha_{all} + C_{all}}\, \Gamma(\alpha_{all})}$$

$$\text{Pr}(D \mid H_1(S)) \propto \frac{(\beta_{in})^{\alpha_{in}}\, \Gamma(\alpha_{in} + C_{in})}{(\beta_{in} + B_{in})^{\alpha_{in} + C_{in}}\, \Gamma(\alpha_{in})} \times \frac{(\beta_{out})^{\alpha_{out}}\, \Gamma(\alpha_{out} + C_{out})}{(\beta_{out} + B_{out})^{\alpha_{out} + C_{out}}\, \Gamma(\alpha_{out})}$$

The Bayesian spatial scan statistic can be computed simply by first calculating the score $F(S) = \text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))$ for each spatial region S, maintaining a list of regions ordered by score. We then calculate $\text{Pr}(D \mid H_0)\text{Pr}(H_0)$, and add this to the sum of all region scores, obtaining the probability of the data $\text{Pr}(D)$. Finally, we can compute the posterior probability $\text{Pr}(H_1(S) \mid D) = \frac{\text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))}{\text{Pr}(D)}$ for each region, as well as $\text{Pr}(H_0 \mid D) = \frac{\text{Pr}(D \mid H_0)\text{Pr}(H_0)}{\text{Pr}(D)}$. Then we can return all regions with non-negligible posterior probabilities and the posterior probability of each. We can also compute the overall probability of an outbreak, $\text{Pr}(H_1 \mid D) = \sum_S \text{Pr}(H_1(S) \mid D) = 1 - \text{Pr}(H_0 \mid D)$. Note that no randomization testing is necessary, and thus overall complexity is proportional to number of regions searched, e.g. $O(N^4)$ for searching over axis-aligned rectangles in an $N \times N$ grid.

5.3.1 Choosing priors

One of the most challenging tasks in any Bayesian analysis is the choice of priors. For any region S that we examine, we must have values of the parameter priors $\alpha_{in}(S)$, $\beta_{in}(S)$, $\alpha_{out}(S)$, and $\beta_{out}(S)$, as well as the region prior probability $\text{Pr}(H_1(S))$. We must also choose the global parameter priors α_{all} and β_{all}, as well as the "no outbreak" prior $\text{Pr}(H_0)$.

Here we consider the simple case of a uniform region prior, with a known prior probability of an outbreak P_1. In other words, if there is an outbreak, it is assumed to be equally likely to occur in any spatial region. Thus we have $\text{Pr}(H_0) = 1 - P_1$, and $\text{Pr}(H_1(S)) = \frac{P_1}{N_{reg}}$, where N_{reg} is the total number of regions searched. The parameter P_1 can be obtained from historical data or estimated by human experts. The model can also be easily adapted to a non-uniform region prior, taking into account our prior beliefs about the size, shape, and location of outbreaks. For example, we could use a non-uniform prior which penalizes highly elongated shapes based on a geometric measure of compactness, as in Duczmal et al. [41, 39]. Alternatively, we could use an *empirical Bayes* approach in which the region prior is learned from data. One possible method would be to examine the *upper level sets* (all cells with $\frac{c_i}{b_i} > k$ for some threshold k), gradually lower the threshold k, and discover what shapes emerge.

For the parameter priors, we assume that we have access to a large number of days of past data, during which no outbreaks are known to have occurred. We can then obtain estimated values of the parameter priors under the null hypothesis by matching the first and second moments of each Gamma distribution to their estimated values from historical data.[1] In other words, we set the mean and variance of the distribution Gamma$(\alpha_{all}, \beta_{all})$ to the estimated mean and variance of the rate parameter q_{all} observed in past data: $\frac{\alpha_{all}}{\beta_{all}} = \hat{\text{E}}\left[q_{all}\right]$, and $\frac{\alpha_{all}}{\beta^2_{all}} = \hat{\text{Var}}\left[q_{all}\right]$. Solving for α_{all} and β_{all}, we obtain $\alpha_{all} = \frac{\left(\hat{\text{E}}[q_{all}]\right)^2}{\hat{\text{Var}}[q_{all}]}$ and $\beta_{all} = \frac{\hat{\text{E}}[q_{all}]}{\hat{\text{Var}}[q_{all}]}$.

Since the values of q_{all} are not known for the historical data, we consider two possible methods of computing the estimated mean and variance of q_{all}. First, since the maximum likelihood estimate of q_{all} is the ratio of total count to total baseline $\frac{C_{all}}{B_{all}}$, we can use the sample mean and sample variance of this ratio as estimates of the distribution of q_{all}: $\hat{\text{E}}\left[q_{all}\right] = \text{E}_{sample}\left[\frac{C_{all}}{B_{all}}\right]$, and $\hat{\text{Var}}\left[q_{all}\right] = \text{Var}_{sample}\left[\frac{C_{all}}{B_{all}}\right]$. This results in an unbiased estimate of the mean of q_{all}, but a conservatively biased estimate (overestimate) of the variance of q_{all}. Thus we call this approach the "conservative Bayes" (CBayes) method.

To obtain an unbiased estimate of the variance of q_{all}, we note that the observed variance of $\frac{C_{all}}{B_{all}}$ can be broken into the sum of two components, one resulting from the variation in $\frac{C_{all}}{B_{all}}$ given q_{all} and one resulting from the variation in q_{all}. In other words, we have:

$$\begin{aligned}\text{Var}\left[\frac{C_{all}}{B_{all}}\right] &= \text{E}\left[\text{Var}\left[\frac{C_{all}}{B_{all}} \mid q_{all}\right]\right] + \text{Var}\left[\text{E}\left[\frac{C_{all}}{B_{all}} \mid q_{all}\right]\right] \\ &= \text{E}\left[\text{Var}\left[\frac{\text{Poisson}(q_{all}B_{all})}{B_{all}} \mid q_{all}\right]\right] + \text{Var}\left[\text{E}\left[\frac{\text{Poisson}(q_{all}B_{all})}{B_{all}} \mid q_{all}\right]\right] \\ &= \text{E}\left[\frac{q_{all}B_{all}}{B^2_{all}}\right] + \text{Var}\left[\frac{q_{all}B_{all}}{B_{all}}\right] = \text{E}\left[\frac{q_{all}}{B_{all}}\right] + \text{Var}\left[q_{all}\right] = \text{E}\left[\frac{C_{all}}{B^2_{all}}\right] + \text{Var}\left[q_{all}\right]\end{aligned}$$

Thus we set $\hat{\text{E}}\left[q_{all}\right] = \text{E}_{sample}\left[\frac{C_{all}}{B_{all}}\right]$ as in the CBayes approach, but now we set the variance $\hat{\text{Var}}\left[q_{all}\right] = \text{Var}_{sample}\left[\frac{C_{all}}{B_{all}}\right] - \text{E}_{sample}\left[\frac{C_{all}}{B^2_{all}}\right]$. We call this approach to prior selection the "unbiased Bayes" (UBayes) method.

We have now described two methods for calculation of the global parameter priors, α_{all} and β_{all}. The calculation of priors $\alpha_{in}(S)$, $\beta_{in}(S)$, $\alpha_{out}(S)$, and $\beta_{out}(S)$ is identical except for two differences: first, we must condition on the observed rates inside or outside region S respectively, and second, we must assume the alternative hypothesis $H_1(S)$ rather than the null hypothesis H_0. Repeating the above derivation for the "out" parameters, we obtain $\alpha_{out}(S) = \frac{\left(\hat{\text{E}}[q_{out}(S)]\right)^2}{\hat{\text{Var}}[q_{out}(S)]}$ and $\beta_{out}(S) =$

[1] Note that the current data is not used to estimate the α and β. Thus our method differs from "empirical Bayes" methods that use the same data for estimating priors and computing likelihoods; nevertheless, our method is still "empirical" in the sense that our priors are data-driven.

$\frac{\hat{\text{E}}[q_{out}(S)]}{\hat{\text{Var}}[q_{out}(S)]}$. Then for the CBayes and UBayes methods, we have $\hat{\text{E}}\left[q_{out}(S)\right] = \text{E}_{sample}\left[\frac{C_{out}(S)}{B_{out}(S)}\right]$, where $C_{out}(S)$ and $B_{out}(S)$ are respectively the total count $\sum_{G-S} c_i$ and total baseline $\sum_{G-S} b_i$ outside the region. For UBayes, we have $\hat{\text{Var}}\left[q_{out}(S)\right] = \text{Var}_{sample}\left[\frac{C_{out}(S)}{B_{out}(S)}\right] - \text{E}_{sample}\left[\frac{C_{out}(S)}{B^2_{out}(S)}\right]$, and for CBayes, we have $\hat{\text{Var}}\left[q_{out}(S)\right] = \text{Var}_{sample}\left[\frac{C_{out}(S)}{B_{out}(S)}\right]$.

Our derivation for the “in” parameters is very similar, with one major difference: we must account for the impact of an outbreak on the disease rate inside region S. Recall that our historical data is assumed to have no outbreaks, and thus gives us an estimate of the prior distribution of disease rate inside region S when no outbreak is occurring. We assume that the outbreak will increase q_{in} by a multiplicative factor m; to account for this in the Gamma distribution Gamma$(\alpha_{in}, \beta_{in})$, we multiply α_{in} by m while leaving β_{in} unchanged. Thus we obtain $\alpha_{in}(S) = \frac{m\left(\hat{\text{E}}[q_{in}(S)]\right)^2}{\hat{\text{Var}}[q_{in}(S)]}$ and $\beta_{in}(S) = \frac{\hat{\text{E}}[q_{in}(S)]}{\hat{\text{Var}}[q_{in}(S)]}$. Then for the CBayes and UBayes methods, we have $\hat{\text{E}}\left[q_{in}(S)\right] = \text{E}_{sample}\left[\frac{C_{in}(S)}{B_{in}(S)}\right]$, where $C_{in}(S)$ and $B_{in}(S)$ are respectively the total count $\sum_S c_i$ and total baseline $\sum_S b_i$ inside the region. For UBayes, we have $\hat{\text{Var}}\left[q_{in}(S)\right] = \text{Var}_{sample}\left[\frac{C_{in}(S)}{B_{in}(S)}\right] - \text{E}_{sample}\left[\frac{C_{in}(S)}{B^2_{in}(S)}\right]$, and for CBayes, we have $\hat{\text{Var}}\left[q_{in}(S)\right] = \text{Var}_{sample}\left[\frac{C_{in}(S)}{B_{in}(S)}\right]$. Since we typically do not know the exact value of m, here we use a discretized uniform distribution for m, ranging from $m = 1 \ldots m_{max}$ at intervals of Δm.[2] Then scores can be calculated by averaging likelihoods over the distribution of m.

5.3.2 Computational considerations

As discussed above, a naïve approach to calculating the Bayesian spatial scan statistic requires us to calculate the score function $F(S) = \text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))$ for each spatial region S. Thus, if we search over the space of all axis-aligned rectangular regions in an $N \times N$ grid, we must search $O(N^4)$ regions. As in Chapter 3, we can search each region in $O(1)$ by preconstructing a grid of the cumulative counts $cc_{ij} = \sum_{k=1\ldots i}\sum_{l=1\ldots j} c_{kl}$, and similarly for the baselines. Then the total count or total baseline of a region may be calculated by adding/subtracting at most four cumulative counts, regardless of the size of the region. Thus the total time to search an $N \times N$ grid G is $O(N^4)$, and since in the Bayesian approach we do not need to do randomization testing, this is the total complexity of our algorithm.

We can speed up our search by applying the “fast spatial scan” algorithm of Chapter 3, allowing us to rapidly find the region S^* with the highest score $F(S)$. The fast spatial scan uses a top-down, branch-and-bound search to prune regions that cannot have the highest score, thus allowing us to find S^* while searching only a small fraction of regions. A novel multiresolution data structure known as an overlap-kd tree enables efficient search, resulting in 20-2000x speedups on a variety of real-world datasets.

However, two issues make it difficult to apply the fast spatial scan in the Bayesian framework. First, we must ensure that the criteria of Chapter 3 hold: the score function must increase with the total count of a region, decrease with the total baseline of a region, and (for a constant ratio of count to baseline) increase with count and baseline. It can be proven that the likelihood $\text{Pr}(D \mid H_1(S))$

[2] In our experiments, we use $m_{max} = 3$ and $\Delta m = 0.2$.

meets these criteria, so for the uniform region prior given above, the score function will also meet these criteria, and the fast spatial scan can be used. For non-uniform priors, this may not be the case, so we must adjust our upper bound accordingly. More precisely, in the non-uniform case, we can find an upper bound of $F(S')$ for all regions $S' \subset S$ by upper bounding both the likelihood Pr$(D \mid H_1(S'))$ and the prior Pr$(H_1(S'))$. Then we can prune a set of regions S' if the upper bound on $F(S')$ is lower than the highest score found so far.

A second issue which complicates the application of the fast spatial scan method is that, in the Bayesian framework, calculation of posterior probabilities requires us to compute and divide by the likelihood of the data Pr(D), which necessitates computing the sum of scores for all spatial regions S. This makes pruning difficult, since pruned regions may add a significant amount of probability mass to the total. There are three possible solutions to this problem. First, we can do less pruning: we can bound the maximum total contribution of a set of regions S' to the probability of the data, and only prune these regions if their total probability is guaranteed to be small. A second solution would be to assume an empirical Bayesian prior Pr$(H_1(S))$ that is equal to zero for any region that is pruned, and only gives probability to unpruned regions. In the uniform region prior case, we can set Pr$(H_1(S)) = \frac{P_1}{n_{reg}}$ if region S is searched, and Pr$(H_1(S)) = 0$ if region S is pruned, where n_{reg} is the total number of regions searched (not pruned). A third alternative is to work with the posterior odds ratios $\frac{\text{Pr}(H_1(S) \mid D)}{\text{Pr}(H_0 \mid D)} = \frac{\text{Pr}(D \mid H_1(S))\text{Pr}(H_1(S))}{\text{Pr}(D \mid H_0)\text{Pr}(H_0)}$ instead of the posterior probabilities. This is useful because computation of the denominator Pr(D) is not required, and the region S^* with highest posterior odds ratio also has the highest posterior probability. Moreover, we can easily compute a lower bound on the posterior probability of an outbreak given the posterior odds ratio: for a region with a posterior odds ratio of x, the posterior outbreak probability is at least $\frac{x}{1+x}$. A tighter lower bound may be achieved by maintaining a list of the k-best regions, giving a posterior outbreak probability of at least $\frac{\sum_{i=1...k} x_i}{1+\sum_{i=1...k} x_i}$.

5.4 Results: detection power

We evaluated the Bayesian and frequentist methods on two types of simulated respiratory outbreaks, injected into real Emergency Department and over-the-counter drug sales data for Allegheny County, Pennsylvania. All data were aggregated to the zip code level to ensure anonymity, giving the daily counts of respiratory ED cases and sales of OTC cough and cold medication in each of 88 zip codes for one year.

For these datasets, we are given only a count c_i^t for each zip code s_i for each day t, and the baselines are not known a priori. Thus we first infer the baselines b_i^t for each zip code s_i for each day t, using the mean count of the previous 28 days: $b_i^t = \frac{1}{28} \sum_{x=1...28} b_i^{t-x}$. We then use these counts and baselines to compute the alpha and beta priors as above, using eight weeks of past data. For example, $\hat{\text{E}}\left[q_{all}\right] = \text{E}_{sample}\left[\frac{C_{all}}{B_{all}}\right] = \frac{1}{56} \sum_{t=t_0-56...t_0-1} \frac{C_{all}^t}{B_{all}^t}$, where C_{all}^t and B_{all}^t denote the total count and baseline respectively for day t, and t_0 denotes the current day. Zip code centroids were mapped to a 16×16 grid (i.e. all counts for each zip code were mapped to the grid cell containing the centroid of that zip code), and all rectangles up to 8×8 were examined.

We first considered simulated aerosol releases of inhalational anthrax (e.g. from a bioterrorist attack), generated by the Bayesian Aerosol Release Detector, or BARD [70]. The BARD simulator uses a Bayesian network model to determine the number of spores inhaled by individuals in affected areas, the resulting number and severity of anthrax cases, and the resulting number of respiratory

Table 5.1: Days to detect and proportion of outbreaks detected, 1 false positive/month

	FLOO_ED	FLOO_ED	FLOO_ED	BARD_ED	BARD_ED	FLOO_OTC	FLOO_OTC
method	(4,14)	(2,20)	(1,20)	(.125)	(.016)	(40,14)	(25,20)
frequentist	1.859 (100%)	3.324 (100%)	6.122 (96%)	1.733 (100%)	3.925 (88%)	3.582 (100%)	5.393 (100%)
CBayes_max	**1.740** **(100%)**	**2.875** **(100%)**	**5.043** **(100%)**	**1.600** **(100%)**	**3.755** **(88%)**	5.455 (63%)	7.588 (79%)
UBayes_max	**1.710** **(100%)**	**2.848** **(100%)**	**4.875** **(100%)**	**1.633** **(100%)**	**3.679** **(88%)**	5.461 (63%)	7.588 (79%)
CBayes_tot	1.882 (100%)	3.195 (100%)	5.777 (100%)	**1.633** **(100%)**	3.811 (88%)	**3.475** **(100%)**	**5.195** **(100%)**
UBayes_tot	1.847 (100%)	3.184 (100%)	5.516 (100%)	**1.633** **(100%)**	3.811 (88%)	**3.475** **(100%)**	**5.195** **(100%)**

ED cases on each day of the outbreak in each affected zip code. Our second type of outbreak was a simulated "Fictional Linear Onset Outbreak" (or "FLOO"), as in Chapter 4. A FLOO(Δ, T) outbreak is a simple simulated outbreak with duration T, which generates $t\Delta$ cases in each affected zip code on day t of the outbreak ($0 < t \leq T/2$), then generates $T\Delta/2$ cases per day for the remainder of the outbreak. Thus we have an outbreak where the number of cases ramps up linearly and then levels off. While this is clearly a less realistic outbreak than the BARD-simulated anthrax attack, it does have several advantages: most importantly, it allows us to precisely control the slope of the outbreak curve and examine how this affects our methods' detection ability.

To test detection power, a semi-synthetic testing framework similar to Chapter 4 was used: we first run our spatial scan statistic for each day of the last nine months of the year (the first three months are used only to estimate baselines and priors), and obtain the score F^* for each day. Then for each outbreak we wish to test, we inject that outbreak into the data, and obtain the score $F^*(t)$ for each day t of the outbreak. By finding the proportion of baseline days with scores higher than $F^*(t)$, we can determine the proportion of false positives we would have to accept to detect the outbreak on day t. This allows us to compute, for any given level of false positives, what proportion of outbreaks can be detected, and the mean number of days to detection. We compare three methods of computing the score F^*: the frequentist method (F^* is the maximum likelihood ratio $F(S)$ over all regions S), the Bayesian maximum method (F^* is the maximum posterior probability Pr$(H_1(S) \mid D)$ over all regions S), and the Bayesian total method (F^* is the sum of posterior probabilities Pr$(H_1(S) \mid D)$ over all regions S, i.e. total posterior probability of an outbreak). For the two Bayesian methods, we consider the CBayes and UBayes methods for calculating priors, thus giving us a total of five methods to compare. In Table 5.1, we compare these methods with respect to proportion of outbreaks detected and mean number of days to detect, at a false positive rate of 1/month. Methods were evaluated on seven types of simulated outbreaks: three FLOO outbreaks on ED data, two FLOO outbreaks on OTC data, and two BARD outbreaks (with different amounts of anthrax release) on ED data. For each outbreak type, each method's performance was averaged over 100 or 256 simulated outbreaks for BARD or FLOO respectively. In Table 5.1, the best-performing methods for each dataset are shown in bold type; these include the method with lowest average time to detection, as well as any method whose performance is not significantly different (using a paired t-test with $\alpha = .05$).

In Table 5.1, we observe very different results for the ED and OTC datasets. For the five runs on ED data, all four Bayesian methods consistently detected outbreaks faster than the frequentist method. This difference was most evident for the more slowly growing (harder to detect) outbreaks, especially FLOO(1,20). Across all ED outbreaks, the Bayesian methods showed an mean improvement of between 0.24 days (CBayes_tot) and 0.55 days (UBayes_max) as compared to the frequentist approach; "max" methods performed substantially better than "tot" methods, and "UBayes" methods performed slightly better than "CBayes" methods. For the two runs on OTC data, on the other hand, the CBayes_max and UBayes_max methods performed much worse (over two days slower) than the frequentist method. On the other hand, the CBayes_tot and UBayes_tot methods again outperformed the frequentist method, by an average of 0.15 days. We believe that the main reason for these differing results is that the OTC data is much noisier than the ED data, and exhibits much stronger seasonal trends. As a result, our baseline estimates (using mean of the previous 28 days) are reasonably accurate for ED, but for OTC the baseline estimates will lag behind the seasonal trends (and thus, underestimate the expected counts for increasing trends and overestimate for decreasing trends). The "max" methods perform badly on the OTC data because a large number of baseline days have the total posterior probability of an outbreak close to 1. In this case, the maximum region posterior varies wildly from day to day, depending on how much of the total probability is assigned to a single region, and is not a reliable measure of whether an outbreak has occurred. On the other hand, the total probability of an outbreak will still be (slightly) higher for outbreak than non-outbreak days, so the "tot" methods can perform well on OTC as well as ED data. Thus, our main result is that the Bayesian methods CBayes_tot and UBayes_tot, which use the total posterior probability of an outbreak to decide when to sound the alarm, consistently outperform the frequentist method for both ED and OTC datasets.

5.5 Results: computation time

As noted above, the Bayesian spatial scan must search over all rectangular regions for the original grid only, while the frequentist scan (in order to calculate statistical significance by randomization) must also search over all rectangular regions for a large number (typically $R = 999$) of replica grids. Thus, as long as the search time per region is comparable for the Bayesian and frequentist methods, we expect the Bayesian approach to be approximately 1000x faster. In Table 5.2, we compare the run times of the Bayesian and frequentist methods for searching a single grid and calculating significance (p-values or posterior probabilities for the frequentist and Bayesian methods respectively), as a function of the grid size N. We note that the speed of the various Bayesian methods (CBayes vs. UBayes, "tot" vs. "max") is essentially identical, so we do not differentiate between these in the table. All rectangles up to size $N/2$ were searched, and for the frequentist method $R = 999$ replications were performed. The results confirm our intuition: the Bayesian methods are 900-1200x faster than the frequentist approach, for all values of N tested. However, the frequentist approach can be accelerated dramatically using the "fast spatial scan" algorithm discussed in Chapter 3. Comparing the fast spatial scan to the Bayesian approach, we see that the fast spatial scan scales better as a function of grid size: thus it is faster than the Bayesian approach for sufficiently large grid sizes ($N \geq 256$), but slower for smaller grids. Either method can search a 256×256 grid, and calculate significance (p-values or posteriors respectively) in 10-12 hours, as compared to months for the standard (naïve frequentist) approach. Thus we now have two ways to make the spatial scan computationally feasible for large datasets: to apply the fast spatial scan discussed in Chapter 3, or to use the Bayesian framework presented here. For even larger grid sizes, it may be possible to extend the fast spatial scan to the Bayesian framework: this would give us

Table 5.2: Comparison of run times for varying grid size N

method	$N = 16$	$N = 32$	$N = 64$	$N = 128$	$N = 256$
Bayesian (naïve)	0.7 sec	10.8 sec	2.8 min	44 min	12 hrs
frequentist (naïve)	12 min	2.9 hrs	49 hrs	∼31 days	∼500 days
frequentist (fast)	20 sec	1.8 min	10.7 min	77 min	10 hrs

the best of both worlds, searching only a single grid, and using a fast algorithm to do so. We are currently investigating this potentially useful synthesis, and we discuss this possibility in more detail in Chapter 8.

5.6 Discussion

We have presented a Bayesian spatial scan statistic, and demonstrated several ways in which this method is preferable to the standard (frequentist) scan statistic approach. In Section 5.4, we demonstrated that the Bayesian method, with a relatively non-informative prior distribution, consistently outperforms the frequentist method with respect to detection power. Since the Bayesian framework allows us to easily incorporate prior information about size, shape, and impact of an outbreak, it is likely that we can achieve even better detection performance using more informative priors, e.g. obtained from experts in the domain. In Section 5.5, we demonstrated that the Bayesian spatial scan can be computed in much less time than the naïve frequentist method, since randomization testing is unnecessary. This allows us to search large grid sizes using a naïve search algorithm, and even larger grids might be searched by extending the fast spatial scan to the Bayesian framework.

We now consider three other arguments for use of the Bayesian spatial scan. First, the Bayesian method has easily interpretable results: it outputs the posterior probability that an outbreak has occurred, and the distribution of this probability over possible outbreak regions. This makes it easy for a user (e.g. public health official) to decide whether to investigate each potential outbreak based on the costs of false positives and false negatives; this type of decision analysis cannot be done easily in the frequentist framework. Another useful result of the Bayesian method is that we can compute a "map" of the posterior probabilities of an outbreak in each grid cell, by summing the posterior probabilities Pr$(H_1(S) \mid D)$ of all regions containing that cell. This technique allows us to deal with the case where the posterior probability mass is spread among many regions, by observing cells which are common to most or all of these regions. We give an example of such a map in Figure 5.1.

Second, calibration of the Bayesian statistic is easier than calibration of the frequentist statistic. As noted above, it is simple to adjust the sensitivity and specificity of the Bayesian method by setting the prior probability of an outbreak P_1, and then we can "sound the alarm" whenever posterior probability of an outbreak exceeds some threshold. In the frequentist method, on the other hand, many regions in the baseline data have sufficiently high likelihood ratios that no replicas beat the original grid; thus we cannot distinguish the p-values of outbreak and non-outbreak days. While one alternative is to "sound the alarm" when the likelihood ratio is above some threshold (rather than when p-value is below some threshold), this is technically incorrect: because the baselines for each day of data are different, the distribution of region scores under the null hypothesis will also differ from day to day, and thus days with higher likelihood ratios do not necessarily have lower p-values. Third, we argue that it is easier to combine evidence from multiple detectors within the Bayesian framework, i.e. by modeling the joint probability distribution. We are in the process

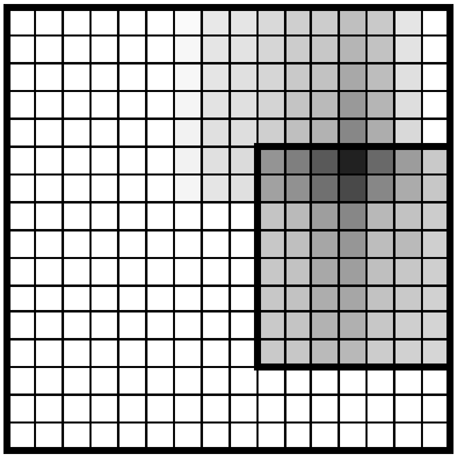


Figure 5.1: Output of Bayesian spatial scan on baseline OTC data, 1/30/05. Cell shading is based on posterior probability of an outbreak in that cell, ranging from white (0%) to black (100%). The bold rectangle represents the most likely region (posterior probability 12.27%) and the darkest cell is the most likely cell (total posterior probability 86.57%). Total posterior probability of an outbreak is 86.61%.

of examining Bayesian detectors which look simultaneously at the day's Emergency Department records and over-the-counter drug sales in order to detect emerging clusters, and we believe that combination of detectors is an important area for future research. We discuss this "multivariate Bayesian scan statistic" in more detail in the following section.

In conclusion, we note that, though both Bayesian modeling [28, 105] and (frequentist) spatial scanning [88, 78] are common in the spatial statistics literature, this is (to the best of our knowledge) the first model which combines the two techniques into a single framework. In fact, very little work exists on Bayesian methods for spatial cluster detection. One notable exception is the literature on spatial cluster modeling [51, 94], which attempts to infer the location of cluster centers by inferring parameters of a Bayesian process model. Our work differs from these methods both in its computational tractability (their models typically have no closed form solution, so computationally expensive MCMC approximations are used) and its easy interpretability of results. Thus we believe that this is the first Bayesian spatial cluster detection method which is powerful and useful, yet computationally tractable. We are currently running the Bayesian and frequentist scan statistics on daily OTC sales data from over 20,000 stores, searching for emerging disease outbreaks on a daily basis nationwide. Additionally, we are working to extend the Bayesian statistic to fMRI data, with the goal of discovering regions of brain activity corresponding to given cognitive tasks [156, 163, 118]. We believe that the Bayesian approach has the potential to improve both speed and detection power of the spatial scan in this domain as well.

5.7 The multivariate Bayesian scan statistic

We are currently working on a Bayesian multivariate cluster detection approach, the "multivariate Bayesian scan statistic" (MBSS). The primary goal of this work is to combine multiple data sources in a realistic statistical framework, in order to increase detection power and to distinguish between potential causes of a detected cluster.

Let us consider an example where we are monitoring three streams of data: over-the-counter sales of cough medication, over-the-counter sales of nasal decongestants, and respiratory emergency department visits. A standard spatial scan approach would perform a separate statistical test

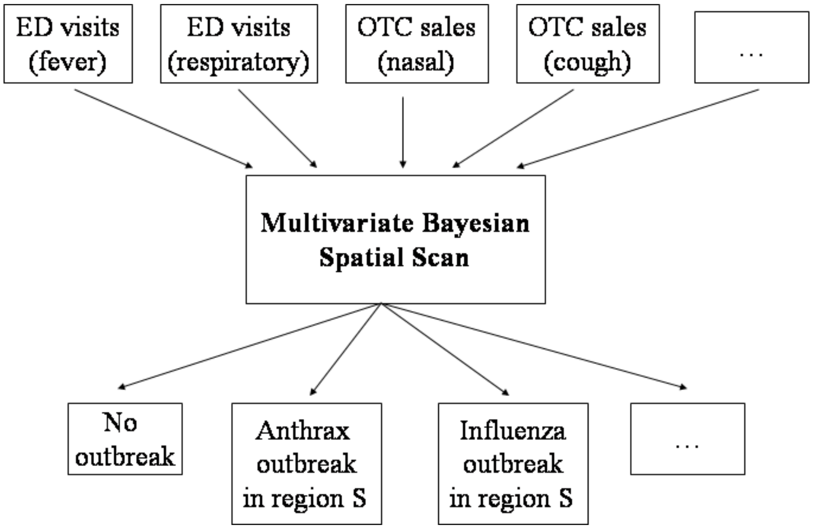

ED visits (fever)
ED visits (respiratory)
OTC sales (nasal)
OTC sales (cough)
...
Multivariate Bayesian Spatial Scan
No outbreak
Anthrax outbreak in region S
Influenza outbreak in region S
...

Figure 5.2: A pictorial diagram of the multivariate Bayesian scan statistic, showing multiple input streams and multiple types of outbreak.

for each of these data streams; this has the disadvantages of multiple testing (expected number of false positives is proportional to number of data streams) as well as making it difficult to interpret any positive results. Our proposed approach instead simultaneously monitors all of the data streams, computing the joint probability of all observed data under normal conditions and in the presence of various types of spatially localized outbreak. This gives us increased power to detect outbreaks that affect multiple data streams: for example, an outbreak of influenza-like illness is likely to increase the counts for all three streams (as well as other streams such as fever emergency department visits and over-the-counter thermometer sales). Thus by simultaneously monitoring all of these streams, we can detect an outbreak with proportionally smaller impact on the counts for each individual stream, thus allowing detection closer to the onset of the disease. Additionally, this simultaneous monitoring allows us to distinguish between different potential causes of a detected cluster of disease cases. For example, we would expect an outbreak of inhalational anthrax to affect those streams monitoring cough and fever symptoms, but not to have a major impact on sales of nasal decongestants, while influenza would lead to significant increases in all three symptom types.

In the MBSS framework, we are given a set of outbreak types $O = \{O_k\}$, $k = 1 \dots K$, and a set of data streams $\{d_m\}$, $m = 1 \dots M$. An example of such a model, with multiple outbreak types and multiple data streams, is given in Figure 5.2. The outbreak types may be either specific illnesses (influenza, anthrax, etc.) or non-specific syndromes (e.g. flu-like illness). The data streams may include sources such as ED visits (with each stream representing a different chief complaint, e.g. respiratory) and OTC drug sales (with each stream representing a different product group, e.g. nasal decongestants). We are also given a set of spatial regions S to search, where each S consists of a different set of spatial locations s_i. Finally, we are given the dataset $D = \{c^t_{i,m}\}$, where each $c^t_{i,m}$ is the count in spatial location s_i at time t for data stream d_m. Our goal, then, is to compute the posterior probability $\Pr(H_1(S, O_k) \mid D)$ that each outbreak type has affected each spatial region, given the multivariate dataset D.

Applying Bayes' Theorem, we obtain:

$$\Pr(H_1(S,O_k) \mid D) = \frac{\Pr(D \mid H_1(S,O_k))\Pr(H_1(S,O_k) \mid O_k)\Pr(O_k)}{\Pr(D)}$$

$$\Pr(H_0 \mid D) = \frac{\Pr(D \mid H_0)\Pr(H_0)}{\Pr(D)}$$

In this equation, $\Pr(H_0)$ is the prior probability of the null hypothesis (no outbreaks) and $\Pr(O_k)$ is the prior probability that outbreak type O_k has occurred. To simplify our calculations, we assume here that all outbreak types are mutually exclusive, and thus $\Pr(H_0) + \sum_k \Pr(O_k) = 1$; multiple simultaneous outbreaks can be dealt with as separate hypotheses. The probability $\Pr(H_1(S,O_k) \mid O_k)$ is the prior probability that outbreak type O_k will affect a given spatial region S. This distribution can be different for different outbreak types: for instance, the location of outbreaks of cryptosporidiosis or other water-borne illnesses can be predicted based on water distribution information; and we would expect a highly contagious disease such as avian influenza to affect a larger spatial area than a wind-dispersed outbreak of inhalational anthrax. Again, we assume that an outbreak affects exactly one spatial region, so we have $\sum_S \Pr(H_1(S,O_k) \mid O_k) = 1$. The most challenging part of our method is to compute the probability of the data (i.e. joint probability of all data streams) given each possible combination of outbreak and region, as well as the probability of the data under the null hypothesis of no outbreaks. We discuss this part of the method in more detail below. Finally, the normalizing factor $\Pr(D)$ can be computed by summing the products $\Pr(D \mid H)\Pr(H)$ for each hypothesis H.

To compute the probability of the data given the null hypothesis H_0 or an alternative hypothesis $H_1(S,O_k)$, we can use a Gamma-Poisson model as in the univariate Bayesian scan statistic discussed above. We assume that each count $c^t_{i,m}$ has been drawn from a Poisson distribution with mean $q^t_{i,m}b^t_{i,m}$, where $b^t_{i,m}$ is the "baseline" (or expected count) of stream m in spatial location s_i at time t, and $q^t_{i,m}$ is the "relative risk." Each baseline can be inferred from the time series of past counts for the given stream and given spatial location, using one of the time series analysis methods given in Chapter 4. These inferences can either be performed independently, or we can take into account the correlation between streams in the same or nearby locations. Under normal conditions, the relative risk $q^t_{i,m}$ is drawn from the Gamma prior distribution for that stream, $\text{Gamma}(\alpha_m, \beta_m)$, which is learned from the time series of past counts as above.

On the other hand, if an outbreak is present, the relative risks within the affected area will be drawn from a different Gamma prior with higher mean value. Each outbreak type will affect different data streams to different degrees, and some data streams may not be affected. The parameter prior distributions for each outbreak type can either be learned from available outbreak data, or estimated based on expert knowledge of that outbreak. Because a conjugate prior is used, we can derive a closed form solution for the marginal likelihood of the data under each hypothesis, efficiently computable as a function of the aggregate counts, aggregate baselines, and parameter priors, as above. We can then combine these likelihoods with the prior probabilities in order to obtain the posterior probability of each hypothesis.

As discussed in the following chapter, a number of other methods have been proposed in the biosurveillance literature for combining multiple data streams. However, these methods generally do not take spatial information into account, and do not allow discrimination between multiple types of outbreak. One exception is the work of Cooper et al. on PANDA [30], which models both anthrax and influenza, and uses emergency department records and over-the-counter medication sales to distinguish between these two types of outbreak. Our MBSS work differs from PANDA because it focuses on detecting spatial clusters of disease from aggregated data, while PANDA uses a Bayesian network representation and person-specific models but does not explicitly consider spatial data.

Chapter 6

Application to disease surveillance

6.1 Introduction

Epidemiologists have been analyzing biosurveillance data spatially since the seminal work of John Snow on the disease cholera [141]. During an 1854 epidemic of cholera in London, Snow discovered spatial clustering of cholera deaths around a single water pump. This enabled him to discover that cholera is caused by contaminated water, and to halt the epidemic by closing the contaminated pump. Since Snow's work, spatial statistical methods have come to play an increasingly large role in disease surveillance [44, 91]. In particular, spatial scan statistics [78] have become a well-used and thriving analytic method, owing in large part to the popularity of Martin Kulldorff's SaTScan software [87] in the public health community. Scan statistics have also been incorporated into several other experimental biosurveillance systems such as RODS [145], ESSENCE [98], BioSense [99], and many others [66, 167].

In this chapter, I will discuss the spatial disease surveillance task in more detail, and describe our new SSS (Spatial Scan Statistics) surveillance system. The SSS system was developed by myself and colleagues at the Auton Laboratory (Carnegie Mellon University) and RODS Laboratory (University of Pittsburgh), and is based on the new spatial cluster detection methods presented in this dissertation. This system enables us to monitor nationwide public health data (e.g. emergency department visits and over-the-counter drug sales) on a daily basis, searching for emerging outbreaks of disease. Every day, SSS receives data from over 20,000 stores and hospitals nationwide, uses our automatic cluster detection methods to find potential outbreaks of disease, and makes these results available to public health officials through a web-based graphical interface. We currently have several public health departments using our software to help them detect epidemics, and their feedback has been valuable for the iterative development of our system and the underlying models and methods. I am also working to integrate our cluster detection methods with several other systems for large-scale disease surveillance, in order to address not only spatial surveillance but other aspects of the disease surveillance task.

Before presenting the SSS system in Section 6.4 of this chapter, I will discuss the role of spatial disease surveillance in early detection of disease outbreaks. Section 6.2 discusses the importance of early detection and the need for spatial and syndromic surveillance, and Section 6.3 discusses the many challenges inherent in the spatial surveillance task. Sections 6.5 and 6.6 present results of the deployed SSS system. Section 6.5 discusses our experiences running the system for daily prospective surveillance, and presents some of the most interesting clusters detected. Section 6.6

is a detailed case study based on retrospective analysis of the Walkerton gastrointestinal outbreak. Finally, Section 6.7 presents a general overview of the biosurveillance literature, focusing primarily on spatial methods.

While many members of the Auton Laboratory and RODS Laboratory played significant roles in the development of the SSS system, I would most like to thank Maheshkumar Sabhnani for his efforts in implementing the system and developing the user interface. Additionally, I am grateful to Michael Wagner and the RODS Laboratory for their efforts in data collection and for making this data available to us. The description of the SSS system presented here has been adapted from our papers in ADKDD 2005 [131] and the 2005 National Syndromic Surveillance Conference [132]. I wish to thank my co-authors Maheshkumar Sabhnani, Andrew Moore, Michael Wagner, Rich Tsui, and Jeremy Espino for their contributions to these papers. Also, parts of this chapter have been adapted from our chapter in the *Handbook of Biosurveillance* [115]; I wish to thank my co-author Andrew Moore and editor Michael Wagner for their contributions. Finally, analysis of the Walkerton outbreak was performed in collaboration with Rick Davies and the ECADS collaborators; the discussion here has been adapted from our paper in the 2005 National Syndromic Surveillance Conference [32] and our paper [33].

6.2 Importance of spatial surveillance for early outbreak detection

Early detection of disease outbreaks is important for several reasons. First, we must deal with the very real, and scary, possibility of a bioterrorist attack– an intentional release of a deadly pathogen such as anthrax, smallpox, or bubonic plague. In 2001, letters containing anthrax spores were sent to various senate and media offices, causing five deaths. The World Health Organization (WHO) estimates that a large quantity (e.g. 100 kg) of aerosolized anthrax, released over a major city such as Washington, D.C., could kill between 1 million and 3 million people, and hospitalize millions more. A potentially even greater threat is that posed by emerging infectious diseases such as Severe Acute Respiratory Syndrome (SARS) or avian influenza. WHO has stated that avian influenza could lead to a global human pandemic, resulting in between 2 million and 7 million fatalities. This is widely considered to be a conservative estimate, and other estimates have put the number of potential fatalities as high as 150 million. A third reason for early detection is that it enables better epidemiological responses to many commonly occurring outbreaks (such as seasonal influenza and gastrointestinal outbreaks) which kill or hospitalize many thousands of people every year. Finally, we can detect and respond to patterns of symptoms due to other factors, such as environmental pollution, which may not be directly caused by pathogens.

We focus here on the case of a bioterrorist anthrax attack, and consider why early detection is important. Inhalational anthrax is a highly virulent disease: left untreated, it has approximately a 95% chance of being fatal within 2-3 weeks. However, anthrax is a treatable disease, and the earlier an affected patient is treated (e.g. with ciprofloxacin or other powerful antibiotics), the greater the chance of survival. Meselson et al. [102] estimate that there is a “window of opportunity” of approximately four days within which it is possible to mitigate the effects of an attack. Patients treated within the incubation period (the first 3-4 days, before any symptoms are present) have only a 1% chance of mortality, while the mortality rate climbs to 45% or higher once the patient becomes symptomatic. Early detection of an anthrax outbreak can lead to earlier treatment, both for individuals with early-stage symptoms, and also for those individuals who are currently asymptomatic but are likely to have been affected. One estimate, from DARPA, is that a two-day improvement in detection time over our current capabilities could reduce fatalities by a factor of six. Additionally, as Wagner et al. [149] note, improvements of even an hour over our current outbreak detection capabilities could reduce the economic impact of a bioterrorist anthrax attack by hundreds of millions of dollars. Thus early detection of anthrax could dramatically reduce the cost of the outbreak to society, both in money and in lives. For contagious diseases such as SARS or avian influenza, early detection and response could also dramatically reduce the spread of disease, reducing the number of individuals affected and possibly preventing a full-scale outbreak. Finally, early detection of bioterrorist attacks might have wide-ranging national security benefits, including capture of terrorists and prevention of further attacks.

While early detection of outbreaks is important, it is also difficult to achieve. The most common mode of detection (waiting for some astute physician to notice the outbreak and report it to public health) is often very slow, because the early-stage symptoms of many serious diseases are nonspecific. For example, the early symptoms of anthrax are flu-like, including cough and fever. Thus a physician is unlikely to be able to distinguish anthrax from influenza without results of tests such as a chest X-ray, and the physician is unlikely to call for such tests unless his suspicion has already been aroused. If the physician noticed a large increase in the number of patients reporting some set of symptoms, this might arouse suspicion, but since each individual physician or hospital only sees a small subset of the affected population, this indication of the outbreak might come too late to be useful. As a result, we could see over a week of lag time between the onset of symptoms from anthrax exposure and a definitive diagnosis of anthrax. On the other hand, individuals affected by the anthrax outbreak might display a number of early-stage behaviors which, when viewed in the aggregate, might be indicative of an outbreak. For example, an affected individual might buy over-the-counter drugs, including cough/cold and fever medications; he might be absent from work or school, and might visit a doctor, clinic, hospital, or emergency department. If a large number of individuals were affected in the same locale, we would observe increases in aggregate quantities such as the number of over-the-counter drugs sold or hospital visits. From these increases, we could infer that an outbreak was occurring, as well as pinpointing the affected region. Additionally, based on the population and region affected, and the symptom types indicated by these increases, we could infer a probability distribution over possible causes of the outbreak. By implementing a surveillance system to perform these tasks rapidly and automatically, we can receive early warnings of potential outbreaks with little or no human effort.

Thus one main argument of this dissertation is that we can achieve very early detection of outbreaks by gathering syndromic (or symptom) data, and automatically identifying emerging spatial clusters of symptoms. In collaboration with the RODS Laboratory at the University of Pittsburgh, we are currently gathering daily, nationwide health data including emergency department visits and over-the-counter drug sales; we can then apply the automatic cluster detection methods discussed above to identify clusters that are indicative of emerging outbreaks. We focus on the tasks of detecting outbreaks and pinpointing their locations; the task of differentiating between different types of outbreak is more difficult, but in Chapter 5, we presented some initial steps in this direction based on the multivariate Bayesian scan statistic (MBSS).

6.3 Challenges of spatial disease surveillance

While spatial disease surveillance has great potential as an analytical method for early detection of outbreaks, we must deal with many challenges to make these methods useful for real-world data.

In particular, we focus here on the monitoring of emergency department (ED) and over-the-counter drug sales (OTC) data. We consider many potential phenomena in these data streams which may cause either false positives (detection of clusters which are not epidemiologically relevant) or false negatives (failure to detect a true outbreak), and potential means of dealing with each of these phenomena. Some of these solutions have been built into our current implementation of the SSS system, as discussed in the following section, while others have been deferred to future versions of the system. We note that this discussion focuses only on the statistical and modeling challenges of applying our methods in the real world; a separate challenge is the computational problem of scaling our methods to massive nationwide datasets containing millions of records. Computational issues, and our approach to developing scalable and computationally efficient cluster detection methods, are discussed in detail in Chapter 3.

We can roughly divide the challenges of spatial disease surveillance into three groups: challenges related to data acquisition, challenges related to modeling "normal" baseline data (including all of the phenomena which may cause clusters but are not epidemiologically relevant), and challenges related to modeling outbreaks (and other "relevant" clusters). We discuss each of these challenges in detail in the following subsections.

6.3.1 Challenges of data acquisition

It is clear that even sophisticated models and methods will fail if the data provided is not sufficiently complete or reliable. As an extreme example, we will be completely unable to detect an outbreak if its effects are not present in the monitored data, either because we do not have any data for that region of the country, or because we are not monitoring the affected data streams. Our colleagues at the RODS Laboratory are working hard to increase the proportion of the country covered by our data feeds; we currently have high coverage on the East and West Coasts but lower coverage in the center of the United States.

Data irregularities are another serious problem, as many of these irregularities cause significant anomalies in the data which would be picked out by any anomaly detection algorithms. Irregularities in the OTC data were a major source of false positives in our early use of the SSS system, but have been reduced significantly by improvements to the National Retail Data Monitor. However, many irregularities are still present in the ED data we receive.

A third, and typically less serious, problem is that of missing data. Data are missing when a store fails to report the current day's OTC sales, or when a hospital fails to report the current day's ED visits. We have developed methods to impute the values of missing counts, using exponentially weighted linear regression or other methods of time series analysis to infer the expected counts under the assumption that no outbreak is occurring. Based on this conservative assumption, our power to detect an outbreak is reduced if many of the corresponding data are missing, but we are unlikely to encounter any false positives due to missing data. Our methods for dealing with missing data are discussed in Chapter 4.

As a final example of the challenges of obtaining the right data, we note that the power of our methods can be reduced due to disparities between the set of "clusters" we are searching and the population which was actually affected by the outbreak. For example, searching over only compact regions might cause us to miss elongated clusters, or searching over too coarse a resolution might cause us to miss very small (but epidemiologically relevant) clusters. As another example, if home zip codes are the only data in an emergency department's records, then an attack on a downtown office location might not appear as a spatial cluster. It is possible that appropriate use of commuting statistics [24, 40] can help in this case. Finally, if an outbreak only affects one segment of the population (e.g. children), our power to detect is reduced if we do not segment the population appropriately.

6.3.2 Modeling baseline data

A second set of challenges is posed by modeling "baseline data," i.e. the behavior of the monitored ED and OTC data streams when no outbreaks are occurring. In our early experiences of applying cluster detection to over-the-counter pharmacy data, it was immediately clear that simplistic assumptions in the underlying model can lead to false alarms: there are many non-disease-related reasons for clusters of over-the-counter medication purchases to occur. As a result, we must consider the many ways in which the real data does not correspond to our model assumptions, and either adjust the model or clean the data accordingly.

For example, day-of-week and seasonal trends must be incorporated into our time series analysis methods in order to obtain accurate baseline estimates of expected counts. If these trends are not accurately predicted, we will have increased likelihood of false positives when baselines are underestimated, and increased likelihood of false negatives when baselines are overestimated. A major difficulty relates to the fact that we are using inferred baseline values to perform anomaly detection, while our simplistic model treats these values as known rather than inferred. As a result, our anomaly detection methods will pick out not only real anomalies, but also regions where the baseline values have been underestimated. One way of dealing with this problem is to use conservative baseline values (e.g. intentionally overestimate baselines by some margin). For example, we could use the maximum (rather than mean) counts in historical data, or we could add some number of standard deviations to the inferred mean. A second approach would be to use the unbiased baseline values, but to "dial down" the sensitivity of the method by only reporting the most significant (highest scoring) clusters, and recalibrating their statistical significance based on the historical distribution of maximum scores. A third approach would be to account for the extra variance resulting from the inferred baselines (for example, using a t-distributed scan statistic rather than a Poisson or Normal), but this results in a more complicated statistic that is harder to efficiently compute.

More generally, our current statistics are likely to have an increased false positive rate due to many sources of model misspecification. Iterative improvements to the underlying model can reduce false positives due to many of these sources. For example, our current models assume a spatially uniform relative risk under the null hypothesis: this means that any spatial variation in risk should be reflected in the underlying baselines. An alternative would be to allow some variation in risk under the null to account for unmodeled, spatially-varying effects. Additionally, our current models do not account for spatial and temporal correlations, though the RATS aggregation method discussed in Chapter 4 is one way to deal with spatially correlated data. Using time series analysis methods such as ARIMA would be one way to deal with temporal autocorrelation. Finally, the typical Poisson model does not account for overdispersed data, though our Gaussian scan statistic model (derived in Chapter 2) can account for overdispersion. In practice, we find that OTC data (but not ED data) is highly overdispersed, requiring the use of a method that can account for overdispersion.

Finally, we consider some of the many phenomena which may cause clusters in ED and OTC data but are not epidemiologically relevant. One such phenomenon is promotional sales of over-the-counter medications: in this case, a store or chain sells large numbers of units not because people are sick, but because the medications are on sale. Our data feeds specify promoted versus non-promoted sales, so we can either filter out clusters due to promotions, or explicitly model the effects of a promotion on the counts of affected stores. A related phenomenon, which accounted for many false positives in early runs of our SSS system, was large spikes in the sales of individual stores. These spikes could have been due to promotions, to bulk purchases by a single buyer (e.g. a chain of hotels), or to inventory movements. One way of dealing with these spikes would be to count the number of transactions rather than the number of units sold, but this information is not currently available to us. Instead we consider two possible methods of dealing with single-store spikes: filtering out regions with increases resulting from a single store, and explicitly modeling single-store spikes. We discuss the filtering of single store increases (the “L-filter”) in the following section, and the use of a Bernoulli-Poisson store model in Chapter 2.

Other epidemiologically irrelevant clusters may be caused by unmodeled covariates; many of these false positives can be avoided by directly including the appropriate covariate as part of our model. A simpler, though less accurate, alternative is to filter out clusters resulting from these factors as a post-processing step. Examples of such covariates include holidays (typically counts drop during a major holiday, but are increased before and after the holiday), socio-demographic effects (in certain areas, sales may exhibit trends corresponding to social security checks or other monthly income sources), and weather (cough and cold sales are increased by cold weather; also, people tend to stock up on medications before and after severe weather). Another interesting effect, discussed in more detail in Section 6.5, is increased counts due to temporary movements of population: this could include populations displaced by severe weather (e.g. the devastation of New Orleans caused by Hurricane Katrina) or temporary population increases in popular tourist destinations. These effects could be dealt with directly if the populations were known, or indirectly by normalizing counts using sales of a baseline product such as soda or bottled water. Finally, some clusters of symptoms may correspond to already-known causes, whether known outbreaks (e.g. seasonal influenza) or environmental causes (e.g. wildfires in California). We want to be able to model expected counts resulting from known causes in order to detect other outbreaks which are simultaneously occurring. For example, we should be able to detect an anthrax attack even if it takes place in a region already affected by seasonal influenza. The use of the multivariate Bayesian (MBSS) approach should help us to model known outbreaks and distinguish these from other relevant clusters.

6.3.3 Challenges of modeling outbreaks and other relevant clusters

A third set of challenges is posed by modeling of outbreaks and other “relevant” clusters. We are typically not focused on detecting a specific type of outbreak, but instead we want to be able to detect any outbreak including those of previous unknown diseases. As a result, we focus on the modeling of “baseline” data as discussed above, and want to detect any significant increases in counts as compared to the baseline. Our use of spatial scan statistics does make some assumptions about the clusters we want to detect: most importantly, that they are spatially localized (i.e. we want to detect a spatial region of increased counts). Epidemics that display no spatial clustering will not be detected by spatial surveillance. Additional assumptions may be made by the individual statistics being used: for example, our persistent cluster statistic assumes a constant relative risk over the course of an outbreak, while the emerging cluster statistic assumes a monotonically increasing relative risk. Both methods assume that the relative risk due to an outbreak is spatially uniform in the affected region, but other statistics can be derived to allow spatially varying relative risk. In addition to these "general" detectors which find any significant deviations from the background data, we could also derive models for specific outbreak types that we are interested in detecting (e.g. avian influenza, anthrax). More accurate models of individual outbreak types could be included within the multivariate Bayesian scan statistic approach, giving us higher power to detect these specific outbreaks without reducing our power to detect more general outbreak patterns.

All of these factors present new and challenging opportunities for better modeling of ED and OTC data, and continued iterative improvements of our models and methods will improve our ability to differentiate real outbreaks from false positives. We note that even our current, simple methods (implemented within the SSS system discussed in Section 6.4) are able to provide useful information and to distinguish real outbreaks from false positives. This ability is demonstrated in our discussion of prospective surveillance using the SSS system (Section 6.5) and our retrospective analysis of the Walkerton gastroenteritis outbreak (Section 6.6).

6.4 Description of the SSS system for spatial disease surveillance

In this section, we present Spatial Scan Statistics (SSS), a new system for spatial disease surveillance. The SSS system was created in collaboration with Maheshkumar Sabhnani, Andrew Moore, Michael Wagner, Rich Tsui, and Jeremy Espino [131, 132], and implements many of the cluster detection methods discussed in this thesis. The SSS software is available for download from the Auton Laboratory website (www.autonlab.org), and the RODS Laboratory website (rods.health.pitt.edu).

Our current implementation of the SSS system monitors sales of over-the-counter (OTC) medications from over 20,000 stores throughout the nation. We can also use this system to monitor Emergency Department (ED) visits from thousands of hospitals nationwide, though at this point our ED data feeds are not quite as reliable. Thus our current implementation focuses on using the OTC sales for automatic detection of outbreaks, and ED data is used as a secondary source to investigate potential outbreaks. Monitoring is performed on a daily basis (currently with one day of lag time from the date of sale), enabling us to rapidly detect emerging clusters of disease. We are currently working to reduce the lag time, as well as to improve the quality of the ED data received. While our eventual goal is to simultaneously monitor and combine information from multiple data streams using the multivariate Bayesian scan statistic (MBSS) approach, our current system instead relies on the frequentist, expectation-based scan statistics discussed in Chapter 4, using the fast spatial scan algorithm of Chapter 3 to speed up our search as necessary.

The main purpose of our SSS system is to provide a tool for the automatic detection of emerging disease outbreaks. By providing intelligent algorithms to detect outbreaks in a timely manner, we hope to reduce the human and economic costs of outbreaks, whether due to a bioterrorist attack or a naturally occurring epidemic. A second purpose of our SSS system is to demonstrate that the general cluster detection techniques discussed in this thesis can achieve high performance not only on simulated data but on real public health data. In order to detect useful clusters in real-world data, we must cope with the many factors that make OTC and ED data difficult to model, including seasonal and day-of-week trends, missing data, and covariates such as weather, holidays, and promotional sales. Some of these modeling issues are incorporated into the current system, while others have been deferred to future versions of SSS.

Another challenge of building the SSS system results from the generality of our detection task. Our goal is to present public health users with all of the "interesting" clusters resulting from any phenomena of which they should be made aware, but to suppress any "uninteresting" clusters that are not epidemiologically relevant. Thus we typically do not have any specific model of what an outbreak looks like, and we also may not have models of the many “uninteresting” phenomena which may result in false positives. While we have enumerated (and considered possible solutions) for many of these phenomena in the previous section, this list is unlikely to be exhaustive, and more phenomena are likely to arise when incorporating other data sources into our models. One possible solution is provided by the multivariate Bayesian (MBSS) approach, in which we can specify a separate, scenario-based model for each outbreak type and for each uninteresting phenomenon, then combine these into a single model which enables us to differentiate between interesting and uninteresting clusters. Nevertheless, construction of all of these models is likely to be a time-consuming task requiring large amounts of expert knowledge, and it is extremely unlikely that even an expert will be able to identify and develop all the necessary models from scratch. Our eventual goal is to learn these models automatically from user feedback, but this is an extremely challenging learning task. A more immediate goal is to provide a tool that not only shows the detected clusters to the expert users, but also allows them to investigate and provide feedback on these clusters. This feedback loop can then be used for iterative refinement of our models and methods, leading to continual improvement of our detection and investigation tools and providing valuable insights into the complex process of disease outbreak detection.

Our system searches for spatio-temporal patterns in the over-the-counter drug sales from pharmacies, groceries, and other stores throughout the United States. Given some search region (which can be a city, county, state, or even the entire country), the algorithm first maps this search region to a uniform, rectangular $N \times N$ grid. It then searches over all axis-aligned rectangular regions on the grid, in order to find regions that have shown a recent anomalous increase in sales. As discussed in Chapter 4, our algorithm has two parts, first inferring the expected (baseline) sales for each grid cell and then detecting regions that show high deviation in sales from the estimated baselines. These detected regions are labeled as alerts– clusters of increased OTC sales that may indicate disease outbreaks. We use several variants of the expectation-based scan statistic for emerging clusters, including different temporal windows sizes W and different methods of time series analysis; more details of these methods are given in Chapter 4. Given our limited ability to distinguish clusters caused by outbreaks from clusters with other causes, we present selected alerts to public health officials only after they have been filtered by some simple rules to remove unimpressive anomalies. We can then incorporate the public health feedback to improve the performance of our system.

6.4.1 System overview

In Figure 6.1, we present an overview of our SSS system for prospective disease surveillance. This system can be divided into three major components: input (automatically gathering nationwide hospital and pharmacy data), analysis (performing automatic cluster detection on the input data streams), and output (making the detected clusters available for investigation by public health users). We now consider each component of this system in more detail.

Input to the SSS system is provided by the National Retail Data Monitor (NRDM), developed and operated by the RODS Laboratory at the University of Pittsburgh. The NRDM, described in detail in [148, 150], receives daily OTC data from the national and local vendors. This data consists of daily store level sales of 9000 OTC products used for the symptomatic treatment of infectious diseases. The NRDM groups individual product sales into 18 groups of similar products (e.g. baby/child electrolytes, cough/cold, thermometers, stomach remedies, and internal analgesics).

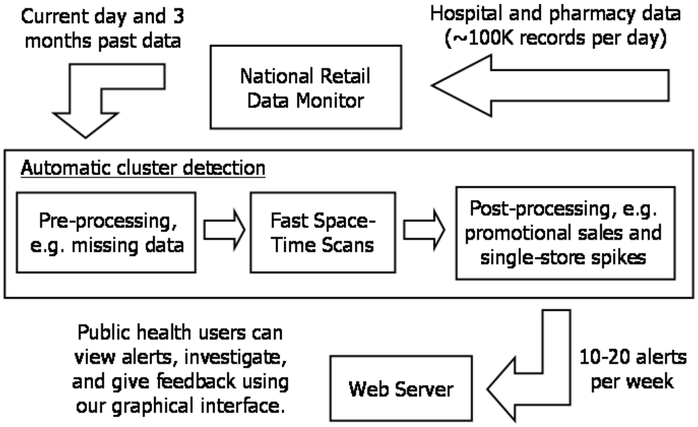

Current day and 3 months past data
National Retail Data Monitor
Hospital and pharmacy data (~100K records per day)
Automatic cluster detection
Pre-processing, e.g. missing data
Fast Space-Time Scans
Post-processing, e.g. promotional sales and single-store spikes
Public health users can view alerts, investigate, and give feedback using our graphical interface.
Web Server
10-20 alerts per week

Figure 6.1: Overview of the SSS system.

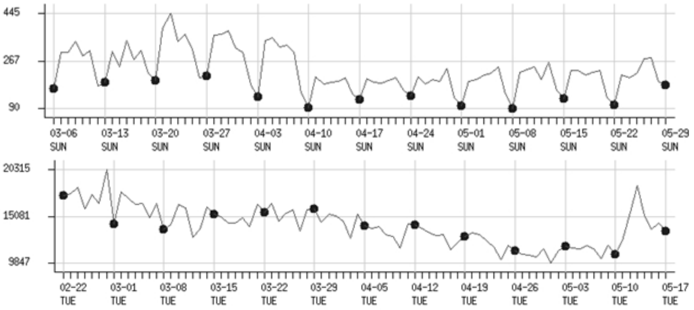


Figure 6.2: Examples of a) day-of-week and b) seasonal trends in over-the-counter sales data.

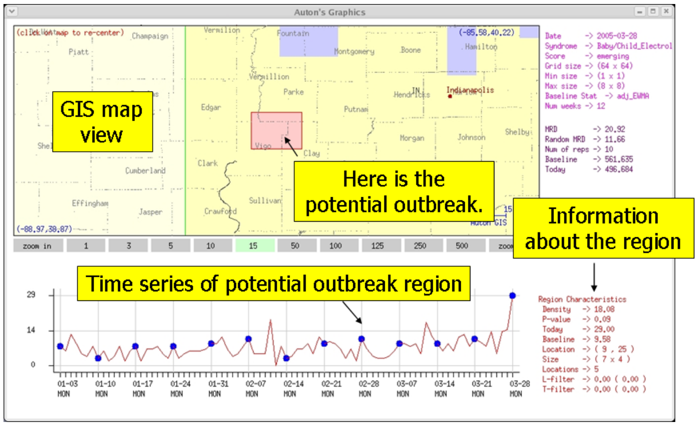
We receive data from the NRDM (from over 20,000 stores nationwide) on a daily basis, with a one-day delay from the date of sale. Each record includes the store ID, its corresponding zip code, date of sale, and units sold for a particular syndrome. In addition to receiving the current day's counts (number of units sold in each product group) for each store, we also obtain and process the past three months of data (around 5.5 million records) to estimate the baseline (i.e. number of sales we would expect to see) for each store. For space-time statistics with a larger temporal window, we also use the counts and baselines for up to seven days prior to the current day.

As discussed in the previous section, there are various challenges with estimating the store baseline sales. First, there are strong seasonal and weekly trends in the OTC data. Figure 6.2 shows a sample weekly trend in pediatric electrolyte sales. Sales on a typical Monday and Tuesday tend to be higher than on Friday and Saturday. The weekly trend exhibits spatial variation, depending on many unmodeled factors such as region of the country, urban or rural community, etc. Figure 6.2 also shows a sample seasonal trend in cough and cold medication sales. Average daily sales in the month of March were approximately 5000 units higher than in April. We have also noticed a sudden rise in sales for days following a national holiday. We address the seasonal and day-of-week trends by incorporating them into the baseline time series analysis. Missing data provides another challenge: the current data storage schema does not differentiate between missing data (i.e. stores that have not reported sales for a specific date by the time of analysis) and zero counts (i.e. stores that sold zero units on that date). To deal with this limitation, we assume that data are missing only if a store reports no sales for all product categories. If a store has zero counts for some product categories and non-zero counts for others, the zero counts are assumed to result from zero sales rather than from missing data. We infer all missing data points from the time series of counts for that location, using the exponentially weighted linear regression technique described in Chapter 4. Once the time series has no missing data, any reasonable univariate time series algorithm that accounts for day-of-week and seasonal trends can be applied to estimate recent baseline sales; see Chapter 4 for more details.

After we receive and pre-process the past three months of national OTC data, we define multiple search regions with differing resolution: in addition to performing a scan of the entire country, we also perform scans in individual states or counties. This provides scan results specifically tailored to interested state and local health departments; additionally, scanning over multiple resolutions ensures that we detect large-scale anomalies as well as clusters that are more spatially localized but still epidemiologically relevant. As noted above, the search region is mapped to a rectangular two-dimensional grid of size $N \times N$. We need to know the store locations in order to map them onto the grid cells; however, due to data privacy concerns, we do not have access to the exact longitude and latitude of each store. Instead, we are given the zip code containing each store, and use the longitude and latitude of the zip code centroid to populate the grid cells.

The search algorithm then scores every possible axis-aligned rectangular region using the recent baselines (expected counts) and observed counts in the region. Baseline values can be aggregated either for individual stores (the “building-aggregated time series” method, or BATS) for individual grid cells (the “cell-aggregated time series” method, or CATS), or on-the-fly for an entire search region (the “region-aggregated time series” method, or RATS). Additionally, a variety of methods are used for time-series analysis. Details on the aggregation techniques and time series analysis methods are given in Chapter 4. We also perform significance testing on the score of each region by randomization. This helps us remove anomalous regions that could be explained as being generated by chance. The k-best regions (i.e. those significant regions with the highest scores, and therefore the lowest p-values) are reported as possible disease outbreaks.

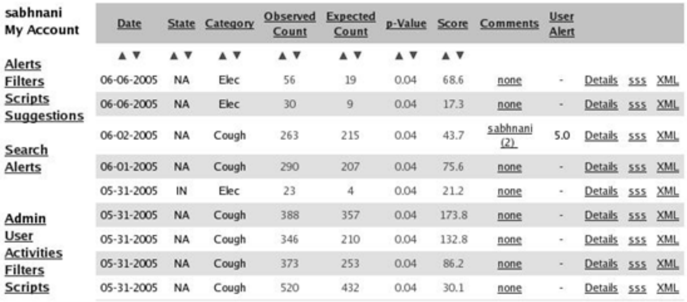
Once we have this set of potential outbreak regions, we perform two simple post-processing steps (“filters”) to remove regions due to uninteresting phenomena. We initially saw many false positives resulting from “single store” anomalies: individual stores with large spikes in sales on a given day. Two possible explanations for these single store anomalies are bulk purchases by a single buyer (e.g. restocking by a hotel, clinic, etc.) or promotional sales. We addressed this issue by only reporting those regions that have shown increased counts due to multiple stores: in other words, we filter out a region if removing any single store from that region would cause its score to become insignificant. This “location filter” (L-filter) is a simple, conservative method of dealing with unmodeled single-store phenomena. Other possible solutions would be to use the Bernoulli-Poisson model described in Chapter 2, or to produce detailed models of each individual single-store phenomenon. Second, we also saw many false positives due to slight increases in counts corresponding to a large spatial region. Any increase in disease rate, no matter how small, becomes statistically significant if it corresponds to a large enough underlying population or baseline; however, most public health officials are only interested in substantial increases in disease rate, so these slightly increased rates can be thought of as statistically, but not epidemiologically, significant. These increases could result from model misspecification, unmodeled covariates, or underestimates of the baseline for the given region. In order to make a simple adjustment for such potentially unmodeled fluctuations in day-to-day counts, we also apply a conservative “threshold filter” (T-filter), which



```
Date      -> 2005-03-28
Syndrome  -> Baby/Child_Electrol
Score     -> emerging
Grid size -> (64 x 64)
Min size  -> (1 x 1)
Max size  -> (8 x 8)
Baseline Stat  -> adj_EWMA
Num weeks -> 12

MRD          -> 20.92
Random MRD   -> 11.66
Num of reps  -> 10
Baseline     -> 561.635
Today        -> 496.684
```

```
Region Characteristics
 Density   -> 18.08
 P-value   -> 0.09
 Today     -> 29.00
 Baseline  -> 9.58
 Location  -> ( 9 , 25 )
 Size      -> ( 7 x 4 )
 Locations -> 5
 L-filter  -> 0.00 ( 0.00 )
 T-filter  -> 0.00 ( 0.00 )
```

Figure 6.3: Screen shot of the SSS viewer application, investigating a potential alert in Indiana.



sabhnani
My Account

Alerts
Filters
Scripts
Suggestions

Search
Alerts

Admin
User
Activities
Filters
Scripts

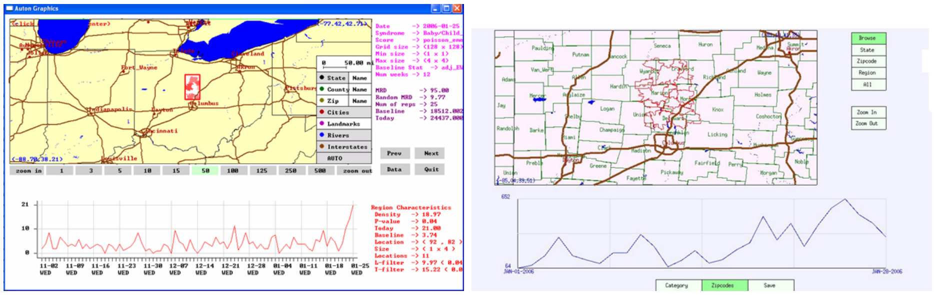
Date	State	Category	Observed Count	Expected Count	p-Value	Score	Comments	User Alert			
▲▼	▲▼	▲▼	▲▼	▲▼	▲▼	▲▼					
06-06-2005	NA	Elec	56	19	0.04	68.6	none	-	Details	sss	XML
06-06-2005	NA	Elec	30	9	0.04	17.3	none	-	Details	sss	XML
06-02-2005	NA	Cough	263	215	0.04	43.7	sabhnani (2)	5.0	Details	sss	XML
06-01-2005	NA	Cough	290	207	0.04	75.6	none	-	Details	sss	XML
05-31-2005	IN	Elec	23	4	0.04	21.2	none	-	Details	sss	XML
05-31-2005	NA	Cough	388	357	0.04	173.8	none	-	Details	sss	XML
05-31-2005	NA	Cough	346	210	0.04	132.8	none	-	Details	sss	XML
05-31-2005	NA	Cough	373	253	0.04	86.2	none	-	Details	sss	XML
05-31-2005	NA	Cough	520	432	0.04	30.1	none	-	Details	sss	XML

Figure 6.4: Screen shot of a user web page for SSS.

assumes that the baselines were underestimated by some specified amount (e.g. 10-15%). If both the "single-store" adjusted score and the "threshold" adjusted score are still significant, we report the region as a potential outbreak.

Once we have established the set of alerts to report, we must make these alerts available for investigation by the public health users. User testing for early versions of our system revealed that it was insufficient to present users with detected clusters without providing tools for investigation of these clusters. Thus we developed a web-based graphical interface which enables users to investigate, manage, and provide feedback on alerts. More precisely, our interface consists of two parts. First, we developed the SSS viewer tool, which allows users to investigate an alert by browsing the data on a GIS map and by "drilling down" into region-level and store-level time series data. A screen shot of the viewer tool is shown in Figure 6.3. Second, we developed the SSS web interface, which enables users to manage and track multiple alerts. The web interface also provides easy opportunities for users to provide and view feedback on individual alerts. This feedback has two functions: sharing the workload of investigating alerts between different end users, and providing us (the SSS developers) with user feedback on which alerts were genuine and which were uninteresting or due to non-outbreak reasons. A screen shot of the web interface is shown in Figure 6.4.

The current version of the web interface allows users to view alerts, rank their importance, add feedback comments, and give suggestions. Users can also search for alerts using different criteria, such as zip code, score, observed counts, and expected counts. Additionally, users can add their



```
Date          -> 2006-01-25
Syndrome      -> Baby/Child_
Score         -> poisson_eme
Grid size     -> (128 x 128)
Min size      -> (1 x 1)
Max size      -> (4 x 4)
Baseline Stat   -> adj_EW
Num weeks -> 12

MRD            -> 95.00
Random MRD     -> 9.77
Num of reps    -> 25
Baseline       -> 18512.002
Today          -> 24437.000

Region Characteristics
 Density    -> 18.97
 P-value    -> 0.04
 Today      -> 21.00
 Baseline   -> 3.74
 Location   -> ( 92 , 82 )
 Size       -> ( 1 x 4 )
 Locations -> 11
 L-filter   -> 9.97 ( 0.04
 T-filter   -> 15.22 ( 0.0
```

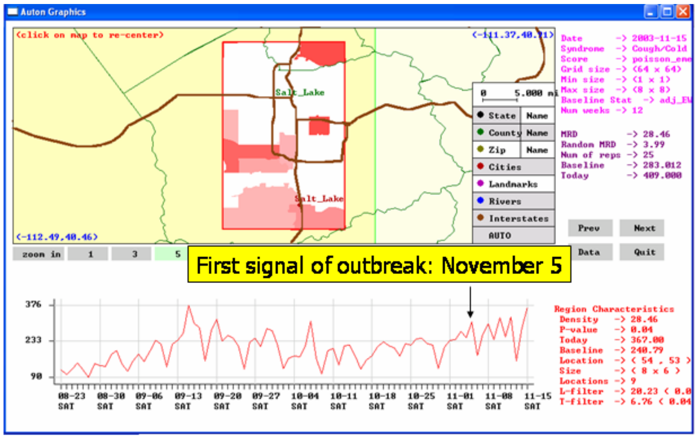
Figure 6.5: Alert in Columbus, OH, resulting from a possible GI outbreak. The left figure shows the increase in pediatric electrolyte sales detected by SSS, and the right figure shows a confirmatory increase in GI Emergency Department visits.

custom-defined input scripts to the pool of scripts that run daily. Users can set their own grid resolution, change baseline evaluation time series method, set aggregation level, etc. By enabling users to create their own input scripts and to give feedback on the resulting alerts, we hope to learn what results and settings are most relevant to real users in the surveillance task. This feedback will help us better manage these alerts and distinguish true outbreaks more efficiently. In the future, we also plan to provide more features (e.g. tracking of previously reported alerts for post analysis purposes) to the end users, thus improving their ability to investigate and manage alerts.

Finally, we plan to give users improved capability for ad hoc browsing and analysis of multiple, multivariate time series using the new TCUBE tool, currently under development by Maheshkumar Sabhnani and other colleagues in the Auton Laboratory [130]. By combining this tool with SSS, we hope to give users the flexibility to perform any desired investigations while also focusing their attention on the clusters that we believe to be most relevant.

6.5 SSS in practice: prospective surveillance and clusters detected

We now discuss our experiences running the SSS system for prospective surveillance of the nationwide OTC sales data. We have been running the SSS system daily on OTC data since late 2003. Initially, the system reported a large number of false positives, making it difficult for users to focus on the most relevant clusters. Two main improvements enabled us to significantly reduce the false positive rate. First, improvements in the data quality provided by the National Retail Data Monitor reduced the number of false positives due to data irregularities. Second, adding the post-processing filters discussed above enabled us to remove many false positives that were statistically significant according to our model but clearly did not correspond to actual outbreaks. We now obtain between 10 and 20 alerts per week. Some of these alerts can be diagnosed as likely to be due to data irregularities or model misspecifications, while others are of potential epidemiological relevance. These potentially relevant clusters then require further investigation, either by our team or by public health users, to determine whether or not the detected cluster corresponds to an actual outbreak or other, irrelevant phenomena. Because of limited manpower, our team was unable to investigate many such clusters, especially those that occurred in regions where we were not in contact with state and local public health departments. Nevertheless, we were able to discover a number of interesting and potentially relevant clusters, both due to disease outbreaks and due to other phenomena.



```
Date        -> 2003-11-15
Syndrome    -> Cough/Cold
Score       -> poisson_ene
Grid size -> (64 x 64)
Min size   -> (1 x 1)
Max size   -> (8 x 8)
Baseline Stat  -> adj_EW
Num weeks -> 12

MRD           -> 28.46
Random MRD  -> 3.99
Num of reps -> 25
Baseline      -> 283.012
Today         -> 409.000
```

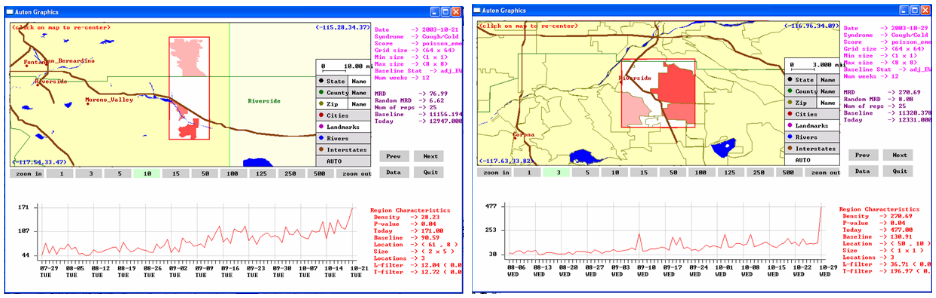
```
Region Characteristics
 Density    -> 28.46
 P-value    -> 0.04
 Today      -> 367.00
 Baseline   -> 240.79
 Location   -> ( 54 , 53 )
 Size       -> ( 8 x 6 )
 Locations -> 9
 L-filter   -> 20.23 ( 0.0
 T-filter   -> 6.76 ( 0.04
```

Figure 6.6: Alert in Salt Lake City, UT, resulting from an early outbreak of seasonal influenza. The figure shows the increase in cough/cold medication sales detected by SSS.

On January 25, 2006, the SSS system detected a spike in the sales of pediatric electrolytes near Columbus, Ohio. This increase is shown in Figure 6.5. We first did some preliminary investigation of the cluster using the SSS viewer tool. This investigation revealed that the increase emerged gradually over the course of January 23-25, was not limited to a single store or chain, was not due to promotional sales, and did not affect other categories of OTC sales. As a result of our preliminary investigation, we hypothesized that this increase resulted from a small, localized gastrointestinal outbreak starting January 23rd. Because we are not currently in contact with Ohio health officials, we were unable to obtain a definitive confirmation of this potential outbreak. However, we were able to obtain and analyze emergency department records for this area. As shown in Figure 6.5, gastrointestinal emergency department visits were also significantly increased between January 23 and 27, peaking on the 25th. Other types of ED visits were not significantly increased. This evidence supports our hypothesis of a small and localized GI outbreak.

Another relevant cluster that we found corresponded to an early, unusually severe outbreak of seasonal influenza that took place in the Salt Lake City area of Utah in November 2003. Our system observed the first signal of this outbreak on November 5, detecting an increase in cough and cold OTC sales in the Salt Lake City area. As shown in Figure 6.6, cough and cold sales remained high throughout November, triggering more alerts on the 12th, 14th, and 15th. Note that we also observe a false positive on September 15, which appears to have been due to a database error (we also observed many other false positives across the nation on this date). Such database errors were common in 2003 and 2004, but are now much less common due to improvements in the NRDM. It is also interesting to note that a more detailed study of the Utah outbreak, conducted by the RODS Laboratory and available on their website, suggests that the outbreak could have been detected sooner (possibly as early as October 25) by focusing on pediatric cough and cold sales. According to the RODS case study, the outbreak was visible from ED chief complaint monitoring (respiratory and constitutional categories) around November 7, suggesting that monitoring of OTC sales enabled more timely detection of this outbreak than ED visits.

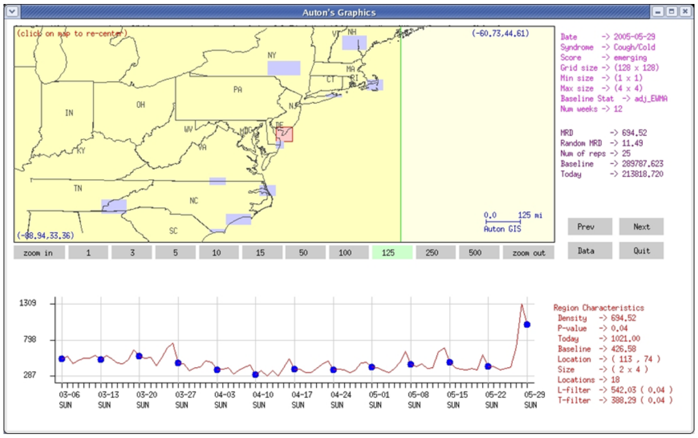
Another potentially relevant cluster that we found does not directly correspond to a disease outbreak, but instead was due to an environmental hazard. Between October 21 and November 4, 2003, a series of major, uncontrolled wildfires in southern California caused seriously degraded air


Auton Graphics
(click on map to re-center)
(-115.28,34.37)
Fontana San_Bernardino
Riverside
Moreno_Valley
Riverside
(-117.54,33.47)
10.00 mi
State Name
County Name
Zip Name
Cities
Landmarks
Rivers
Interstates
AUTO
zoom in 1 3 5 10 15 50 100 125 250 500 zoom out
Prev Next Data Quit
171
107
44
07-29 TUE 08-05 TUE 08-12 TUE 08-19 TUE 08-26 TUE 09-02 TUE 09-09 TUE 09-16 TUE 09-23 TUE 09-30 TUE 10-07 TUE 10-14 TUE 10-21 TUE
Auton Graphics
(click on map to re-center)
(-116.96,34.09)
Riverside
Corona
(-117.63,33.82)
3.000 mi
477
253
30
08-06 WED 08-13 WED 08-20 WED 08-27 WED 09-03 WED 09-10 WED 09-17 WED 09-24 WED 10-01 WED 10-08 WED 10-15 WED 10-22 WED 10-29 WED

```
Date      -> 2003-10-21
Syndrome  -> Cough/Cold
Score     -> poisson_eme
Grid size -> (64 x 64)
Min size  -> (1 x 1)
Max size  -> (8 x 8)
Baseline Stat  -> adj_EW
Num weeks -> 12

MRD          -> 76.99
Random MRD   -> 6.62
Num of reps  -> 25
Baseline     -> 11156.194
Today        -> 12947.000

Region Characteristics
 Density    -> 28.23
 P-value    -> 0.04
 Today      -> 171.00
 Baseline   -> 90.59
 Location   -> ( 61 , 8 )
 Size       -> ( 2 x 5 )
 Locations  -> 3
 L-filter   -> 12.04 ( 0.0
 T-filter   -> 12.72 ( 0.0
```

```
Date      -> 2003-10-29
Syndrome  -> Cough/Cold
Score     -> poisson_eme
Grid size -> (64 x 64)
Min size  -> (1 x 1)
Max size  -> (8 x 8)
Baseline Stat  -> adj_EW
Num weeks -> 12

MRD          -> 270.69
Random MRD   -> 8.08
Num of reps  -> 25
Baseline     -> 11320.378
Today        -> 12331.000

Region Characteristics
 Density    -> 270.69
 P-value    -> 0.04
 Today      -> 477.00
 Baseline   -> 130.91
 Location   -> ( 50 , 10 )
 Size       -> ( 1 x 1 )
 Locations  -> 3
 L-filter   -> 36.71 ( 0.0
 T-filter   -> 196.97 ( 0.
```

Figure 6.7: Two alerts in Riverside, CA, resulting from the 2003 wildfires, on October 21 and October 29 respectively. Both figures show increases in cough/cold medication sales detected by SSS.


Auton's Graphics
(click on map to re-center)
(-60.73,44.61)
NH
VT
NY
MA
CT
RI
PA
NJ
IN
OH
MD
DE
WV
VA
KY
TN
NC
SC
(-88.94,33.36)
0.0 125 mi
Auton GIS
zoom in 1 3 5 10 15 50 100 125 250 500 zoom out
Prev Next Data Quit
1309
798
287
03-06 SUN 03-13 SUN 03-20 SUN 03-27 SUN 04-03 SUN 04-10 SUN 04-17 SUN 04-24 SUN 05-01 SUN 05-08 SUN 05-15 SUN 05-22 SUN 05-29 SUN

```
Date      -> 2005-05-29
Syndrome  -> Cough/Cold
Score     -> emerging
Grid size -> (128 x 128)
Min size  -> (1 x 1)
Max size  -> (4 x 4)
Baseline Stat  -> adj_EWMA
Num weeks -> 12

MRD          -> 694.52
Random MRD   -> 11.49
Num of reps  -> 25
Baseline     -> 289787.623
Today        -> 213818.720

Region Characteristics
 Density    -> 694.52
 P-value    -> 0.04
 Today      -> 1021.00
 Baseline   -> 426.58
 Location   -> ( 113 , 74 )
 Size       -> ( 2 x 4 )
 Locations  -> 18
 L-filter   -> 542.03 ( 0.04 )
 T-filter   -> 388.29 ( 0.04 )
```

Figure 6.8: Alert showing increased sales in tourist destinations over Memorial Day weekend, 2005. SSS reported several alerts, including the highlighted alert in Rehoboth Beach, Delaware.

quality, leading to widespread increases in cough and cold medication sales. Our SSS system first detected the wildfires on October 21, the day that the fires started; we saw a large number of alerts throughout southern California. One such alert, in the Riverside area, is shown in Figure 6.7. We also saw many alerts in southern California between October 29 and October 31, corresponding to the period of worst air quality from the fires. One such alert, also in the Riverside area, is shown in Figure 6.7. It is interesting to note that OTC cough and cold medications are not an effective cure for respiratory problems due to smoke inhalation, but we were nevertheless able to pick up substantial evidence of the regions affected by smoke from the fires based on the patterns of cough and cold sales.

As a final example, we show one cluster detected by SSS that we do not consider to be epidemiologically relevant. During the 2005 Memorial Day weekend, we noted a large number of detected clusters along the eastern coast of the United States. Figure 6.8 displays one such cluster, in Rehoboth Beach, Delaware. Further investigation revealed that these alert regions corresponded to popular tourist destinations, especially beach resort areas. Thus it was clear that the combination of a long holiday weekend and warm weather led to a temporary increase in population due to an influx of tourists, resulting in increased OTC sales as well as increased ED visits. These false positives demonstrate the need to model and suppress clusters resulting from temporary population shifts. As noted in Section 6.3, two ways to deal with such clusters are to measure and adjust for population, if such information is available, or otherwise to normalize by total sales or sales of a baseline product.

Many of the epidemiologically irrelevant causes of clusters discussed in Section 6.3 have also been observed in the OTC data, including sales trends due to inclement weather (such as hurricanes in Florida). Although these are interesting results, they underscore the difficulty of determining which increases in sales are due to real outbreaks, and which increases are due to a variety of other unmodeled factors.

6.6 SSS case study: The Walkerton GI outbreak

Disease surveillance systems should be evaluated in a real-world setting before recommending their widespread deployment. This evaluation is difficult to accomplish prospectively, because disease outbreaks occur sporadically and are difficult to anticipate. Additionally, establishment of ground truth for evaluation is very difficult in prospective mode. However, evaluation of disease surveillance systems can be accomplished retrospectively using historical data from known and well-characterized outbreaks. We focus here on one such case study, an outbreak of gastroenteritis in Walkerton, Ontario, and examine the effectiveness of our SSS system for early detection of this outbreak.

The Walkerton outbreak occurred in the Grey-Bruce area of Ontario in May 2000, centered in the town of Walkerton, and resulted in an estimated 2321 cases of gastroenteritis. Eventually 1346 of these cases were individually identified. Of the 1346 identified cases, 65 individuals required hospitalization, 27 developed hemolytic-uremic syndrome, and six died. The first calls to the region's public health unit raising concern of an outbreak were made on Friday, May 19. This detection was made by an astute (and lucky) physician, who happened to observe multiple cases of pediatric bloody stools, a rare enough occurrence to trigger his suspicion. The first effective intervention was a presumptive boil water advisory issued on Sunday, May 21.

Our study of the Walkerton outbreak [32, 33] was conducted as a research protocol with human research ethics board approval from the University of Ottawa Heart Institute, the South Bruce Grey Health Center and the Owen Sound Hospital. Our investigating group (the "ECADS collaborators") was led by Dr. Rick Davies of the Ottawa Heart Institute. With the assistance of the local hospitals, hospital corporations and Public Health Unit, we accessed electronic health records data (including free-text chief complaint, age, gender, and demographic data) from 392,699 ER visits made to 10 hospitals in the Grey-Bruce Region of Ontario from January 1, 1999 until December 31, 2001. Five of the 10 hospitals brought their electronic systems online during 1999 and could only provide data for part of that year; data were complete for all 10 hospitals for 2000 and 2001.

Free-text chief complaints were categorized into syndromes of interest using a version of the RODS (Real-time Outbreak and Disease Surveillance) system [145], provided to us by the CNPHI (Public Health Agency of Canada) and QUESST (Ministry of Health and Long Term Care of Ontario) projects. A processed data set containing 1) the categorized chief complaint, 2) the hospital visited, 3) town of residence, 4) gender and 5) age group was used for subsequent analyses, which were done at the University of Ottawa Heart Institute and at the Auton Laboratory, Carnegie Mellon University.

Table 6.1: Results of spatial scan using the SSS software. SSS was able to detect the Walkerton outbreak on May 19.

Date	Most significant cluster	Score	False positive rate
May 16	11 cases, not near Walkerton	1.28	48.5%
May 17	7 cases, in and near Walkerton	0.38	79.9%
May 18	3 cases, not near Walkerton	2.87	20.4%
May 19	15 cases, in Walkerton	15.1	0.1%
May 20	33 cases, in Walkerton	42.1	0%
May 21	45 cases, in Walkerton	58.5	0%

While our complete study [33] describes a wide variety of methods used to detect and characterize the Walkerton outbreak, we focus here on automatic detection using our SSS system, and compare this system to several other methods, UNALERT (univariate time series analysis) and WSARE (What's Strange About Recent Events) [159, 160, 161]. For each method, we wish to measure the relationship between timeliness of detection (which day the Walkerton outbreak could have been detected) and false positive rate (the proportion of false alarms in the three years of baseline data). For a given day of the outbreak, the false positive rate is defined as the percentage of non-outbreak days that were found to be more significant than that outbreak day. In other words, this is the percentage of false positives we would have had to accept in order to have detected the Walkerton outbreak on the given day.

Thus we used the SSS software to conduct a spatial scan focusing on cases classified as GI for the entire Grey-Bruce region, for each of the five days before the boil water advisory. In Table 6.1, we present the results of this scan. For each day, we indicate the location of the most significant cluster detected, its score $F(S)$, and the resulting false positive rate needed for detection. We also show screen shots of the SSS software runs for May 19 and May 21 in Figure 6.9. Our results demonstrate that the spatial scan would have identified an abnormality in Walkerton on May 19 (two days before the boil water advisory) with a false positive rate of 0.1% (one false positive in the three years of data). We also note that the cluster in Walkerton on May 17 may be a preliminary indicator of the outbreak, but it was not significant enough to trigger an alert, unless we were willing to accept a false positive every 1.25 days. However, if we had access to other data sources (such as over-the-counter drug sales), it may have been possible to automatically detect the Walkerton outbreak on the 17th.

We compared the SSS detection results to two other methods: univariate time series analysis (using a standard control chart) and What's Strange About Recent Events (WSARE). Our results indicate that, if we performed univariate analysis on the time series of total GI visits for all Grey-Bruce hospitals, we would not have been able to detect the Walkerton outbreak until May 20. Thus SSS was able to detect a full day faster than univariate analysis, demonstrating that the spatial scan can achieve more timely detection as well as pinpointing the location of an outbreak. Similarly, WSARE was able to detect the Walkerton outbreak on May 20 with a false positive rate of 1.4%. Comparing WSARE to SSS, we note that SSS was able to detect the Walkerton outbreak one day earlier than WSARE. The tradeoff, of course, is that WSARE is a more general detector, and can detect a wider range of outbreak types and other anomalous patterns. For example, the anomalous pattern found by WSARE on May 20 was, "Normally 0.2% of all records are GI syndromes from

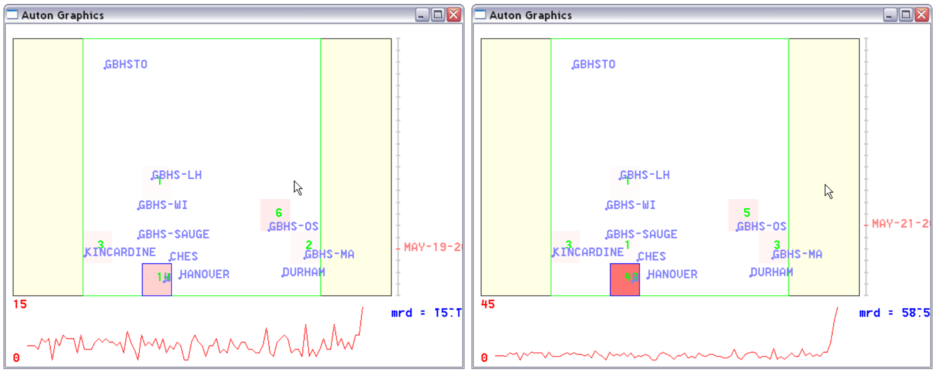


Figure 6.9: Results of running SSS on the Walkerton gastroenteritis outbreak. The left figure shows the most significant cluster on May 19, consisting of 15 cases in Walkerton. The right figure shows the most significant cluster on May 21, consisting of 45 cases in Walkerton.

Walkerton, but recently 5.8% of all records are GI syndromes from Walkerton." This is a clear indication that WSARE found a pattern that was directly relevant to the Walkerton GI outbreak, specifying both the outbreak type and approximate location.

In conclusion, the Walkerton outbreak ultimately resulted in six deaths and caused thousands to become ill. Our study [33] shows that surveillance of emergency room chief complaints would have provided important information regarding this outbreak, and might have advanced its detection by as much as several days. An additional benefit is provided by the automatic detection and characterization of outbreaks by tools such as SSS, reducing the burden of this surveillance on public health. We demonstrated that SSS was able to automatically detect the Walkerton outbreak on May 19, two days before the boil water advisory. Though the outbreak was actually detected on the 19th by an astute physician, we note several important observations. First, SSS was able to detect on the 19th with only one false positive in the three years of data, while false positives due to physicians reporting unusual symptoms are much more common. Second, the physician's detection relied on much more specific information (multiple cases of pediatric bloody stools) which was unavailable to SSS; if the same physician had not examined all of these cases, or if they had not been sufficient to trigger his suspicion, the outbreak would not have been detected as quickly. Finally, it is likely that incorporation of other data sources (such as over-the-counter drug sales) would have enabled us to automatically detect the outbreak several days earlier.

6.7 Related work in biosurveillance

As noted above, our work is most closely related to the spatial and space-time scan statistics approaches of Kulldorff [78, 81]. These methods were discussed in detail, and compared to our new approaches, in the previous chapters; here we describe other methods for spatial and syndromic surveillance. Unlike the approaches based on scan statistics, which both detect clusters and pinpoint their spatial location, the other spatial methods in the literature either do not find specific clusters, or do not evaluate the statistical significance of discovered clusters. More general overviews of the literature on spatial and syndromic biosurveillance can be found in the books by Lawson et al. [91, 95], Elliott et al. [44], and Wagner et al. [147]. In addition to the spatial cluster detection methods discussed above, these methods include general and focused clustering methods, disease mapping approaches, and spatial cluster modeling, as well as a variety of non-spatial methods. We discuss each of these areas in the following subsections.

We note that, with a few exceptions such as the Bayesian mixture modeling techniques of Lawson and Clark [90, 93] and Gangnon and Clayton [51], none of these methods attempt to model the locations and spatial extents of clusters, nor are any judgments made (i.e. by statistical significance testing) as to whether the resulting variations in risk are due to chance. Nevertheless, the wider literature on disease mapping and modeling has several advantages over scan statistics, including the ability to directly model correlations and both fixed and random effects.

6.7.1 General clustering and space-time interaction

General clustering methods are hypothesis testing methods which test for a general tendency of the data to cluster. In other words, these methods attempt to answer the question, “Is this data set more spatially clustered than we would expect?” Such methods do not identify specific clusters, but instead give a single result of “spatially clustered” or “not spatially clustered.” These methods are useful if we want to know whether anything unexpected is going on, but do not care about the specific locations of unexpected events. Examples of such methods include Whittemore et al. [157], who use the mean distance between all pairs of cases as a test statistic, and Bonetti and Pagano [19], who use the interpoint distance distribution (M-statistic) to measure the amount of clustering. Tango [143] proposes a quadratic form test, comparing the numbers of observed and expected counts weighted by a covariance matrix. Several other methods test for general clustering in data with non-uniform populations, by combining case information with information about unaffected individuals (controls) drawn from the underlying population. These methods typically ask the question, “are cases closer to other cases than they are to non-cases”? Cuzick and Edwards [31] consider several test statistics based on the k nearest neighbors of each case. Anderson and Titterington [8] propose an “integrated squared distance” statistic based on non-parametric density estimation: the test statistic is the squared difference between density estimates for cases and controls, integrated over the test region. Similarly, Diggle and Chetwynd [36] compare the second moments of the case and control distributions.

Closely related to the tests of “general” clustering are tests for space-time interaction. These tests answer the question of whether there is space-time clustering of events, even after adjusting for purely spatial and purely temporal clustering. In other words, space-time interaction is present when points that are close together in time also tend to be close together in space, and vice-versa. The two best-known tests for space-time interactions are the Knox [77] and Mantel [101] tests. The Knox test requires specification of threshold values for “closeness” in time and space (i.e. two points are close together in space if their spatial distance is less than d_s, and close together in time if their temporal distance is less than d_t). Then spatial and temporal distances are computed for each pair of points, and if many points that are “close” in time are also “close” in space and vice-versa then space-time interaction is present. Mantel’s test is a generalization of the Knox test which computes the product of functions of the temporal and spatial distances for each pair of points, and uses the sum of these products as a test statistic. Baker [10] extends the Knox test to cases where the values of the critical parameters that define “closeness” in space and time are unknown. Kulldorff [79] proposes an extension of the Knox test which adjusts for shifts over time in the distribution of the underlying population of individuals. Finally, Rogerson [129] combines a “local” variant of the

Knox test with cumulative sum methods in order to detect "emerging" space-time interaction by prospective surveillance.

6.7.2 Focused tests for detection of increased risk near a prespecified source

Focused clustering methods are hypothesis testing methods which, given a prespecified spatial location, attempt to answer the question, "Is there an increase in risk in areas near this location?" These methods can be used to examine potential environmental hazards, for example, testing for an increased risk of lung cancer near a coal-burning power plant. Since the locations are specified in advance, these methods are primarily used to test locations that have been identified by other means. This is different than spatial scan methods, which are used to discover and pinpoint significant clusters without a priori knowledge of their locations. Reviews of the literature on focused clustering include Hills and Alexander [67] and Bithell [18], as well as the chapters in Lawson et al. [91] and Elliott et al. [44].

Many tests for focused clustering have been proposed, with different assumptions about the distribution of relative risk near the focus. Besag and Newell [15] test for "hot-spot clustering," assuming a constant increase in relative risk for locations within some distance of the focus, and using the population radius containing a given number of cases as their test statistic. The tests of Waller et al. [152] and Lawson [89] assume that relative risk varies as $1 + \frac{k}{d}$, where k is constant and d is the distance from the focus. For example, Waller's test statistic is the sum of the deviations of observed incidence in each location from its expectation under the null, weighted by the amount of exposure. Diggle [35] instead assumes that relative risk decreases exponentially with distance to the focus, and proposes a test for case/control data based on this assumption. Finally, Stone [142] proposes a non-parametric test which assumes that relative risk is monotonically decreasing with distance to the focus.

6.7.3 Disease mapping

Disease mapping approaches have the goal of producing a spatially smoothed map of the variation in disease risk. For example, a very simple disease mapping approach might plot the observed disease rate (number of observed cases per unit population) in each area; more advanced approaches use a variety of Bayesian models and other spatial smoothing techniques to estimate the underlying risk of disease in each area. These methods do not explicitly identify cluster locations, but disease clusters may be inferred manually by identifying high-risk areas on the resulting map. Nevertheless, no hypothesis testing is typically done, so we cannot draw statistical conclusions as to whether these high risk areas have resulted from true disease clusters or from chance fluctuations.[1] Disease mapping is discussed in detail in Lawson et al. [92, 91] and Elliott et al. [44].

Breslow and Day [21] consider various methods of smoothing data for disease mapping, including kernel smoothing and kernel density estimation. Their methods do not assume any underlying model of the data, and are most useful for exploratory data analysis. Other disease mapping approaches are model-based, enabling statistical analysis and testing for areas of significantly increased risk. A variety of hierarchical models have been proposed: the top level of the hierarchy typically assumes that counts are Poisson distributed with mean proportional to a known expected count

[1] More precisely, while Bayesian disease mapping methods can produce the posterior probability of elevated risk at each individual spatial location, they cannot draw statistical conclusions about the cluster as a whole.

multiplied by an unknown relative risk. The second level of the hierarchy may assume that risks are drawn from a Gamma distribution, as in Clayton and Kaldor [28] and Mollié [104], who use empirical Bayes methods to estimate these risks. Alternatively, the log relative risk can be modeled using a Gaussian distribution, as in Waller et al. [151] and Mollié [105]. This latter representation has the advantage of being able to represent both fixed and random effects: for example, the log relative risk θ_i can be modeled as a vector of random effects with $\theta_i \sim \text{Gaussian}(\mu, \sigma)$ and non-informative hyperpriors. These models can be simulated using Markov Chain Monte Carlo (MCMC) methods, sampling from the posterior probability distribution, as in Besag et al. [16] and Clayton and Bernardinelli [29]. The disadvantage of these "fully Bayesian" approaches is that MCMC may be slow to converge to the true posterior, and it may be hard to confirm when convergence has occurred. An alternative would be to estimate the unknown vector of hyperparameters using maximum likelihood: such "empirical Bayesian" approximations may produce reasonable estimates of the relative risks, but may fail to account for model uncertainty. In any case, the model-based disease mapping approaches enable us to account for overdispersion of cases and other sources of heterogeneity. Spatial correlation can also be included by incorporating an additional, spatially structured random effect term. An alternative method of accounting for correlations is given by Wolpert and Ickstadt [73, 158], who propose a hierarchical random field model with spatially correlated counts.

6.7.4 Spatial cluster modeling

Spatial cluster modeling methods attempt to combine the benefits of disease mapping and spatial cluster detection, by constructing a probabilistic model in which the underlying clusters of disease are explicitly represented. A typical approach is to assume that cases are generated by some underlying process model which depends on a set of cluster centers, where the number and locations of cluster centers are unknown. Then we attempt to simultaneously infer all the parameters of the model, including the cluster centers and the disease risks in each area. Thus precise cluster locations are inferred, and while no formal significance testing is done, the method is able to compare models with different numbers of cluster centers, giving an indication of both whether there are any clusters and where each cluster is located. One typical disadvantage of such methods is computational: the underlying models rarely have closed-form solutions, and the Markov Chain Monte Carlo methods used to approximate the model parameters are often computationally intensive. Examples of such methods include Lawson et al. [90, 93] and Gangnon and Clayton [51]. For a more detailed discussion of spatial cluster modeling, see Lawson and Denison [94].

In the Lawson and Clark model [93], the intensity of disease cases is expressed as a product of the overall disease rate, the population at risk, and a spatially-varying relative risk function. This relative risk function is parameterized in terms of an unknown number of clusters κ, a corresponding set of κ cluster locations, and further parameters corresponding to the risk decay around clusters. All of the model parameters are inferred simultaneously: since the number of components is unknown, reversible jump Markov Chain Monte Carlo sampling is used [61]. The model can also be adapted to aggregated data by setting the expected count in a region equal to the integral over that area of the intensity function, and covariates and random effects can also be included.

A second type of spatial cluster modeling approach is based on mixture models; examples of such approaches include Schlattmann and Böhning [134] and Richardson and Green [128]. In these models, the dataset is assumed to consist of cases, each of which is drawn from one of an unknown number of mixture components. These methods enable explicit modeling of population heterogeneity, as different mixture components may have very different properties. Both the number of components, and the properties of each component, are inferred using either empirical Bayes or Markov Chain Monte Carlo.

Finally, Gangnon and Clayton consider a range of methods which span the space from scan statistics [53, 52] to Bayesian cluster modeling [51, 54]. In [53, 52], they consider three methods: a weighted scan statistic, the weighted average of likelihood ratios (WALR), and the maximum weighted average of likelihood ratios containing a given cell (WALRS). All three of these methods can be thought of as approximations to our Bayesian spatial scan statistic: the weighted scan statistic is a maximum a-posteriori estimate of the most probable cluster, the WALR statistic is an approximation of the total posterior probability of an outbreak, and the WALRS statistic is an approximation of the total posterior probability of an outbreak containing a given cell. We believe that the use of the full Bayesian model, and the exact calculation of posterior probabilities given the model, is preferable to any of these approximations. In later work, Gangnon and Clayton also consider a hierarchical Bayesian model [51, 54], but use a different prior distribution (based on Markov Connected Component Fields), and the result is a model which cannot be computed efficiently but only approximated by simulation.

6.7.5 Non-spatial surveillance methods

Syndromic surveillance includes a wide variety of methods for early detection of disease epidemics. Excellent surveys of the literature on syndromic surveillance are given in the *Handbook of Biosurveillance* [147], as well as the review papers by Wagner et al. [149], Mandl et al. [100], and Buckeridge et al. [22]. Here we briefly consider some of the many methods for syndromic surveillance that do not explicitly take spatial data into account. Many of these methods are based on time series analysis; we consider here both univariate and multivariate methods.

Univariate temporal methods consider a single time series, and signal alerts when the current count is significantly higher than its historical expectation. The simplest such method is the Shewhart chart [137], which alerts whenever the current count is more than some number of standard deviations from its mean. Other variants of the control chart track the smoothed count, using either a moving average (MA) or exponentially weighted moving average (EWMA) [71]. Cumulative sum (CUSUM) methods [123, 72] consider the accumulated deviation from the mean over multiple time steps, and signal alerts when the accumulated deviation is sufficiently large. These methods have some useful theoretical properties, including bounds on their frequency of alerts under the null. Many other regression models can be used for forecasting the current counts. These methods include the Generalized Linear Mixed Models (GLMM) approach of Kleinman et al. [76] and the cyclical regression model of Serfling [136]. ARIMA and other standard time series analysis methods [63], as well as newer approaches based on wavelet decomposition [168], can also be used for forecasting. Le Strat and Carrat [96] used Hidden Markov Models (HMMs) to monitor influenza-like illnesses and poliomyelitis. Similarly, Kalman filters can be used for temporal outbreak detection [65]. One advantage of HMMs and Kalman filters over other temporal methods is that they allow explicit representation and modeling of disease state (i.e. which diseases, if any, are occurring).

In many time series monitoring domains, multivariate signals have been considered. The simplest (and most common) multivariate equivalent of a control chart is the Hotelling T^2 method. This method learns the joint distribution on a set of signals from historical data, then alerts if the current multivariate signal is sufficiently far from its expectation. This allows us to detect when any of the individual signals, or the relationship between these signals, deviate significantly. Many other multivariate methods have been developed, including multivariate versions of the EWMA and CUSUM methods. These are all very general and useful methods for determining whether any of several data streams deviate from normal conditions; however, unlike the spatial scan statistic, none of these methods are able to provide spatial information about the size, shape, and location of potential outbreaks. A detailed description and comparison of multivariate methods is given by Burkom et al. [25, 26, 27].

Another class of methods for multivariate biosurveillance uses association rule mining to search for unusually frequent patterns in public health data (such as over-the-counter drug sales). For example, DuMouchel [42] developed empirical Bayes screening to search for frequent multi-item associations. This method can be used to detect unusual patterns of medication sales indicative of an epidemic, for example, an increase in the number of patients who bought both anti-diarrheal and fever medications.

Finally, Bayesian networks are another useful tool for multivariate biosurveillance, as they allow efficient representation and inference of the relationships between large numbers of variables. Bayesian networks have been used within biosurveillance in a number of contexts, including What's Strange About Recent Events [160, 161], which searches for anomalous patterns in recent records using a Bayesian network to infer the background model, and PANDA [30], which builds a huge network representing every person residing in a city and uses this network to infer whether an anthrax attack has occurred.

Chapter 7

Application to brain imaging

7.1 Introduction

In this chapter, we apply our cluster detection methods to the analysis of brain imaging data. We focus here on *functional imaging*, which measures brain activity, and thus our task is to detect clusters which correspond to regions of increased or decreased brain activity. For example, we might want to detect regions of the brain that have been damaged by strokes or by a variety of neurodegenerative diseases, including Alzheimer's and Parkinson's. Here we focus on an application in cognitive neuroscience, in which the main goal is to detect clusters of brain activity that allow us to differentiate between cognitive states. Thus we want to distinguish between subjects performing different cognitive tasks (for example, reading a book versus watching a movie), and to determine which areas of the brain are most active in performing each task. In this cluster detection task, we analyze data produced by functional magnetic resonance imaging (fMRI). An fMRI scanner measures the changes in blood oxygenation resulting from neural activity in a subject's brain: three-dimensional "snapshots" of the subject's brain activity are taken at regular intervals (typically every 1-3 seconds) while the subject is performing a cognitive task or receiving some stimulus inside the scanner [165, 163]. Worsley [163] notes that the fMRI response to a stimulus is delayed and dispersed by about six seconds, which can be modeled by convolution of the stimulus pattern with a hemodynamic response function consisting of a sharp peak followed by a slight drop below the normal activation level. Thus we can detect clusters of brain activity by finding regions of the brain that show significantly increased activation in response to the presentation of the stimulus.

This task can be mapped directly into our general statistical framework for cluster detection: at each time step t, we have a three-dimensional grid consisting of $64 \times 64 \times 14$ voxels s_i, where the measured "activation" of each voxel corresponds to the amount of activity in that region of the brain.[1] We can use this activation directly as the count c_i^t corresponding to spatial location s_i at time step t, or we can first pre-process the data by normalizing and smoothing. Thus one possibility would be to use the entire sequence of fMRI images produced for a given subject and stimulus, inferring the baseline level of activation for each voxel from historical data (i.e. the previous counts c_i^t), and detecting regions where the voxels have increased activation. In our results below, we consider an even simpler version of the fMRI cluster detection task, where we compare a single

[1] As noted above, $64 \times 64 \times 14$ was the maximum spatial resolution of fMRI images available for our experiments. Other fMRI images may have $128 \times 128 \times 14$ or higher resolutions.

brain image taken from the subject after receiving the experimental stimulus (corresponding to the peak of the subject's hemodynamic response) to another image taken after the subject receives a "control" stimulus (again corresponding to the peak of response). Thus we can use the measured activation of each voxel in the "experimental" image as our set of counts c_i, and the measured activation of each voxel in the "control" image as our set of baselines b_i. Then we can detect clusters of brain activity that are activated by the experimental stimulus by finding regions $S = \{s_i\}$ where the counts c_i are significantly higher than expected, given the baselines b_i.

As in the disease surveillance domain, a variety of confounding factors make it difficult to detect clusters in brain images. One difficulty is temporal variation in the measured activation due to the phenomenon of fMRI drift [140], which can be caused by instabilities in the fMRI scanner or changing physiological responses of subjects. A second difficulty is the challenge of modeling the non-linear hemodynamic response function [165]. Third, while Gaussian distributions are often used to model fMRI activation, our results suggest that these distributions are not normally distributed, often having lower kurtosis (lighter tails) than would be expected from a Gaussian distribution. Finally, huge challenges are presented by differences across subjects. As Wang et al. [156] note, different subjects' brains have substantially different sizes and shapes, and different subjects may generate different spatial patterns of brain activation given the same cognitive state. Furthermore, an individual subject's response may differ between trials based on physiological state and mental distraction, and in some cases we may detect no signal because the subject did not attend to the stimulus. Solutions to all of these domain-specific challenges are beyond the scope of this thesis, though many of them have been considered in the related work discussed below. We believe that many of these challenges can be addressed within our cluster detection framework, as in the disease surveillance domain, by a combination of pre-processing (to deal with irregularities in the data), post-processing (to filter out irrelevant regions), and the iterative development of more complex models which are appropriate to the brain imaging domain.

In the remainder of this chapter, I consider the application of our cluster detection methods to finding clusters of increased brain activity in fMRI data. The main purpose of this brief exposition is to demonstrate the applicability of our methods to brain imaging: our fast multidimensional spatial scan makes these methods computationally feasible, and the flexibility of our statistical framework enables us to detect useful and relevant clusters of brain activity. In Section 7.2, I present our preliminary results in the brain imaging domain, reviewing the speed results discussed in Chapter 3 and considering the quality of the clusters found. Finally, in Section 7.3, I present an overview of the fMRI brain imaging literature, focusing on cluster detection. Parts of this chapter have been adapted from our paper in NIPS 2004 [118]. I wish to thank my co-authors Andrew Moore, Francisco Pereira, and Tom Mitchell for their contributions to this work. Additionally, I am extremely grateful to Tom Mitchell and his group for making their fMRI brain imaging data available to us.

7.2 Results

Our first results in the brain imaging domain were presented in Chapter 3, where we demonstrated that our fast multidimensional spatial scan algorithm was able to detect clusters 7-148x faster than the naïve spatial scan approach. This speedup makes it computationally feasible to perform cluster detection using the spatial scan on brain imaging data, reducing the run time from weeks to hours.

We now consider the regions found by our scans, and whether these correspond to useful and relevant clusters of brain activity. As noted above, the purely spatial fMRI cluster detection task

Table 7.1: Clusters of brain activity detected in fMRI data, "word versus baseline" task. All detected clusters were found to be significant ($p < .05$) using the discriminative thresholded scan statistic with $\epsilon = .01$.

subject	cluster coordinates	area of brain
08170	(20-24, 34-43, 13)	visual cortex
08179	(16-17, 40-42, 12-13)	visual cortex
08179	(24, 56, 10)	unknown, possibly noise
08179	(11-14, 22-25, 12-13)	Broca's area
08179	(44-45, 22-23, 11-12)	Broca's area (opposite side)
08179	(40-42, 28-30, 13)	Wernicke's area

consists of a three-dimensional grid of voxels. For each voxel, we have a count c_i and a baseline b_i, where c_i corresponds to the measured amount of fMRI activation in that voxel under the experimental condition, and b_i corresponds to the measured amount of fMRI activation in that voxel under the null or control condition. We used data from an experiment by Mitchell et al. [103] where the subject is given words, one at a time, and must read these words and identify them as verbs or nouns. We considered two tests: the easier "word versus baseline" task, where the goal was to distinguish subjects reading a word from the baseline condition of that subject fixating on a cursor, and the harder "noun versus verb" task, where the goal was to distinguish whether the subject was reading a noun or a verb. In each case, we detected significant clusters of increased brain activity using the discriminative thresholded scan statistic discussed in Chapter 2, with values of the detection threshold ϵ ranging from 0 to 0.05. As noted in Chapter 3, the classical scan statistic ($\epsilon = 0$) was unable to find relevant clusters, instead detecting large regions (e.g. $\frac{1}{4}$ of the entire brain) that were only slightly increased in count. When we used a larger threshold value, our statistics detected smaller regions with more substantial increases, and some of these regions may correspond to relevant clusters of brain activity.

We first attempted the more difficult "noun versus verb" task, searching for clusters with more than 1% increase in activation ($\epsilon = .01$) in data from six different subjects. Our results were inconsistent, with no regions found in four of the six subjects. While the other two subjects did have significant regions of activity, our domain expert was unable to identify these as corresponding to relevant areas of the brain. Thus we considered the simpler "word versus baseline" task, focusing on these two subjects. In this task, we were able to find relevant regions, as identified by our domain expert; a list of these regions and the corresponding functional areas of the brain is shown in Table 7.1. For both subjects, we detected significant clusters of activity in the visual cortex. For one subject, this was the only significant region detected, while the other subject also had significant clusters in the language centers of the brain (Broca's and Wernicke's areas). These clusters make sense given the nature of the experimental task; however, more data is needed before we can draw conclusive cross-subject comparisons.

While a detailed consideration of the brain imaging domain is beyond the scope of this thesis, future work will address a number of aspects of this domain. Our main goal will be to improve the detection power of our methods to obtain consistent cross-subject results, both for simple tasks such as the "word versus baseline" task, and for more challenging tasks such as the "noun versus verb" task. First, we plan to compare our current, purely spatial approach to the space-time approach discussed above: we believe that using data from the entire sequence of fMRI images rather than only a single image will increase detection power, as will the use of our expectation-based scan statistics. Second, we must derive models and statistics that are most appropriate for this application domain, perhaps using a thresholded Gaussian statistic which accounts for spatial correlations between adjacent voxels. Further performance gains may be achieved by incorporating the hemodynamic response function directly into our models (using a parametrized space-time scan statistic as discussed in Chapter 4), as well as accounting for confounding factors such as fMRI drift.

7.3 Related work in brain imaging

We now provide a brief overview of the literature on cluster detection in fMRI data. As noted above, two of the main goals of cluster detection in the brain imaging domain are to differentiate between subjects performing different cognitive tasks, and to find which regions of the brain are most active in performing each task. It is also possible to differentiate between cognitive states without performing cluster detection, though these methods will not pinpoint the relevant regions of the brain. For example, Mitchell et al. [103, 156] have considered a general, classifier-based approach for distinguishing cognitive states. These approaches train a Gaussian Naïve Bayes classifier to predict the subject's cognitive state given their observed fMRI data, and are able to achieve high accuracies for tasks including "reading a sentence versus viewing a picture," "reading ambiguous versus non-ambiguous sentences," and "reading words in different semantic categories." In [156], classifiers were trained across multiple subjects, enabling accurate generalization across subjects in these discrimination tasks. While these are impressive results, we focus here on methods which will identify significant clusters of brain activity rather than simply performing classification of fMRI images, thus revealing which areas of the brain are indicative of the differences between cognitive states. We do not attempt to perform cross-subject generalization, but we believe that this is an important area for future work.

As noted by Perone Pacifico et al. [126], a common approach to identifying activated voxels in fMRI images is to perform a separate hypothesis test for each voxel (typically after applying some method of spatial smoothing) and to report all voxels that are significant at some level α. A variety of such tests have been proposed, ranging from Kolmogorov-Smirnov tests [5] to nonlinear regression in a Bayesian framework [56]. One of the most common methods for finding clusters of activity, known as "statistical parametric mapping" [50], uses generalized linear models to predict the activity of each voxel given the stimulus. In all of these cases, because separate statistical tests are being performed on thousands of voxels, some adjustment for multiple hypothesis testing is necessary, for example the use of permutation tests [11] or random field theory (as discussed below) to estimate the null distribution of the maximum value of the test statistic. Because statistical significance is tested on a per-voxel basis, clusters of brain activity must be inferred by grouping individually significant voxels, and no per-cluster false positive rate is guaranteed. One recent exception to this statement is the work of Perone Pacifico et al. [126], who use a random field approach to bound the proportion of false clusters discovered, and apply their method to the traditional scan statistic (maximizing the number of points in a two-dimensional window) and to analysis of fMRI images. Nevertheless, their results cannot be used for the more general spatial scan statistic approaches that we consider here, including detection of clusters with variable size and shape, as well as the use of flexible, model-based score functions.

Another well-known method of detecting clusters in fMRI data was originated by Worsley et al. [164, 166, 163]. This method, first used for analysis of positron emission tomography (PET) data in [164] and generalized to other types of brain imaging data in [166], detects clusters by first smoothing the data to make it approximately Gaussian, allowing simple t or F test statistics to be used. To account for spatial correlation between voxels, a ***random field*** approach is used [163]: we first compute the test statistic for each voxel, forming an image $T(s)$ of test statistics for the activation, then choose a threshold t, and declare as activated all points where $T(s) > t$. The threshold t can be computed by using the Euler characteristic of random fields, which approximates the p-value of the global maximum $T(s)$. The advantage of this approach is that a simple exact expression has been obtained for the expected Euler characteristic when no activation is present: this distribution was calculated for Gaussian random fields by Adler [2] and for many other types of random field by Worsley [162]. As a result, we can directly obtain the p-value of the maximum value of the test statistic without performing randomization testing.

Random field approaches have both advantages and disadvantages as compared to spatial scanning: like our Bayesian spatial scan, no randomization testing is necessary, and thus rapid detection of significant clusters can be performed. Additionally, the random field approaches explicitly account for the correlation structure of the data. However, because only single voxels are tested, these approaches cannot bound the per-cluster false positive rate. Additionally, the assumption of a fixed correlation structure limits detection to compact clusters of a given size (based on the smoothing bandwidth) and a fixed shape. Siegmund and Worsley [139] extend the random field method to signals of unknown width by maximizing a Gaussian random field in $N+1$ dimensions (N dimensions for the location plus one dimension for the width). Nevertheless, this method still cannot search over varying cluster shapes as in the spatial scan approach. Finally, because of the need to compute the distribution of the Euler characteristic, random field methods cannot be used for general score functions $F(S)$. On the other hand, our generalized spatial scan framework enables us to efficiently compute the most significant regions (and their p-values) for a wide range of score functions, thus giving us the flexibility to choose those models and statistics which are most appropriate for a given application domain.

Chapter 8

Conclusions and future work

8.1 Introduction

In this thesis, I have presented a variety of methods for accurate and computationally efficient cluster detection in diverse application domains. Our methods improve on the previous state of the art in several aspects. First, they exhibit higher accuracy and better ability to detect relevant clusters while excluding irrelevant clusters (Chapter 4). Second, they are much more computationally efficient, typically achieving two to three orders of magnitude speedup (Chapters 3 and 5). Third, our generalized framework (Chapter 2) expands the scope of applications to which cluster detection techniques can be applied, and fourth, we can incorporate information such as prior knowledge and multiple data streams to further improve detection power (Chapter 5).

As discussed in Chapter 1, the spatial scan is a powerful statistical method with high potential utility for a variety of application domains. However, the usability of this method had been restricted in practice by both computational intractability (making it infeasible for use on massive real-world datasets) and a lack of modeling flexibility (making it unable to distinguish relevant from irrelevant clusters in real-world application). In this thesis, I have proposed a variety of techniques which make spatial scanning both practical and useful for massive datasets in real application domains.

Using the generalized spatial scan framework we developed in Chapter 2, we considered several ways of making scan statistics more accurate and thus more useful in practice. Our expectation-based scan statistic approach (discussed in Chapters 2 and 4) enables us to account for the spatial and temporal variation in the underlying model, by learning expected counts from the time series of previous data then finding spatial clusters with higher than expected counts. Further improvements were gained by extending this model to a space-time scan statistic, and developing new statistics for the detection of emerging and persistent clusters, as discussed in Chapter 4. The generality of our framework also allows us to create new, computationally efficient statistics that can account for other aspects of the domain model: for example, Gaussian scan statistics that can account for overdispersion of counts, or thresholded scan statistics that can avoid detecting statistically but not practically significant clusters. As discussed in Chapter 6, many other aspects of an application domain can be dealt with either by preprocessing (e.g. to deal with missing data) or postprocessing (e.g. filtering out irrelevant regions), further increasing our detection power. Finally, our Bayesian scan statistics (discussed in Chapter 5) can incorporate prior information about the size, shape, and impact of clusters, leading both to higher detection power and more easily interpretable results.

Within our generalized spatial scan framework, we considered two ways of making the spatial scan computationally feasible and very fast for applications involving very large datasets (such as monitoring nationwide public health data). Our “fast spatial scan” algorithm, presented in Chapter 3, accelerates the spatial scan between 100-1000x without any loss of accuracy (i.e. it finds exactly the same clusters and p-values as a naïve scan). This enables us to run the algorithm in under 20 minutes, instead of taking days, on nationwide data. In the public health domain, we must discover and report emerging outbreaks of disease as quickly as possible, so this speedup is essential for the practical utility of spatial scan for outbreak detection. An alternative method of speeding up the spatial scan is given in Chapter 5: our Bayesian cluster detection method eliminates the need for randomization testing, thus enabling a 1000x speedup over the naïve frequentist approach.

In addition to advancing the state of the art in cluster detection methods, we also focused on applying these methods to detect useful and relevant clusters in several application domains. Chapters 4-6 applied our cluster detection techniques to the early detection of disease outbreaks from public health data, such as emergency department records and over-the-counter drug sales data. We demonstrated high detection power for both semi-synthetic tests (synthetic outbreaks injected into real baseline data) and retrospective analysis of known disease outbreaks. Chapter 6 also described how we put these techniques into real-world practice: our SSS tool is currently performing nationwide disease surveillance on a daily basis, and has already demonstrated its ability to detect useful and relevant clusters in practice. Chapter 7 applied our techniques to fMRI brain imaging data, with the goal of finding clusters of brain activity that can distinguish between different cognitive states. While this work is still in the early stages, I presented some simple “proof of concept” results demonstrating that our methods can rapidly detect relevant clusters of brain activity. In addition to continuing our work in these two application domains, we are also in the process of extending and applying these methods for many other domains. Some of these applications include:

- Detection of terrorist groups, using the Bayesian spatial scan statistic to combine probabilistic link and group data with our analysis of individual suspects.
- Network intrusion detection, searching for patterns indicative of an attack.
- Detection of anomalous spatial patterns in container shipping data.
- Tumor detection from medical imaging data, including the detection of brain tumors from structural MRI data and breast cancer from mammography data.
- Combination of data from multiple noisy sensors, e.g. to analyze water quality data.

Many of these applications pose new and interesting challenges, including modeling of the relevant features of each domain, application of our methods to link and network data, and combination of multiple data streams. We are also extending this work in a number of other ways, and some of the most interesting extensions are discussed in the following section.

8.2 Future work

We now briefly consider six important directions for future work: extension of our methods to multiple data streams and multiple cluster models, real-time detection and investigation of clusters, incorporation of other types of data, detection of irregular clusters, tracking disease state over time, and automatic learning of relevant versus irrelevant clusters from user feedback. Each of these extensions has the potential to dramatically improve the generality and utility of our methods in real-world practice, as we discuss in the following subsections.

8.2.1 Extension to multiple data streams

As discussed in Chapter 5, one of the most important extensions of our method is the multivariate Bayesian scan statistic (MBSS), which allows us to combine information from multiple data streams in a principled Bayesian framework. In the disease surveillance domain, this will accomplish two main goals: increasing our power to detect outbreaks that affect multiple streams, and enabling us to differentiate between multiple potential causes of an outbreak. More generally, the MBSS framework allows us to consider many potential causes of an observed cluster, both relevant causes (such as a disease outbreak) and irrelevant causes (such as weather or promotional sales). Then by proposing a separate, scenario-based generative model for each type of cluster, we can distinguish between the different causes and decide which clusters are and are not relevant. As discussed below, we hope to eventually automate the learning of these models based on user feedback. The MBSS method outputs the joint posterior probability distribution over all possible regions and causes, given all streams of data; thus it not only identifies and pinpoints potential clusters but also explains them in terms of its causal models. We believe that the extension to multiple streams and multiple cluster models will make this method valuable for a wide variety of application domains. Implementation of MBSS is in progress, and we discuss this method in more detail in Chapter 5.

8.2.2 Real-time cluster detection and investigation

As discussed above, our "fast spatial scan" and "Bayesian spatial scan" methods each enable us to perform automatic cluster detection in under one hour (instead of days or weeks) for massive real-world datasets. It is likely that the fast spatial scan can be made even faster in future work, and we are considering several ways to accomplish this, including more aggressive forms of region pruning, simultaneous testing for multiple clusters and multiple parameter values, better use of cached statistics, and several detailed implementation issues. The fast spatial scan can also be efficiently parallelized, since each of the $R = 1000$ replications can be performed in parallel, and there are also multiple opportunities for parallel computation within each replication.

An even more exciting avenue for future work arises from the fact that the fast spatial scan and Bayesian scan achieve their speedups in very different ways: the fast spatial scan reduces the time per replication by 100-1000x, while the Bayesian scan eliminates the need for multiple replications (thus searching only a single grid instead of 1000). By combining these two methods as discussed in Chapter 5, it should be possible to create a method which searches only a single grid but does so using a fast search rather than a naïve search. We estimate that this combination should reduce the run time to under two minutes for national-level data. Even larger speedups should be possible by incorporating the new Time Series Aggregation Cube (TCUBE) technology under development by the Auton Laboratory [130]: by enabling much faster aggregation of time series data, this cached data structure should reduce scan time to a matter of seconds.

Our eventual goal is to be able to detect clusters *in real time* even for massive datasets. This will allow near-instantaneous notification of emerging clusters (e.g. disease epidemics), especially if data is made available incrementally and continuously streamed into the system. Moreover, users (such as public health officials) will be able to run large numbers of queries on a "point, click, and compute" basis, enabling them to rapidly obtain all the information they need to investigate, evaluate, and respond to detected clusters. As discussed in Chapter 6, Maheshkumar Sabhnani and I are working to combine our SSS cluster detection system with a viewer tool based on TCUBE, thus enabling ad hoc browsing and spatial analysis of data from multiple, multivariate time series. This will give the user the capability to rapidly and easily investigate the clusters found by SSS, as well as performing any other desired investigations. As I discuss further below, a longer-term goal is to automate much or all of the process of cluster detection and investigation, requiring only top-level guidance from human users.

8.2.3 Extension to other data types

Our current multidimensional spatial scan approach takes as input a set of counts c_i and baselines b_i associated with spatial locations s_i, where each s_i corresponds to a point in d-dimensional Euclidean space. This allows us to scan over records with any real-valued attributes, whether these are spatial attributes such as latitude and longitude, or other attributes such as time, age, height, and weight. Our scan returns hyper-rectangular regions corresponding to ranges of each attribute and containing all s_i within these ranges.

For computational efficiency, our current method discretizes each real-valued attribute uniformly into a user-specified number of buckets, thus creating a d-dimensional grid structure. This discretization makes it easy to handle ordered categorical values (e.g. age deciles, movie ratings) as well as real-valued attributes. However, the discretization of real-valued attributes risks losing detection power and spatial precision when it aggregates distinct spatial locations into a single grid cell. In some applications, such as brain imaging, the data we receive has already been aggregated to a grid structure. But in other applications, such as disease surveillance, we receive information about the exact coordinates of each data point and would like to maintain this level of precision. In this case, we can map points to a *non-uniform grid*, drawing grid lines corresponding to the distinct spatial coordinates of our data points in each dimension. An extreme version of this would be to have a grid of size $M_1 \times M_2 \times \ldots \times M_d$, where M_i is the number of distinct coordinates in dimension i. Since this would create grids of overly large size for most realistic datasets, we can "round" coordinates which are close to each other to the same value, allowing us to reduce the size of the grid as desired (with the tradeoff being decreased precision). One problem with this approach is that the resulting grid is likely to be very "sparse," containing many grid cells with low or even zero counts. Thus a search of all gridded regions (even one accelerated by the fast spatial scan techniques discussed above) may waste valuable time by searching regions which are spatially distinct but contain identical sets of data points. Our solution to this problem was to propose a *smart naïve* approach for iterating over all distinct axis-aligned rectangular regions. This technique uses an "incremental addition" algorithm somewhat similar to Kulldorff's method for searching over circles; it should be significantly faster than a naïve gridded approach when the grid is sparse, but since no pruning is done, it must actually search over all of the distinct rectangles. We are also working to extend our *fast spatial scan* pruning approach to non-aggregate data: our proposed approach is to use an overlap-kd tree with *local gridding* (creating a different resolution of non-uniform grid at each node of the tree), pruning regions when possible, and using the smart naïve approach on regions that cannot be pruned.

Thus we have considered how to search over multidimensional data with real-valued and ordered discrete-valued attributes. We now consider how to deal with unordered discrete-valued attributes:

the difficulty here is that we can no longer focus our search on ranges of the attribute, but must consider all of the exponentially many combinations of attribute values as distinct regions. One possibility would be to make some sort of simplifying assumption, e.g. assuming that a cluster contains no more than k distinct values of the attribute. It is also possible that, for attributes with relatively few values, that cached data structures such as TCUBE (discussed above) and AD-trees [106] will make it possible to search exhaustively over all sets of values. For all of these data types, we must think carefully about how to estimate the baselines or expected counts for each record or range of attributes. One possible approach may be to use Bayes Nets as an underlying probabilistic model, building on prior work by Schneider and Moore [135]. This may also allow integration of our work with the What's Strange About Recent Events (WSARE) methods of Wong et al. [159, 160, 161].

One eventual goal of our work is to develop general methods for finding those patterns in data which are most significant, relevant, or anomalous. We plan to achieve this by a generalization of cluster detection, in which we search for subsets of the data which are different from their expectation (or from the rest of the data) in some similar way, as if all of these records had been affected by some common (but possibly unknown) process. In a Bayesian framework, we can combine the prior probability that a process has affected a particular set of records with the data likelihood given this process and affected "group," and use machine learning methods to infer the type and parameters of the underlying process. This method extends cluster detection, in that the "group" is not limited to a spatial region (or spatially proximate set of points) but can represent any portion of the attribute space or any subset of records. Additionally, searching over groups has several advantages over the typical method of searching for individual "anomalous" points: we can detect subtler trends which would not be obvious from any single record, as well as finding sets of records which are not individually anomalous, but have interesting patterns of interaction.

8.2.4 Detection of irregular clusters

While our discussion above has focused on detecting either axis-aligned or rotated rectangular clusters, it may also be useful to extend our methods to the detection of irregularly shaped clusters. For example, given data for a set of zip codes, we might want to search over all regions S containing a connected set of zip codes, including both rectangular and non-rectangular regions. Scanning over irregular regions would give us higher power to detect clusters with areas that cannot be approximated by a (rotated) rectangle. Additionally, scanning over connected regions allows us to perform cluster detection in a general metric space (given only the connections and distances between adjacent points), rather than requiring the points to be embedded in some d-dimensional Euclidean space. This enables us to apply the spatial scan to link and network datasets, and many other datasets that are not "spatial" in the traditional Euclidean sense.

However, it is clear that the set of all connected regions will be very large for most practical applications, making it computationally infeasible to search over all such regions. Additionally, we might believe that some such regions are much more likely to be clusters than others: for example, we might believe that clusters are unlikely to be highly irregular in shape, or that they are likely to follow the shape of a river or highway. Thus any method of searching for irregular clusters must answer two questions: what subset of regions to search, and how to weight the likelihoods of these regions. In the frequentist setting, we can search for the regions that maximize a penalized likelihood ratio statistic, and calculate the statistical significance of these regions by randomization. For example, Duczmal et al. [41] assume that compact clusters are more likely than highly elongated clusters, and thus they use a penalized statistic that multiplies the likelihood ratio by a measure of the cluster's compactness. In the Bayesian setting, on the other hand, we can weight regions S by assigning them different prior probabilities $\Pr(H_1(S))$, ensuring that $\Pr(H_0) + \sum_S \Pr(H_1(S)) = 1$.

We now consider three possible ways of choosing which subset of regions to search: using natural features of the domain, scanning some set of "not-too-irregular" regions, or searching only those regions that are most likely to be potential clusters. We are currently investigating several variants of the spatial scan that use natural features to choose scan regions. As one example (joint work with Maheshkumar Sabhnani and Andrew Moore), we can run a spatial scan along a river or highway. This method enables us to detect outbreaks that are carried by water, by airborne release from a moving vehicle, or by human-to-human transmission along a transportation route. In this case, we scan over the set of regions $S = (x_1, x_2, d)$ consisting of all points within distance d of some point on the river between starting point x_1 and ending point x_2, where x_1, x_2, and d are all allowed to vary. Another possibility would be to group the zip codes by water pressure zone and use these as our set of scan regions; this method is being used in joint work with Vahan Grigoryan, Maheshkumar Sabhnani, and Mike Wagner.

Another option is to choose some subset of the connected regions consisting of clusters that are "not too irregular." These methods decide which regions to search based only on the spatial distribution of locations. For example, the "flexible scan statistic" of Tango and Takahashi [144] searches over connected regions S consisting of some location s_i and some (possibly empty) subset of its $K - 1$ nearest neighbors. This algorithm is only feasible for small to medium cluster sizes, as its run time is over one week for $K > 30$. We are investigating another method which first forms a hierarchical clustering of the spatial locations, then scans over all nodes in the resulting hierarchy of regions. This method is somewhat similar to Kulldorff's tree-based scan statistic [83], but learns the hierarchy from data rather than being provided with this hierarchy in advance. Because the number of regions searched scales only linearly with the number of locations, this method is very fast even for large datasets, but detection power will be substantially reduced for any clusters that cut across our partition of the space. To improve detection power, at the expense of increased computation cost, we can search connected regions consisting of pairs (or larger sets) of nodes that are nearby in the hierarchy.

The final class of methods for irregular cluster detection attempt to focus the search on those regions which are most likely to be clusters. As noted by Patil and Taillie [124], these methods take one of two approaches: to perform a heuristic search over the set of connected regions using some stochastic optimization method, or to use some preprocessing step to identify a subset of candidate regions to search. Duczmal et al. [38, 39] propose two heuristic search methods (using simulated annealing and genetic algorithms respectively) which enable them to find a connected region that approximately maximizes the penalized likelihood ratio statistic. Because heuristic methods are used, convergence to the most significant cluster is not guaranteed, and this may also affect the precision of the resulting p-value. Patil and Taillie [124, 125] instead perform an exhaustive search over a subset of the connected regions. This subset is computed by finding all distinct *upper level sets* of the graph, where an upper level set consists of all locations with rate higher than some threshold. They then search over the connected components of the upper level sets. We are currently investigating another method (joint work with Maheshkumar Sabhnani) which uses a discriminative random field (DRF) to identify potential clusters then searches only over these clusters. This method differs from the others because it relies heavily on the ability of the DRF to find relevant clusters; the resulting scan may be over a very small number of potential clusters or none at all.

We note that all of these “focused search” methods choose a different set of clusters to search for each dataset, and thus for each replica dataset if randomization testing is performed. This has two effects: first, because the search must be performed again for each replica, computationally expensive search methods will lead to very large run time. Second, any bias or variance in the search method will lead to imprecision in the calculated p-value, and a large number of replicas may be necessary. Nevertheless, the p-value is asymptotically unbiased (i.e. uniformly distributed on $[0, 1]$ under the null) as long as no distinction is made between original and replica datasets in our search. We could avoid this issue by using the Bayesian spatial scan, but in this case we must decide whether to condition the prior probability of attack on the number of regions found, and this decision depends on the properties of our focused search method.

8.2.5 Tracking disease state over time

In disease surveillance, it is important not only to detect potential clusters of disease but also to investigate and evaluate these clusters. One necessary tool for public health users is to be able to *track* the detected clusters over time: for example, they might want to examine the current progression of a previously detected cluster, or look back in time to find other clusters that overlap with the current cluster. We are in the process of implementing these tools (using the new TCUBE structure for computationally efficient time series aggregation and analysis) and adding them to our SSS cluster detection software. Nevertheless, these methods of tracking clusters are ad hoc investigation tools, and do not improve our ability to detect clusters. On the other hand, we might be able to achieve better detection power by drawing inferences about the underlying disease state in each spatial location, and tracking this disease state over time. One possible approach would be to use a separate hidden Markov model (HMM) for each location, using the current and past observations (counts and baselines) to infer the most likely sequence of disease states for the current and previous time steps. Brigham Anderson is currently working on a multivariate HMM model for disease surveillance, and one of our long-term goals is to combine this method with spatial scan, thus creating a method which can both track disease state and take spatial information into account. An even more powerful approach might be to move from the HMM approach to one which incorporates a Markov random field (MRF) or hidden Markov random field (HMRF). This would allow us to model the probabilistic relationships between adjacent spatial locations, enabling us to account for spatial correlation and to explicitly model the spread of disease.

8.2.6 Automatic learning from user feedback

As discussed above, the multivariate Bayesian scan statistic approach will allow us to discriminate between relevant and irrelevant clusters by incorporating probabilistic models of each potential cause of a cluster. One challenge is that the number of potential causes may be very large, including both a variety of relevant causes (e.g. different outbreak types in the disease surveillance domain) and a variety of irrelevant causes (e.g. inclement weather). It would take huge amounts of an expert’s time and knowledge to model each of these potential causes, and many such causes may not be recognized until they are observed in practice. Thus we need some way of quickly and easily adapting our models, without imposing an undue burden on human users. One possibility is to learn the causal models *automatically* from user feedback. The idea is that users can classify detected clusters into groups by labeling them as corresponding to one of a given set of causes, some of which are labeled as relevant and some as irrelevant. A model for each potential cause is learned automatically from the labeled examples corresponding to that cause. An initial set of causes can be provided in advance, but the system's utility would be increased by using our anomaly detection methods to automatically propose new causal models when none of the pre-existing causes are able to explain the data.

One eventual goal of this work is to generalize our cluster detection methods into a general and widely applicable system for the automatic discovery of relevant patterns. Our proposed system combines pattern detection and investigation with two forms of model learning: learning a model of the environment from observation, and learning a model that discriminates "relevant" from "irrelevant" patterns using human feedback. We combine these techniques in an iterative process consisting of four stages. In the *detection* stage, the system automatically detects potentially interesting subsets of the data, using the general pattern detection techniques discussed above to find subsets of the observed data which are sufficiently anomalous *as a group* to invalidate its current environmental model. In the *investigation* stage, the system applies a given set of "tools" (e.g. statistical tests) to gather more information about each detected pattern; patterns that are not sufficiently "robust" or "significant" may be discarded at this stage. In the *explanation* stage, the system proposes hypotheses as to potential causes of each pattern. A new hypothesis can be derived from the data using our Bayesian data mining methods: in addition to detecting an anomalous group, these methods can infer a possible process that explains the anomaly with high probability. We can also obtain hypotheses from the current relevance model (which consists of a set of potential causes of anomalies, each labeled as relevant or irrelevant), and determine whether any of these might be sufficient to explain the anomalous group. More statistical tests can be used to compare these hypotheses (e.g. examining how well the pattern generalizes to other, held out data), enabling the system to choose a "best" hypothesis for each pattern. At this stage, we can discard patterns that can be "explained away" as being due to some irrelevant cause.

The final, and most interesting, stage of our system is *model revision* via incorporation of human feedback. In order to maximize the autonomy of the system, and minimize the burden on its human "supervisor," we use an *active learning* framework in which the system chooses a small set of potentially relevant patterns and presents these for the human to critique. As in the standard active learning task, the system must choose between "exploitation" (presenting patterns that are most likely to be relevant) and "exploration" (presenting patterns that will improve the system's ability to discriminate between relevant and irrelevant). However, the system's queries consist of a *detected pattern* rather than an individual data point: each pattern includes an anomalous group of points, the system's explanatory hypothesis as to the cause of this anomaly, and the results of the investigations performed to verify this hypothesis. The response to these queries can also be more complex: in addition to deciding whether the pattern is "relevant" or "irrelevant," the human can also "correct" the system's work in several ways, such as proposing a new hypothesis of what is causing the anomaly, or modifying the anomalous group by adding or removing data points. In addition to enabling the system to update its relevance model, this feedback may also require revising the environment model, and possibly changing the set of investigation tools and their parameters. For example, if the system incorrectly identifies a group as anomalous, it must correct this misconception by generalizing its environment model to a larger hypothesis space, incorporating both the original model and the system's best hypothesis explaining the anomalous group. Our work toward this general system for pattern discovery is still in the early stages, but we believe it has the potential to make a significant contribution to disease surveillance and many other application areas.

Bibliography

[1] A. M. Abrams, M. Kulldorff, and K. Kleinman. Empirical/asymptotic p-values for Monte Carlo-based hypothesis testing: an application to cluster detection using the scan statistic. *Advances in Disease Surveillance*, 2006, in press.

[2] R. J. Adler. *The Geometry of Random Fields*. Wiley, 1981.

[3] D. Agarwal, J. M. Phillips, and S. Venkatasubramanian. The hunting of the bump: on maximizing statistical discrepancy. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2006, in press.

[4] R. Agrawal, J. Gehrke, D. Gunopulus, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In *Proc. ACM-SIGMOD Intl. Conf. on Mgmt. of Data*, pages 94–105, 1998.

[5] G. Aguirre, E. Zarahn, and M. D'Esposito. A critique of the use of the Kolmogorov-Smirnov statistic for the analysis of BOLD fMRI data. *Magnetic Resonance in Medicine*, 39:500–505, 1998.

[6] S. E. Alm. On the distribution of the scan statistic of a two dimensional Poisson process. *Advances in Applied Probability*, 29:1–16, 1997.

[7] S. E. Alm. Approximations and simulations of the distribution of scan statistics for Poisson processes in higher dimensions. *Extremes*, 1:111–126, 1998.

[8] N. H. Anderson and D. M. Titterington. Some methods for investigating spatial clustering, with epidemiological applications. *Journal of the Royal Statistical Society A*, 160:87–105, 1997.

[9] R. M. Assuncao, A. I. Tavares, and M. Kulldorff. An early warning system for space-time cluster detection. Technical report, 2004.

[10] R. D. Baker. Testing for space-time clusters of unknown size. *Journal of Applied Statistics*, 23(5):543–554, 1996.

[11] M. Belmonte and D. Yergelun-Todd. Permutation testing made practical for functional magnetic resonance image analysis. *IEEE Transactions on Medical Imaging*, 20:243–248, 2001.

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

[13] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18:509–517, 1975.

[14] J. Besag and P. Clifford. Generalized Monte Carlo significance tests. *Biometrika*, 76(4):633–642, 1989.

[15] J. Besag and J. Newell. The detection of clusters in rare diseases. *Journal of the Royal Statistical Society A*, 154:143–155, 1991.

[16] J. Besag, J. York, and A. Mollié. Bayesian image restoration with two applications in spatial statistics. *Ann. Inst. Statist. Math.*, 43:1–59, 1991.

[17] C. M. Bishop. Novelty detection and neural network validation. *IEEE Proceedings on Vision, Image, and Signal Processing*, 141(4):217–222, 1994.

[18] J. F. Bithell. The choice of test for detecting raised disease risk near a point source. *Statistics in Medicine*, 14:2309–2322, 1995.

[19] M. Bonetti and M. Pagano. The interpoint distance distribution as a descriptor of point patterns, with an application to spatial disease clustering. *Statistics in Medicine*, 24:753–773, 2005.

[20] C. E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. In *Studi in Onore del Professore Salvatore Ortu Carboni*, pages 13–60, 1935.

[21] N. Breslow and N. Day. *Statistical Methods in Cancer Research, Volume 2: The Design and Analysis of Cohort Studies*. International Agency for Research on Cancer, 1987.

[22] D. L. Buckeridge, H. Burkom, M. Campbell, W. R. Hogan, and A. W. Moore. Algorithms for rapid outbreak detection: a research synthesis. *Journal of Biomedical Informatics*, 38:99–113, 2005.

[23] D. L. Buckeridge, H. S. Burkom, A. W. Moore, J. A. Pavlin, P. N. Cutchis, and W. R. Hogan. Evaluation of syndromic surveillance systems: development of an epidemic simulation model. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):137–143, 2004.

[24] D. L. Buckeridge, M. A. Musen, P. Switzer, and M. Crubezy. An analytic framework for space-time aberrancy detection in public health surveillance data. In *Proc. AMIA Annual Symposium*, pages 120–124, 2003.

[25] H. S. Burkom. Biosurveillance applying scan statistics with multiple, disparate data sources. *Journal of Urban Health*, 80(2 Suppl. 1):i57–i65, 2003.

[26] H. S. Burkom, Y. Elbert, A. Feldman, and J. Lin. The role of data aggregation in biosurveillance detection strategies with applications from ESSENCE. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):67–73, 2004.

[27] H. S. Burkom, S. Murphy, J. Coberly, and K. Hurt-Mullen. Public health monitoring tools for multiple data streams. *Morbidity and Mortality Weekly Report*, 54 (Supplement on Syndromic Surveillance):55–62, 2005.

[28] D. Clayton and J. Kaldor. Empirical Bayes estimates of age-standardized relative risks for use in disease mapping. *Biometrics*, 43:671–681, 1987.

[29] D. G. Clayton and L. Bernardinelli. Bayesian methods for mapping disease risk. In P. Elliot, J. Cuzick, D. English, and R. Stern, editors, *Geographical and Environmental Epidemiology: Methods for Small Area Studies*, pages 205–220, Oxford, 1992. Oxford University Press.

[30] G. F. Cooper, D. H. Dash, J. D. Levander, W.-K. Wong, W. R. Hogan, and M. M. Wagner. Bayesian biosurveillance of disease outbreaks. In *Proc. Conference on Uncertainty in Artificial Intelligence*, 2004.

[31] J. Cuzick and R. Edwards. Spatial clustering for inhomogeneous populations. *Journal of the Royal Statistical Society B*, 52:73–104, 1990.

[32] R. Davies and the ECADS partners and collaborators. Detection of Walkerton gastroenteritis outbreak using syndromic surveillance of emergency room records. *Advances in Disease Surveillance*, 2006, in press.

[33] R. Davies and the ECADS partners and collaborators. Detection and characterization of Walkerton outbreak by emergency room syndromic surveillance. 2006, submitted for publication.

[34] K. Deng and A. W. Moore. Multiresolution instance-based learning. In *Proc. 12th Intl. Joint Conf. on Artificial Intelligence*, pages 1233–1239, 1995.

[35] P. J. Diggle. A point process modelling approach to raised incidence of a rare phenomenon in the vicinity of a prespecified point. *Journal of the Royal Statistical Society A*, 153:349–362, 1990.

[36] P. J. Diggle and A. G. Chetwynd. Second-order analysis of spatial clustering for inhomogeneous populations. *Biometrics*, 47:1155–1163, 1991.

[37] D. Donoho and J. Jin. Higher criticism for detecting sparse heterogeneous mixtures. *Annals of Statistics*, 32(3):962–994, 2004.

[38] L. Duczmal and R. Assuncao. A simulated annealing strategy for the detection of arbitrary shaped spatial clusters. *Computational Statistics and Data Analysis*, 45:269–286, 2004.

[39] L. Duczmal, L. F. Bessegato, and A. L. F. Canado. A genetic algorithm for irregularly shaped spatial clusters. *Advances in Disease Surveillance*, 2006, in press.

[40] L. Duczmal and D. Buckeridge. Using modified spatial scan statistic to improve detection of disease outbreak when exposure occurs in workplace–Virginia, 2004. *Morbidity and Mortality Weekly Report*, 54 (Supplement on Syndromic Surveillance):187, 2005.

[41] L. Duczmal, M. Kulldorff, and L. Huang. Evaluation of spatial scan statistics for irregularly shaped clusters. *J. Comput. Graph. Stat.*, 2005, in press.

[42] W. DuMouchel. Bayesian data mining in large frequency tables, with an application to the FDA spontaneous reporting system. *American Statistician*, 53:177–202, 1999.

[43] M. Dwass. Modified randomization tests for nonparametric hypotheses. *Annals of Mathematical Statistics*, 28:181–187, 1957.

[44] P. Elliott, J. C. Wakefield, N. G. Best, and D. J. Briggs, editors. *Spatial Epidemiology: Methods and Applications*. Oxford University Press, Oxford, 2000.

[45] E. Eskin. Anomaly detection over noisy data using learned probability distributions. In *Proc. 2000 International Conference on Machine Learning*, 2000.

[46] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.

[47] T. Fawcett and F. Provost. Activity monitoring: noticing interesting changes in behavior. In *Proc. 5th Intl. Conf. on Knowledge Discovery and Data Mining*, pages 53–62, 1999.

[48] R. A. Finkel and J. L. Bentley. Quadtrees: a data structure for retrieval on composite keys. *Acta Informatica*, 4:1–9, 1974.

[49] J. Friedman and N. Fisher. Bump hunting in high dimensional data. *Statistics and Computing*, 9(2):1–20, 1999.

[50] K. Friston, A. P. Holmes, K. J. Worsley, J. B. Poline, C. D. Frith, and R. S. J. Frackowiak. Statistical parametric maps in functional imaging- a general linear approach. *Human Brain Mapping*, 2:189–210, 1995.

[51] R. E. Gangnon and M. K. Clayton. Bayesian detection and modeling of spatial disease clustering. *Biometrics*, 56:922–935, 2000.

[52] R. E. Gangnon and M. K. Clayton. A weighted average likelihood ratio scan test for spatial clustering of disease. Technical Report 162, University of Wisconsin-Madison, Department of Biostatistics and Medical Informatics, 2001.

[53] R. E. Gangnon and M. K. Clayton. A weighted average likelihood ratio test for spatial disease clustering. *Statistics in Medicine*, 20:2977–2987, 2001.

[54] R. E. Gangnon and M. K. Clayton. A hierarchical model for spatial clustering of disease. *Statistics in Medicine*, 22:3213–3228, 2003.

[55] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741, 1984.

[56] C. R. Genovese. A Bayesian time-course model for functional magnetic resonance imaging data (with discussion). *Journal of the American Statistical Association*, 95:691–703, 2000.

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

[58] J. Glaz, J. Naus, and S. Wallenstein. *Scan Statistics*. Springer-Verlag, New York, 1999.

[59] S. Goil, H. Nagesh, and A. Choudhary. MAFIA: efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern University, 1999.

[60] P. Good. *Permutation Tests– A Practical Guide to Resampling Methods for Testing Hypotheses*. Springer-Verlag, New York, 2000.

[61] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

[62] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. ACM SIGMOD Intl. Conf. on Management of Data*, pages 47–57, 1984.

[63] J. Hamilton, editor. *Time Series Analysis*. Princeton University Press, 1994.

[64] J. Han, M. Kamber, and A. Tung. Spatial clustering methods in data mining: a survey. In H. Miller and J. Han, editors, *Geographic Data Mining and Knowledge Discovery*. Taylor and Francis, 2001.

[65] A. Harvey. The Kalman filter and its applications in econometrics and time series analysis. *Methods for Operations Research*, 44:3–18, 1981.

[66] R. Heffernan, F. Mostashari, D. Das, A. Karpati, M. Kulldorff, and D. Weiss. Syndromic surveillance in public health practice. *Emerging Infectious Diseases*, 10(5):858–864, 2004.

[67] M. Hills and F. Alexander. Statistical methods used in assessing the risk of disease near a source of possible environmental pollution: a review. *Journal of the Royal Statistical Society A*, 152:353–363, 1989.

[68] A. Hinneburg and D. A. Keim. An efficient approach to clustering in large multimedia databases with noise. In *Proc. 1998 International Conference on Knowledge Discovery and Data Mining*, pages 58–65, 1998.

[69] U. Hjalmars, M. Kulldorff, G. Gustafsson, and N. Nagarwalla. Childhood leukemia in Sweden: using GIS and a spatial scan statistic for cluster detection. *Statistics in Medicine*, 15:707–715, 1996.

[70] W. R. Hogan, G. F. Cooper, M. M. Wagner, and G. L. Wallstrom. A Bayesian anthrax aerosol release detector. Technical report, RODS Laboratory, 2004.

[71] J. S. Hunter. The exponentially weighted moving average. *Journal of Quality Technology*, 18(4):203–207, 1986.

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

[73] K. Ickstadt and R. L. Wolpert. Spatial correlation or spatial variation? a comparison of Gamma/Poisson hierarchical models. Technical report, Duke University, Institute of Statistics and Decision Sciences, 1996.

[74] V. Iyengar. On detecting space-time clusters. In *Proc. 10th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 587–592, 2004.

[75] K. Kleinman, A. Abrams, M. Kulldorff, and R. Platt. A model-adjusted space-time scan statistic with an application to syndromic surveillance. *Epidemiology and Infection*, 133(3):409–419, 2005.

[76] K. Kleinman, R. Lazarus, and R. Platt. A generalized linear mixed models approach for detecting incident clusters of disease in small areas, with an application to biological terrorism. *American Journal of Epidemiology*, 159:217–224, 2004.

[77] E. G. Knox. The detection of space-time interactions. *Applied Statistics*, 13:25–29, 1964.

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

[79] M. Kulldorff. The Knox method and other tests for space-time interaction. *Biometrics*, 55:544–552, 1999.

[80] M. Kulldorff. Spatial scan statistics: models, calculations, and applications. In J. Glaz and N. Balakrishnan, editors, *Scan Statistics and Applications*, pages 303–322. Birkhauser, 1999.

[81] M. Kulldorff. Prospective time-periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society A*, 164:61–72, 2001.

[82] M. Kulldorff, W. Athas, E. Feuer, B. Miller, and C. Key. Evaluating cluster alarms: a space-time scan statistic and cluster alarms in los alamos. *American Journal of Public Health*, 88:1377–1380, 1998.

[83] M. Kulldorff, Z. Fang, and S. J. Walsh. A tree-based scan statistic for database disease surveillance. *Biometrics*, 59:323–331, 2003.

[84] M. Kulldorff, E. J. Feuer, B. A. Miller, and L. S. Freedman. Breast cancer clusters in the northeast United States: a geographic analysis. *American Journal of Epidemiology*, 146(2):161–170, 1997.

[85] M. Kulldorff, R. Heffernan, J. Hartman, R. Assuncao, and F. Mostashari. A space-time permutation scan statistic for the early detection of disease outbreaks. *PLoS Medicine*, 2(3):e59, 2005.

[86] M. Kulldorff, L. Huang, L. Pickle, and L. Duczmal. An elliptic spatial scan statistic. 2005, submitted for publication.

[87] M. Kulldorff and Information Management Services Inc. Satscan v. 4.0: Software for the spatial and space-time scan statistics. Technical report, 2004.

[88] M. Kulldorff and N. Nagarwalla. Spatial disease clusters: detection and inference. *Statistics in Medicine*, 14:799–810, 1995.

[89] A. B. Lawson. On the analysis of mortality events around a prespecified fixed point. *Journal of the Royal Statistical Society A*, 156:363–377, 1993.

[90] A. B. Lawson. Markov Chain Monte Carlo techniques for putative pollution source problems in environmental epidemiology. *Statistics in Medicine*, 14:2473–2486, 1995.

[91] A. B. Lawson. *Statistical Methods in Spatial Epidemiology*. Wiley, Chichester, UK, 2001.

[92] A. B. Lawson, A. Biggeri, D. Böhning, E. Lesaffre, J.-F. Viel, and R. Bertollini, editors. *Disease Mapping and Risk Assessment for Public Health*. Wiley, New York, 1999.

[93] A. B. Lawson and A. Clark. Markov Chain Monte Carlo methods for putative sources of hazard and general clustering. In A. B. Lawson, A. Biggeri, D. Böhning, E. Lesaffre, J.-F. Viel, and R. Bertollini, editors, *Disease Mapping and Risk Assessment for Public Health*, 1999.

[94] A. B. Lawson and D. G. T. Denison, editors. *Spatial Cluster Modelling*. Chapman & Hall/CRC, Boca Raton, FL, 2002.

[95] A. B. Lawson and K. Kleinman, editors. *Spatial and Syndromic Surveillance for Public Health*. Wiley, Chichester, UK, 2005.

[96] Y. Le Strat and F. Carrat. Monitoring epidemiologic surveillance data using hidden Markov models. *Statistics in Medicine*, 18:3463–3478, 1999.

[97] C. Loader. Large deviation approximations to the distribution of scan statistics. *Advances in Applied Probability*, 23:751–771, 1991.

[98] J. Lombardo, H. Burkom, and E. Elbert. A systems overview of the Electronic Surveillance System for the Early Notification of Community-Based Epidemics (ESSENCE II). *Journal of Urban Health*, 80(2 Suppl. 1):i32–42, 2003.

[99] J. Loonsk. BioSense- a national initiative for early detection and quantification of public health emergencies. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):53–55, 2004.

[100] K. D. Mandl, J. M. Overhage, M. M. Wagner, W. B. Lober, P. Sebastiani, and F. Mostashari, et al. Implementing syndromic surveillance: a practical guide informed by the early experience. *Journal of the American Medical Informatics Association*, 11(2):141–150, 2003.

[101] N. Mantel. The detection of cancer clustering and the generalized regression approach. *Cancer Research*, 27:209–220, 1967.

[102] M. Meselson, J. Guillemin, and M. Hugh-Jones et al. The Sverdlovsk anthrax outbreak of 1979. *Science*, 266(5188):1202–1208, 1994.

[103] T. M. Mitchell, R. Hutchinson, R. S. Niculescu, F. Pereira, X. Wang, M. Just, and S. Newman. Learning to decode cognitive states from brain images. *Machine Learning*, 57:145–175, 2004.

[104] A. Mollié. Bayesian mapping of disease. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter, editors, *Markov Chain Monte Carlo in Practice*. Chapman & Hall, 1995.

[105] A. Mollié. Bayesian and empirical Bayes approaches to disease mapping. In A. B. Lawson, A. Biggeri, D. Böhning, E. Lesaffre, J.-F. Viel, and R. Bertollini, editors, *Disease Mapping and Risk Assessment for Public Health*, 1999.

[106] A. W. Moore and M. S. Lee. Cached sufficient statistics for efficient machine learning with large datasets. *J. Artificial Intelligence Research*, 8:67–91, 1998.

[107] F. Mostashari, M. Kulldorff, J. J. Hartman, J. R. Miller, and V. Kulasekera. Dead bird clustering: A potential early warning system for West Nile virus activity. *Emerging Infectious Diseases*, 9:641–646, 2003.

[108] J. I. Naus. The distribution of the size of the maximum cluster of points on the line. *Journal of the American Statistical Association*, 60:532–538, 1965.

[109] D. B. Neill and A. W. Moore. Detecting space-time clusters: prior work and new directions. Technical report, Carnegie Mellon University, 2004.

[110] D. B. Neill and A. W. Moore. A fast grid-based scan statistic for detection of significant spatial disease clusters. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):255, 2004.

[111] D. B. Neill and A. W. Moore. A fast multi-resolution method for detection of significant spatial disease clusters. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*. MIT Press, 2004.

[112] D. B. Neill and A. W. Moore. Rapid detection of significant spatial clusters. In *Proc. 10th ACM SIGKDD Conf. on Knowledge Discovery and Data Mining*, pages 256–265, 2004.

[113] D. B. Neill and A. W. Moore. Anomalous spatial cluster detection. In *Proc. KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, pages 41–44, 2005.

[114] D. B. Neill and A. W. Moore. Efficient scan statistics computations. In A. Lawson and K. Kleinman, editors, *Spatial and Syndromic Surveillance for Public Health*, 2005.

[115] D. B. Neill and A. W. Moore. Methods for detecting spatial and spatio-temporal clusters. In M. M. Wagner, A. W. Moore, and R. Aryel, editors, *Handbook of Biosurveillance*. Elsevier, 2006.

[116] D. B. Neill, A. W. Moore, and G. F. Cooper. A Bayesian spatial scan statistic. In *Advances in Neural Information Processing Systems 18*, pages 1003–1010, 2006.

[117] D. B. Neill, A. W. Moore, and G. F. Cooper. A Bayesian scan statistic for spatial cluster detection. *Advances in Disease Surveillance*, 2006, in press.

[118] D. B. Neill, A. W. Moore, F. Pereira, and T. Mitchell. Detecting significant multidimensional spatial clusters. In *Advances in Neural Information Processing Systems 17*, pages 969–976, 2005.

[119] D. B. Neill, A. W. Moore, and M. R. Sabhnani. Detecting elongated disease clusters. *Morbidity and Mortality Weekly Report*, 54 (Supplement on Syndromic Surveillance):197, 2005.

[120] D. B. Neill, A. W. Moore, M. R. Sabhnani, and K. Daniel. Detection of emerging space-time clusters. In *Proc. 11th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, 2005.

[121] D. B. Neill, A. W. Moore, M. R. Sabhnani, and K. Daniel. An expectation-based scan statistic for detection of space-time clusters. *Advances in Disease Surveillance*, 2006, in press.

[122] S. Openshaw, M. Charlton, A. Craft, and J. Birch. Investigation of leukemia clusters by use of a geographical analysis machine. *Lancet*, 1:272–3, 1988.

[123] E. S. Page. Continuous inspection schemes. *Biometrika*, 41:100–115, 1954.

[124] G. P. Patil and C. Taillie. Geographic and network surveillance via scan statistics for critical area detection. *Statistical Science*, 18(4):457–465, 2003.

[125] G. P. Patil and C. Taillie. Upper level set scan statistic for detecting arbitrarily shaped hotspots. *Envir. Ecol. Stat.*, 11:183–197, 2004.

[126] M. Perone Pacifico, C. R. Genovese, I. Verdinelli, and L. Wasserman. False discovery rates for random fields. Technical Report 771, Carnegie Mellon University, Department of Statistics, 2003.

[127] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.

[128] S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society B*, 59:731–792, 1997.

[129] P. A. Rogerson. Monitoring point patterns for the development of space-time clusters. *Journal of the Royal Statistical Society A*, 164, 87-96.

[130] M. R. Sabhnani, A. W. Moore, and A. W. Dubrawski. Fast extraction of time series from large datasets. Technical Report CMU-ML-06-104, Carnegie Mellon University, 2006.

[131] M. R. Sabhnani, D. B. Neill, A. W. Moore, F.-C. Tsui, M. M. Wagner, and J. U. Espino. Detecting anomalous clusters in pharmacy retail data. In *Proc. KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, pages 58–61, 2005.

[132] M. R. Sabhnani, D. B. Neill, A. W. Moore, F.-C. Tsui, M. M. Wagner, and J. U. Espino. Monitoring pharmacy retail data for anomalous space-time clusters. *Advances in Disease Surveillance*, 2006, in press.

[133] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, Reading, MA, 1990.

[134] P. Schlattmann and D. Böhning. Mixture models and disease mapping. *Statistics in Medicine*, 12:1943–1950, 1993.

[135] J. Schneider and A. W. Moore. Efficient scan statistics with general probabilistic models. Technical report, Carnegie Mellon University, 2003.

[136] R. E. Serfling. Methods for current statistical analysis of excess pneumonia-influenza deaths. *Public Health Reports*, 78, 494-506.

[137] W. A. Shewhart. *Economic Control of Quality of Manufactured Product*. 1931.

[138] G. Shmueli, T. P. Minka, J. B. Kadane, S. Borle, and P. Boatwright. A useful distribution for fitting discrete data: revival of the Conway-Maxwell-Poisson distribution. *Journal of the Royal Statistical Society C*, 54:127–142, 2005.

[139] D. O. Siegmund and K. J. Worsley. Testing for a signal with unknown location and scale in a stationary Gaussian random field. *Annals of Statistics*, 23:608–639, 1995.

[140] A. M. Smith, B. K. Lewis, U. E. Ruttimann, F. Q. Ye, T. M. Sinnwell, Y. Yang, J. H. Duyn, and J. A. Frank. Investigation of low frequency drift in fMRI signal. *NeuroImage*, 9:526–533, 1999.

[141] J. Snow. *On the Mode of Communication of Cholera*. John Churchill, London, UK, 1855.

[142] R. A. Stone. Investigations of excess environmental risks around putative sources: statistical problems and a proposed test. *Statistics in Medicine*, 7:649–660, 1988.

[143] T. Tango. A class of tests for detecting general and focused clustering of rare diseases. *Statistics in Medicine*, 14:2323–2334, 1995.

[144] T. Tango and K. Takahashi. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 4:11, 2005.

[145] F.-C. Tsui, J. U. Espino, V. M. Dato, P. H. Gesteland, J. Hutman, and M. M. Wagner. Technical description of RODS: a real-time public health surveillance system. *Journal of the American Medical Informatics Association*, 10(5):399–408, 2003.

[146] B. W. Turnbull, E. J. Iwano, W. S. Burnett, H. L. Howe, and L. C. Clark. Monitoring for clusters of disease: Application to leukemia in upstate New York. *American Journal of Epidemiology*, 132:S136–S143, 1990.

[147] M. M. Wagner, A. W. Moore, and R. Aryel, editors. *Handbook of Biosurveillance*. Elsevier, 2006, in press.

[148] M. M. Wagner, J. Robinson, F.-C. Tsui, J. U. Espino, and W. Hogan. Design of a national retail data monitor for public health surveillance. *Journal of the American Medical Informatics Association*, 10(5):409–418, 2003.

[149] M. M. Wagner, F.-C. Tsui, J. U. Espino, V. M. Dato, D. F. Sittig, and R. A. Caruana, et al. The emerging science of very early detection of disease outbreaks. *Journal of Public Health Management and Practice*, 7(6):51–59, 2001.

[150] M. M. Wagner, F.-C. Tsui, J. U. Espino, W. Hogan, J. Hutman, J. Hirsch, D. B. Neill, A. W. Moore, G. Parks, C. Lewis, and R. Aller. A national retail data monitor for public health surveillance. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):40–42, 2004.

[151] L. Waller, B. Carlin, H. Xia, and A. Gelfand. Hierarchical spatio-temporal mapping of disease rates. *Journal of the American Statistical Association*, 92:607–617, 1997.

[152] L. A. Waller, B. W. Turnbull, L. C. Clark, and P. Nasca. Chronic disease surveillance and testing of clustering of disease and exposure: Application to leukemia incidence and TCE-contaminated dumpsites in upstate New York. *Environmetrics*, 3:281–300, 1992.

[153] L. A. Waller, B. W. Turnbull, L. C. Clark, and P. Nasca. Spatial analysis to detect disease clusters. In N. Lange, editor, *Case Studies in Biometry*, pages 3–23. Wiley, 1994.

[154] G. L. Wallstrom, M. M. Wagner, and W. R. Hogan. High-fidelity injection detectability experiments: a tool for evaluation of syndromic surveillance systems. *Morbidity and Mortality Weekly Report*, 54 (Supplement on Syndromic Surveillance):85–91, 2005.

[155] W. Wang, J. Yang, and R. Muntz. STING: a statistical information grid approach to spatial data mining. In *Proc. 23rd Conference on Very Large Databases*, pages 186–195, 1997.

[156] X. Wang, R. Hutchinson, and T. Mitchell. Training fMRI classifiers to detect cognitive states across multiple human subjects. In *Advances in Neural Information Processing Systems 16*, pages 709–716, 2004.

[157] A. Whittemore, N. Friend, B. Brown, and E. Holly. A test to detect clusters of disease. *Biometrika*, 74:631–635, 1987.

[158] R. L. Wolpert and K. Ickstadt. Poisson/Gamma random fields for spatial statistics. *Biometrika*, 85:251–267, 1998.

[159] W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner. Rule-based anomaly pattern detection for detecting disease outbreaks. In *Proc. 18th National Conference on Artificial Intelligence*, 2002.

[160] W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner. Bayesian network anomaly pattern detection for disease outbreaks. In *Proc. 20th International Conference on Machine Learning*, 2003.

[161] W.-K. Wong, A. W. Moore, G. F. Cooper, and M. M. Wagner. WSARE: What's strange about recent events? *Journal of Urban Health*, 80(2 Suppl. 1):i66–i75, 2003.

[162] K. J. Worsley. Local maxima and expected Euler characteristic of excursion sets of χ^2, F and t random fields. *Advances in Applied Probability*, 27:943–959, 1994.

[163] K. J. Worsley. Detecting activation in fMRI data. *Statistical Methods in Medical Research*, 12:401–418, 2003.

[164] K. J. Worsley, A. C. Evans, S. Marrett, and P. Neelin. A three dimensional statistical analysis for CBF activation studies in human brain. *Journal of Cerebral Blood Flow and Metabolism*, 12:900–918, 1992.

[165] K. J. Worsley, C. H. Liao, J. Aston, V. Petre, G. H. Duncan, F. Morales, and A. C. Evans. A general statistical analysis for fMRI data. *NeuroImage*, 15:1–15, 2002.

[166] K. J. Worsley, S. Marrett, P. Neelin, A. C. Vandal, K. J. Friston, and A. C. Evans. A unified statistical approach for detecting significant signals in images of cerebral activation. *Human Brain Mapping*, 4:58–73, 1996.

[167] W. K. Yih, B. Caldwell, and R. Harmon. The national bioterrorism syndromic surveillance demonstration program. *Morbidity and Mortality Weekly Report*, 53 (Supplement on Syndromic Surveillance):43–46, 2004.

[168] J. Zhang, F.-C. Tsui, M. M. Wagner, and W. R. Hogan. Detection of outbreaks from time series data using wavelet transform. In *Proc. AMIA Annual Symposium*, pages 748–752, 2003.