Detection of Emerging Space-Time Clusters

Daniel B. Neill, Andrew W. Moore, Maheshkumar Sabhnani, Kenny Daniel
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{neill, awm, sabhnani, kfd}@cs.cmu.edu

ABSTRACT

We propose a new class of spatio-temporal cluster detection methods designed for the rapid detection of emerging space-time clusters. We focus on the motivating application of prospective disease surveillance: detecting space-time clusters of disease cases resulting from an emerging disease outbreak. Automatic, real-time detection of outbreaks can enable rapid epidemiological response, potentially reducing rates of morbidity and mortality. Building on the prior work on spatial and space-time scan statistics, our methods combine time series analysis (to determine how many cases we expect to observe for a given spatial region in a given time interval) with new "emerging cluster" space-time scan statistics (to decide whether an observed increase in cases in a region is significant), enabling fast and accurate detection of emerging outbreaks. We evaluate these methods on two types of simulated outbreaks: aerosol release of inhalational anthrax (e.g. from a bioterrorist attack) and FLOO ("Fictional Linear Onset Outbreak"), injected into actual baseline data (Emergency Department records and over-the-counter drug sales data from Allegheny County). We demonstrate that our methods are successful in rapidly detecting both outbreak types while keeping the number of false positives low, and show that our new "emerging cluster" scan statistics consistently outperform the standard "persistent cluster" scan statistics approach.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Apps- Data Mining

General Terms

Algorithms

Keywords

Cluster detection, space-time scan statistics, biosurveillance

1. INTRODUCTION

In many data mining applications, we are faced with the task of detecting *clusters*: regions of space where some quantity is significantly higher than expected. For example, our goal may be to detect clusters of disease cases, which may be indicative of a naturally occurring disease epidemic (e.g. influenza), a bioterrorist attack (e.g. anthrax release), or an environmental hazard (e.g. radiation leak). In medical imaging, we may attempt to detect tumors or other hazardous growths; in neuroscience, we may be interested in detecting spatial patterns of brain activity (measured by fMRI activation) that correspond to various cognitive tasks. [11] discusses many other applications of cluster detection, including mining astronomical data (identifying clusters of stars or galaxies) and military reconnaissance (monitoring strength and activity of enemy forces). In all of these applications, we have two main goals: to pinpoint the location, shape, and size of each potential cluster, and to determine (by statistical significance testing) whether each potential cluster is likely to be a "true" cluster or simply a chance occurrence.

While most of the prior work on cluster detection is purely spatial in nature (e.g. [1, 10, 6]), it is clear from the above list of applications that *time* is an essential component of most cluster detection problems. We are often interested in clusters which are *emerging* in time: for example, a growing tumor, an outbreak of disease, or an increase in troop activity. In some applications, the time dimension can be dealt with easily, either by applying some purely spatial cluster detection method at each time step, or by treating time as another spatial dimension and thus applying spatial cluster detection in a $d+1$ dimensional space (d spatial dimensions, plus time). The disadvantage of the first approach is that by only examining one day of data at a time, we may fail to detect more slowly emerging clusters. The disadvantage of the second approach is that we may detect less *relevant* clusters: those clusters that have persisted for a long time, rather than those that are newly emerging.

To improve on these methods, it is helpful to consider the guiding question, "How is time, as a dimension, different from space?" We argue that there are three important distinctions which require us to treat spatio-temporal cluster detection differently from spatial cluster detection. First, time (unlike space) has an important point of reference: the present. We often care only about those space-time clusters that are still "active" at the present time, and in these cases we should use a prospective method (searching for clusters which end at the present time) rather than a retrospective method (searching for clusters which end at or before the present time). Second, in the spatial cluster detection framework, we typically assume that we have some baseline denominator data such as a census population (for epidemiology), and that the expected count (e.g. number of disease cases) is proportional to this baseline. In the spatio-temporal framework, we are generally not provided with explicit denominator data; instead, we infer the expected values of the most recent days' counts from the time series of past counts. Finally, and most interestingly, time has an explicit direction or "ar-

KDD'05, August 21–24, 2005, Chicago, Illinois, USA.

row," proceeding from the past, through the present, to the future. We are generally interested in clusters which emerge over time: for example, a disease epidemic may start with only a few reported cases, then increase in magnitude either gradually or rapidly. One major focus of this paper is developing statistical methods which are more appropriate for detecting such emerging clusters.

We focus here on the motivating application of *prospective disease surveillance*: detecting space-time clusters of disease cases resulting from an emerging disease outbreak. In this application, we perform surveillance on a daily (or even hourly) basis, with the goal of finding emerging epidemics as quickly as possible. By detecting epidemics 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 from an outbreak. In this application, we are given the number of disease cases of some given type (e.g. respiratory) in each spatial location (e.g. zip code) on each day. More precisely, since we cannot measure the actual number of cases, we instead rely on related observable quantities such as the number of respiratory Emergency Department visits, or sales of over-the-counter cough and cold medication, in a given spatial location on a given day. We must then detect those increases that are indicative of emerging outbreaks, as close to the start of the outbreak as possible, while keeping the number of false positives low.

2. THE MODEL

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 (as in [18]) 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 $S \subseteq G$.[1] 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, $c_i^t \sim \text{Po}(\lambda_i^t)$ 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. In addition to the temporal window size, three other considerations may impact the performance of our method: 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.

[1] Non-axis-aligned rectangles can be detected by examining multiple rotations of the data, as in [18].

3. SPACE-TIME SCAN STATISTICS

One of the most important statistical tools for cluster detection is the *spatial scan statistic* [15, 10, 11]. 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 [11] 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 prior work, we have addressed both of these problems by proposing the "fast spatial scan" algorithm [18, 19], 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 formulation [15, 10], 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 $D(S) = \frac{\Pr(\text{Data} \mid H_1(S))}{\Pr(\text{Data} \mid H_0)}$, 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 $c_i \sim \text{Po}(q_{all} b_i)$ for all locations s_i. Under $H_1(S)$, we assume that $c_i \sim \text{Po}(q_{in} b_i)$ for all locations $s_i \in S$, and $c_i \sim \text{Po}(q_{out} b_i)$ for all locations $s_i \in G - S$, for some constants $q_{in} > q_{out}$. From this, we can derive an expression for $D(S)$ using the maximum likelihood estimates of q_{in}, q_{out}, and q_{all}: $D(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} \left(\frac{C_{out}}{B_{out}}\right)^{C_{out}} \left(\frac{C_{all}}{B_{all}}\right)^{-C_{all}}$, if $\frac{C_{in}}{B_{in}} > \frac{C_{out}}{B_{out}}$, and $D(S) = 1$ 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 $D(S)$; we denote this region by S^*, and its score by D^*. 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 $c_i \sim \text{Po}(q_{all} b_i)$, 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 D^* 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*. 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 $c_i \sim \text{Po}(b_i)$, against the set of alternative hypotheses $H_1(S)$: for spatial locations $s_i \in S$, all counts c_i are generated by $c_i \sim \text{Po}(qb_i)$, for some constant $q > 1$, and for all other spatial locations $s_i \in G - S$, all counts $c_i \sim \text{Po}(b_i)$. We then compute the likelihood ratio:

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

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

Using the maximum likelihood estimate $q = \max\left(1, \frac{C_{in}}{B_{in}}\right)$, we obtain $D(S) = \left(\frac{C_{in}}{B_{in}}\right)^{C_{in}} e^{B_{in}-C_{in}}$, if $C_{in} > B_{in}$, and $D(S) = 1$ 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 $c_i \sim \text{Po}(b_i)$.

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: $c_i^T \sim \text{Po}(b_i^T)$ for all s_i, to the set of alternative hypotheses $H_1(S)$: $c_i^T \sim \text{Po}(qb_i^T)$ for all $s_i \in S$, for some constant $q > 1$, and $c_i^T \sim \text{Po}(b_i^T)$ elsewhere. Thus the statistic takes the same form as the purely spatial scan statistic, and we obtain: $D(S) = \left(\frac{C}{B}\right)^C e^{B-C}$, if $C > B$, and $D(S) = 1$ 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 D^*. To compute the p-value, we perform randomization testing as before, where each replica grid has counts c_i^T generated from $\text{Po}(b_i^T)$ and all other counts c_i^t ($t \neq T$) copied from the original grid.

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 $D(S, t_{min})$. 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 D^*, 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 $\text{Po}(b_i^t)$ 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 $D(S, t_{min})$. The standard method for computing the space-time scan statistic, proposed for the retrospective case by [13] and for the prospective case by [12], builds on the Kulldorff spatial scan statistic [10] 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 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), and we consider such statistics in [17].

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 $D(S, t_{min})$. For a given region S and value t_{min}, we compare the null hypothesis H_0: $c_i^t \sim \text{Po}(b_i^t)$ for all spatial locations s_i and all $T - W < t \leq T$, to the alternative hypothesis $H_1(S, t_{min})$: $c_i^t \sim \text{Po}(qb_i^t)$ for $s_i \in S$ and $t = t_{min} \ldots T$, for some constant $q > 1$, and $c_i^t \sim \text{Po}(b_i^t)$ elsewhere. Thus we can compute the likelihood ratio:

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

where the products are taken over $s_i \in S$ and $t_{min} \leq t \leq T$. This simplifies to $\max_{q>1} \frac{q^C e^{-qB}}{e^{-B}}$, 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 $D(S, t_{min}) = \left(\frac{C}{B}\right)^C e^{B-C}$ if $C > B$, and $D = 1$ otherwise.

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 $D(S, t_{min})$. For a given region S and value t_{min}, we compare the null hypothesis H_0: $c_i^t \sim \text{Po}(b_i^t)$ for all spatial locations s_i and all $T - W < t \leq T$, to the alternative hypothesis $H_1(S, t_{min})$: $c_i^t \sim \text{Po}(q_t b_i^t)$ for $s_i \in S$ and $t = t_{min} \ldots T$, for some monotoni-

cally increasing sequence of constants $1 \le q_{t_{min}} \le \ldots \le q_T$, and $c_i^t \sim \text{Po}(b_i^t)$ elsewhere. Thus we can compute the likelihood ratio:

$$D(S,t_{min}) = \frac{\max_{1 \le q_{t_{min}} \le \ldots \le q_T} \prod \Pr(c_i^t \sim \text{Po}(q_t b_i^t))}{\prod \Pr(c_i^t \sim \text{Po}(b_i^t))}$$

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

where the products are taken over $s_i \in S$ and $t_{min} \le t \le T$. This simplifies to $\frac{\max_{1 \le q_{t_{min}} \le \ldots \le q_T} \prod q_t^{C_t} e^{-q_t B_t}}{e^{-B}}$, 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 $\sum_{s_i \in S} \sum_{t_{min} \le t \le T} b_i^t$ as above.

Now, we must maximize the numerator subject to the constraints on the q_t. To do so, let $E = E_1 \ldots E_p$ be a partitioning of $t_{min} \ldots T$ 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 $t_{min} \ldots T$ 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:

$$D(S,t_{min}) = \frac{\max_{E_1 \ldots E_p} \max_{Q_1 \ldots Q_p} \prod_{E_j} \left(Q_j\right)^{C_j} e^{-Q_j B_j}}{e^{-B}}$$

where $B_j = \sum_{s_i \in S} \sum_{t \in E_j} b_i^t$ and $C_j = \sum_{s_i \in S} \sum_{t \in E_j} c_i^t$.

In [17], we prove that this expression is maximized when $Q_j = \frac{C_j}{B_j}$ for all j. This allows us to simplify the expression to:

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

Then the question is how to choose the optimal partitioning $E = \{E_j\}$, and in [17] we present the following algorithm. This method uses a stack data structure, where each element of the stack represents a partition E_j by a 5-tuple $\left(t_{start}, t_{end}, C_j, B_j, Q_j\right)$. The algorithm starts by pushing $\left(T, T, C_T, B_T, \max\left(1, \frac{C_T}{B_T}\right)\right)$ 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)
```

As we prove in [17], this "step 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.

4. INFERRING BASELINE VALUES

In order to infer the baselines b_i^t for the "current" days $T-W < t \le T$, 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 ($T-W < t \le T$) from the past counts c_i^t ($1 \le t \le T-W$), using one of the time series analysis methods below. Then whenever we calculate $D(S,t_{min})$ 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 $D(S,t_{min})$ 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 $T-W < t \le T$, and applies the score function $D(S,t_{min})$ to the resulting counts and baselines.

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. Various sampling methods can be used to do this, but this makes randomization extremely computationally expensive; see [17] for more details.

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, $b_i^t = E[c_i^t]$, $T-W < t \le T$, from the time series of "past" counts c_i^t, $1 \le t \le T-W$. A variety of methods are possible, 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. 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 biased methods have the advantage of reducing the number of false positives to a more manageable level (see Section 7.5).

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 exponen-

tially 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. [14] 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).

5. 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 [4, 3]; tests for a general tendency of the data to cluster [9, 16]; and tests which attempt to infer the location of clusters [13, 12, 14]. 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 [13, 12], and the space-time permutation scan statistic [14]. 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" [14] 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.

Several other spatio-temporal cluster detection methods have also been proposed. Iyengar [8] searches over "truncated rectangular pyramid" shapes in space-time, thus allowing detection of clusters which move and grow or shrink over time; the disadvantage is that this much larger set of possible space-time regions can only be searched approximately. Assuncao et al [2] 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.

6. 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 $D(S, t_{min})$, 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 [18] 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 D. As noted above, we can do this in $O(W)$ per region, but since a naive 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 naive search over all such regions computationally infeasible. However, we can apply the fast spatial scan of [18, 19], 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 $D(S', t_{min})$ 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 [18], 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.

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, PA.[2] 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. [7]. 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.

[2]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.

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 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).

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 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 $D^* = D(S^*)$, 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 D^*, 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 [5] (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 *cal-*

ibration: 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 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 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 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 1. We also discuss each outbreak type in more detail below.

7.2 Anthrax outbreaks, ED data

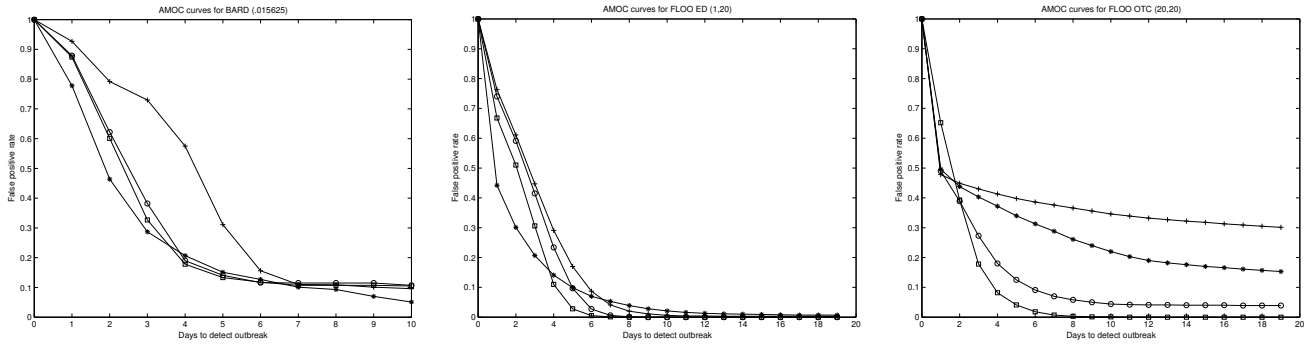
For the anthrax outbreaks, we began with real baseline data for respiratory Emergency Department visits in Allegheny County in 2002. We used this data to simulate epidemics using BARD at two different levels of anthrax release: 0.125 (high) and 0.015625 (low). For each release amount, 60 simulated epidemics 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 of our 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 the number of 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 again 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 days and 5.094 days respectively.

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).



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

Figure 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.

Table 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

Table 2: Comparison of methods. Average days to detect, 1 false positive/month, all datasets.

method	BARD (0.125)	BARD (0.015625)	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

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.

7.5 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 3 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

Table 3: 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

have high false positive rates even for the conservative methods. What conclusions can we draw from this? Because of the variance in our predictions, 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.

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.

9. REFERENCES

[1] 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.

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

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

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

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

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

[7] W. Hogan, G. Cooper, M. Wagner, and G. Wallstrom. A bayesian anthrax aerosol release detector. Technical report, RODS Laboratory, 2004.

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

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

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

[11] 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.

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

[13] 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.

[14] 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*, 2005, in press.

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

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

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

[18] 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.

[19] 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.