# Scalable Spatial Scan Statistics through Sampling

Michael Matheny
University of Utah
mmath@cs.utah.edu

Raghvendra Singh*
Inside Sales
raghvendra@cs.utah.edu

Liang Zhang*
Microsoft
lzhang81@cs.utah.edu

Kaiqiang Wang*
Google
kaiqiang.wang@utah.edu

Jeff M. Phillips†
University of Utah
jeffp@cs.utah.edu

## ABSTRACT

Finding anomalous regions within spatial data sets is a central task for biosurveillance, homeland security, policy making, and many other important areas. These communities have mainly settled on spatial scan statistics as a rigorous way to discover regions where a measured quantity (e.g., crime) is statistically significant in its difference from a baseline population. However, most common approaches are inefficient and thus, can only be run with very modest data sizes (a few thousand data points) or make assumptions on the geographic distributions of the data.

We address these challenges by designing, exploring, and analyzing *sample-then-scan* algorithms. These algorithms randomly sample data at two scales, one to define regions and the other to approximate the counts in these regions. Our experiments demonstrate that these algorithms are efficient and accurate independent of the size of the original data set, and our analysis explains why this is the case. For the first time, these sample-then-scan algorithms allow spatial scan statistics to run on a million or more data points without making assumptions on the spatial distribution of the data.

Moreover, our experiments and analysis give insight into when it is appropriate to trust the various types of spatial anomalies when the data is modeled as a random sample from a larger but unknown data set.

## 1. INTRODUCTION

Statistical spatial anomaly detection has become an important tool for many problems such as bio-surveillance (detecting disease outbreaks), crowd control, weather monitoring, and pinpointing influential players in a social network. As the scale of the data has grown rapidly many of the standard approaches to these problems have become infeasible.

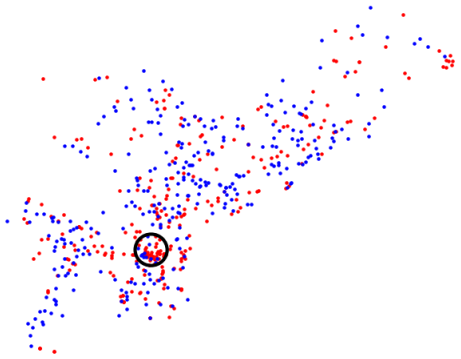

Figure 1: Sampled Philadelphia crime data with thefts as red measured points and a region with high theft circled.

Commonly used adhoc approaches pre-process the data and may affect the underlying statistics in unpredictable ways. These approaches may also restrict the algorithms to run on a subset of the data, again missing existing anomalies.

Another issue often evident in anomaly detection is the multiple comparisons problem [11, 7]. In this scenario, there are many possible hypotheses tested. If any one of them is deemed significant, that hypothesis is reported as anomalous. However, when significance is based on a fixed threshold and fixed data set, then as more and/or richer sets of hypotheses are considered, it is more likely that at least one of these hypotheses will be reported as significant. This issue can be dealt with by adapting the significance threshold to the set of hypotheses considered. However, this is typically overly conservative or adds to the computational complexity of the problem, again limiting the scale to large data. As more studies operate in a "data science" view where large corpuses of data are used for many parallel studies, understanding good practice in these scenarios is of pressing importance.

**Anomaly Detection Pipeline.** In particular, the process of detecting statistically significant spatial anomalies while accounting for multiple comparisons is often broken down into the following three abstract steps.

(S1) Formulate a model of the data and choose a corresponding measure $\phi$ to score the likelihood of an anomaly in a chosen region.

(S2) Scan the data set to find a region $A$ which (approximately) maximizes the measure from (S1).

(S3) Assess whether the score $\phi(A)$ indicates that $A$ is a significant anomaly, either directly raising an alarm, or investigating further.

*Work done while at University of Utah

†Thanks to supported by NSF CCF-1350888, IIS-1251019, CNS-1564287, ACI-1443046, and CNS-1514520.



*SIGSPATIAL'16, October 31-November 03, 2016, Burlingame, CA, USA*



ISBN 978-1-4503-4589-7/16/10...$15.00

DOI: http://dx.doi.org/10.1145/2996913.2996939

The first step (S1) is by now fairly well understood. Kulldorff introduced the Spatial Scan Statistic [13] for Poisson data, which has since been extended to other models by Kulldorff [14] in the extensive SatScan software and more generally by Agarwal *et al.* [2]. There are many recently proposed variants such as the Bayesian [19], expectation-based Poisson [17], and exponential [10] scan statistics.

However, steps (S2) and (S3) are quite time consuming. Often (S2) involves considering *all* possible circular, rectangular, or other geometrically defined regions. Luckily, due to VC-dimension-type arguments, the number of regions with distinct data is typically bounded polynomially in the number of data points (e.g. with $n$ data points, there are $O(n^3)$ circles or $O(n^4)$ rectangles). In some cases, one can ensure that all regions are considered without explicitly measuring $\phi$ in all regions [2]. Another popular approach is to map the data to a discretized grid [18] or a set of pre-defined regions such as counties or zip codes [18]. However, Agarwal *et al.* [1] demonstrated such mappings can introduce large errors due to boundary issues.

Moreover, to avoid dangerously relying on a fixed threshold, (S3) typically involves permutation testing. This involves repeating step (S2) on many random inputs that should not intentionally give rise to a large $\phi(A)$-valued region but might due to peculiarities of randomness. Permutation testing further amplifies the computational bottlenecks in (S2).

**Big Spatial Data.** Despite discretization concerns, many papers have considered data limited to a fixed discretization [18]. In many cases, this limitation is because available data is only available at a certain resolution [22] or because of privacy concerns [12] is only available in that format. However, new data sources are now available at the scale of thousands or even millions of undiscretized spatial data points, and the available super-linear methods are not tractable. For instance, OpenStreet Maps has over 100 million spatially located data points, and Twitter witnesses roughly 400 million tweets per day, many of which include locations. Detecting an anomalous event at this scale which may indicate an interesting social event, pattern, or uprising is infeasible with current statistical anomaly detection approaches.

Furthermore, Agarwal *et al.* [1] proved that sublinear approaches such as streaming cannot provide strong approximation guarantees to the function $\Phi = \max_{A \in \mathcal{A}} \phi(A)$. Thus, standard approaches for large data problems seem hopeless.

Yet, at the same time, it is now common for such data sets to only be available through an already compressed random sample. For instance, one might only access Twitter through a 1% or 10% feed. The yearly American Community Survey (ACS) has replaced the decennial census [4] for most population modeling but samples at a much smaller rate [6]. Moreover, modern big data systems such as BlinkDB [3] or STORM [5] allow fast interactive queries by only making data available through a random sample. In these cases, the "full data set" is actually a random sample of a much larger data set; this property and its approximation implications should be taken into account in any analysis or approach.

## 1.1 Our Approach and Results

We address the scalability of the spatial anomaly detection problem while also considering its effect on the multiple comparisons problem. To do so, we apply data reduction approaches, reducing enormous data sets in size so that less scalable approaches can be applied. We show that despite the lower bounds on the full discrepancy function $\phi$, we can still find many types of spatial anomalies efficiently and robustly. We note that this does not contradict prior space lower bounds (from a streaming context) since we restrict to only consider ranges with a minimum size. Our approach relies on two observations: first, we can approximate the full set of ranges with a much smaller reduced "net" set of ranges, and second, we can approximate $\phi$ in these ranges using random $\varepsilon$-samples.

# 2. PRELIMINARIES

## 2.1 Statistics, Permutation Tests, and Power

**Spatial Scan Statistics.** Consider a data set $X \in \mathbb{R}^2$. Each data point $x \in X$ is given two labels about its baseline value $b(x)$ and its measured value $m(x)$. In the simplest setting, $b(x) = 1$ for all data points (representing the population), and $m(x) \in \{0, 1\}$ and only 1 if it represents some reading that would contribute towards an anomalous event.

Given a region $A \in \mathcal{A}$ where $\mathcal{A} \subset 2^X$, define $b_X(A) = \sum_{x \in A} b(x)/B_X$ and $m_X(A) = \sum_{x \in A} m(x)/M_X$ where $B_X = \sum_{x \in X} b(x)$ and $M_X = \sum_{x \in X} m(x)$. Sometimes for intuition it is nice to attribute $A$ to a subset $\mathbb{R}^2$ (as opposed to combinatorially to a subset of $X$), and often there are restrictions of the subset of $\mathbb{R}^2$ which can define $A$, as in it is a disk or a rectangle.

The Kulldorff scan statistic (or Poisson spatial scan statistic) is defined $\Phi(\mathcal{A}, X) = \max_{A \in \mathcal{A}} \phi_X(A)$ where

$$\phi_X(A) = m_X(A) \ln \frac{m_X(A)}{b_X(A)} + (1 - m_X(A)) \ln \frac{1 - m_X(A)}{1 - b_X(A)}.$$

This will be the default scan statistic for our studies as it is the most common and simplest. It considers the phenomenon that at each location either exists ($m(x) = 1$) or does not ($m(x) = 0$), and in the data as a whole, this phenomenon exists at a fixed rate $p$.

However, via other models of functions $m$ and $b$, one can derive other scan statistics. For instance, see [14, 2] for a discussion including Gaussian, Bernoulli, and Gamma versions. These behave quite similar to $\phi_X$, and a longer version of this paper will extend our analysis to these variants.

**Permutation Tests.** A permutation test randomizes the functions $m$ (and perhaps $b$) while maintaining the aggregate statistics, then recalculates $\Phi$. By repeating this process some number (e.g. $T = 999$ times), we can estimate the fraction of random functions $m$ that would have a $\Phi$ score as high as the input data. Often if the data's $\Phi$ value is larger than 95% of the randomized trials, then we may consider the found region $A$ to be an anomaly. More generally, a practitioner would set a $p$-value and then report the $((1-p)*T)$th largest $\Phi$ value as $\eta_p$, the *significance threshold at level* $p$; the above example has $p = 0.05$ as is most common.

This step calculates a distribution on the values $\Phi$ under random $m$ that otherwise aligns with the input data and then compares the $\Phi$ obtained from the real data to the significance threshold $\eta_p$ of this distribution.

**Power Calculation.** The statistical *power* of a test (such as the 95% threshold test described above) is the empirical probability it rejects the null hypothesis when the null hypothesis is indeed false. To calculate this, we create synthetic data

that has an anomaly and then run any algorithm to detect spatial anomalies on this data. We repeat this experiment several (say 100 times) and report what fraction of the time the algorithm succeeds; this fraction estimates the *power*.

Note one can interpret the power as the true-positive rate. On the other hand, one would want to also estimate the false-negative rate. However, this is precisely the $p$-value, so we can interpret the power as the true-positive rate when the false-negative rate is fixed at the $p$-value.

## 2.2 Random Sampling and Learning Theory

Much is known about the accuracy of randomly sampled data $S \subset X$ with respect to a fixed family of query ranges $\mathcal{A}$. This line of work typically starts with VC-dimension theory [21]. Here a *range space* $(X, \mathcal{A})$ is the family of subsets $Y \subset X$ induced by containment in some range $A \in \mathcal{A}$; that is $\{Y \subset X \mid Y = A \cap X, A \in \mathcal{A}\}$. The *VC-dimension* $\nu$ of a range space $(X, \mathcal{A})$ is the size of the largest subset $Y \subset X$ such that *all* subsets $Z \subset Y$ can be written as $Z = Y \cap A$ for some $A \in \mathcal{A}$. Informally, "well-behaved" range spaces have bounded VC-dimension. For instance, when $X \subset \mathbb{R}^2$, then when $\mathcal{A}$ is defined by all disks $\nu = 3$, and when $\mathcal{A}$ is defined by all axis-aligned rectangles $\nu = 4$. Informally, one can think of the VC-dimension as the number of points or values required to define any particular range $A \in \mathcal{A}$. This corresponds with the property that a range space $(X, \mathcal{A})$ with constant VC-dimension $\nu$ induces at most $|X|^\nu$ distinct subsets of $X$; this polynomial bound is a huge improvement over the worst case $2^{|X|}$ exponential bound. For what follows, we will restrict error parameters $\rho < \varepsilon \leq 1/2$ and $\delta \in (0, 1)$.

An *$\varepsilon$-net* is a subset $S \subset X$ such that for all $A \in \mathcal{A}$ such that $|A \cap X| \geq \varepsilon |X|$ (i.e., the range is large enough) then there must exists some point $x \in Y$ such that $x \in A$, or in other notation $A \cap Y \neq \emptyset$ (i.e., then the range $A$ is "hit" by $Y$). For a range space $(X, \mathcal{A})$ with VC-dimension $\nu$, a random sample $S \subset X$ is an $\varepsilon$-net with probability at least $1 - \delta$ for $|S| = O((d/\varepsilon)\log(1/\varepsilon\delta))$ [9].

An *$\varepsilon$-sample* (or *$\varepsilon$-approximation*) is a subset $S \subset X$ so

$$\max_{A \in \mathcal{A}} \left| \frac{|X \cap A|}{|X|} - \frac{|S \cap A|}{|S|} \right| \leq \varepsilon.$$

That is, for *all* ranges $A \in \mathcal{A}$ the density with respect to $X$ is preserved by $S$ up to an additive error $\varepsilon$. For a range space $(X, \mathcal{A})$ with VC-dimension $\nu$, a random sample $S \subset X$ is an $\varepsilon$-sample with probability at least $1 - \delta$ for $|S| = O((1/\varepsilon^2)(\nu + \log(1/\delta)))$ [21, 15].

A *relative $(\rho, \varepsilon)$-approximation* is a subset of $S \subset X$ so

$$\max_{A \in \mathcal{A}} \left| \frac{|X \cap A|}{|X|} - \frac{|S \cap A|}{|S|} \right| \leq \varepsilon \max\left\{ \rho, \frac{|A \cap X|}{|X|} \right\}.$$

For a range space $(X, \mathcal{A})$ with VC-dimension $\nu$, a random sample $S \subset X$ is a relative $(\rho, \varepsilon)$-approximation with probability at least $1 - \delta$ for $|S| = O((1/\varepsilon^2\rho)(d\log(1/\rho) + \log(1/\delta)))$ [8].

**Takeaways.** These results suggest the following property: much about these query ranges can be preserved as long as one is willing to ignore the error on ranges with few points. This is obvious from the $\rho$ parameter in the relative $(\rho, \varepsilon)$-approximation and also from the meaninglessness of $\varepsilon$-net and $\varepsilon$-sample bounds when $|A \cap X| \leq \varepsilon |X|$.

For all above results, the size of the required sample $|S|$ depends only on the error parameter (and VC-dimension) but *not* on the size of the original data. This implies that these bounds should require a fixed size sample even as the full data set size $|X|$ grows unbounded. In fact, these results all generalize to when $X$ is actually represented by a continuous density function that is essentially infinite. Moreover, the size of the implied minimum query range size (below which the bounds are meaningless) is a fixed fraction of the full data set size, not an absolute fixed size. Under a random sample we should not expect meaningful results on queries below a fixed percentage of the full data set.

Adapting these bounds to understand spatial scan statistics requires more work and is the focus of Section 5.

# 3. SAMPLE-THEN-SCAN ALGORITHMS

The main contribution of our paper is the design, implementation, and analysis of *sample-then-scan* approaches towards computing spatial scan statistics. These algorithms randomly sample the full data set $X$ creating two new data sets. The smaller subset $N \subset X$ of size $n$ serves as a "net" to define a set of regions to scan, so no region on the full data is too far from one "caught" in the net. The larger sample $S \subset X$ of size $s$ is then used to approximate the density of points in each region and to approximate the scan statistic. These two samples are motivated by the $\varepsilon$-net and $\varepsilon$-sample properties discussed in Section 2.2 and validated by our experimental results in Section 4.

This *sample-then-scan* strategy has two main motivations. First, sampling to a much smaller data set obviously and drastically reduces the computational complexity of the algorithms. After sampling, the runtime depends only on the accuracy of the approximation. Second, even a small data set that is not explicitly sampled for the purpose of analysis likely should be modeled as a random sample from some true underlying distribution.

As the analysis in Section 5 will show, it will be important to sample separately from all the points and from those with measured value 1. Otherwise, if there are very few points with measured value 1, we may not obtain enough in our sample. Let $X_m = \{x \in X \mid m(x) = 1\}$ and $X_b = X$. Then we will consider nets $N_m \sim X_a$ and $N_b \sim X_b$ and samples $S_m \sim X_a$ and $S_b \sim X_b$. It will often be convenient to refer to $N = N_m \cup N_b$ and $S = S_m \cup S_b$.

## 3.1 Family of Scan Regions

The next detail of these algorithms is which families of range to consider and how to efficiently enumerate their scan statistic scores to find the maximum one. As with past approaches [1, 18, 13], we focus on either circular or rectangular families of regions.

**Rectangles.** For rectangles $\mathcal{R}$, we use an approach similar to the **Exact** algorithmic approaches in [1] with the added complication of having to evaluate regions defined by $N$ using the larger point set $S$. We refer to this algorithm as allrect. Given a set of $|N| = n$ points and $|S| = s$, allrect scans over all $O(n^4)$ ranges and also aggregates the count of measured and baseline points in each range in $O(n^4 + s\log(n))$ time. We leave the tedious details to the long version.

**More Disks.** For the circular regions, we will consider the set $\mathcal{D}_3$, which contains all combinatorial discs. For a set of points of size $n$, there are $O(n^3)$ such discs. This can be seen by considering any set of points $Y \subset N$ such that there exists a disk $D$ such that all $Y$ are contained in $D$ and no points in $N \setminus Y$ are in $D$. Then we can shrink the disk $D$ to

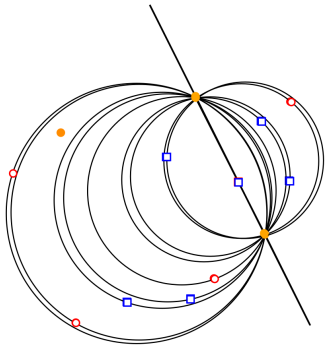

Figure 2: Orange points are in $N$, red points are in $S_b$, and blue points are in $S_m$. The moredisks algorithm splits the points along a hyperplane and then enumerates over a sequence of disks defined by two net points and one point from $S_m \cup S_b$

the minimum disk $D_Y$ so that it contains precisely the same set $Y$. For any such disk $D_Y$, there are generically at most 3 points on its boundary (or if there are more, we can remove all but 3 and the minimum such disk does not change). This quantity 3 is known as the *combinatorial dimension* of $\mathcal{D}_3$. We can thus enumerate all such triples of points and consider the disc with those points on the boundary. Any viable subset $Y \subset N$ is defined by one of these disks.

Enumerating all such disks in $\mathcal{D}_3$ can be done in $O(n^3)$ time, not counting the time to calculate the fraction of measured and background points in each disk. This operation may naively take $O(s)$ time per disk or perhaps $O(\log s)$ using theoretical range counting data structures at the cost of increased space usage and hidden constants. Rather, we describe a direct way to enumerate all such disks while maintaining their contents, described in Algorithm 3.1, very similar to what is used in the SatScan software [14]. The idea is to consider every possible pair of points $p_1, p_2 \in N$ and enumerate all disks which have these two on the boundary. Let $\ell$ be the line through these two points. The largest radius disks containing $p_1$ and $p_2$ on the boundary are in the limit halfspaces with $\ell$ on its boundary. We sweep through all such disks with $p_1$, $p_2$ and one other point $p$ on its boundary with the midpoint of the sweep being a disk with only $p_1 p_2$ defining its diameter; see Figure 2. Given such a pair of points, we can assign a value for each other point $p$ as the signed distance from $\ell$ to the center of the smallest enclosing disk; see Algorithm 3.2. This value corresponds with the disk order. Each point below $\ell$ leaves each disk once and never enters again. Each point above $\ell$ starts outside the disk, enters it once, and then never leaves. Thus, this sweep can be performed on $S$ in $O(s)$ time after sorting all points according to this order in $O(s \log s)$ time. Therefore, without complex range searching, this procedure can be executed in $O(n^2 s \log s)$ time. We call this algorithm moredisks.

We also experiment with a version called alldisks which only considers disks where $p \in N$ which runs in time $O(n^2 s \log n)$.

**Center Disks.** We also develop algorithms that are based on scanning the set $\mathcal{D}_2$, the set of disks which has some point in the data set $S$ as its center. There are only $O(s^2)$ subsets of $S$ (of size $s$) defined by such disks. Each point in the set $S$ defines a center, and expanding outwards from this point defines at most $s$ different disks each containing a set of points. Since there are $s$ possible center points and from each $O(s)$ unique radii, there are only $O(s^2)$ such possible

**Algorithm 3.1** moredisks

```
Scan Statistic Score Φ = 0
for (p1, p2) in N do
  Sort p ∈ S increasing value of Disk_Order(p1, p2, p)
  L = {s ∈ S | s below line going through p1, p2}.
  Set M_D = |L ∩ S_a| and B_D = |L ∩ S_b|
  for p ∈ S in increasing Disk_Order(p1, p2, p) do
    ξ = +1 if p ∈ L and ξ = −1 otherwise
    if (p ∈ S_m) then M_D = M_D + ξ
    if (p ∈ S_b) then B_D = B_D + ξ
    m_A = M_D/|S_m|; b_A = B_D/|S_b|; φ = φ(m_A, b_A)
    if (b_A ∈ [β_ρ, 1 − β_ρ] & m_A ∈ [β'_ρ, 1 − β'_r ho] & φ > Φ)
    then Φ = φ
return Φ
```

**Algorithm 3.2** Disk_Order

```
Input: (p1, p2, p) to assign order of p against p1, p2
Generate center c of disk that goes through p1, p2, and p
u = the normal to the line going through p1 and p2
return ⟨u, (c − (p1+p2)/2)⟩
```

subsets of points. In fact, this intuition implies a way to enumerate all discs and the corresponding counts of points (of $S$) contained within each one (as formalized in Algorithm 3.3). For each point $p_1 \in S$ consider it as a center of a disk. Then sort all other points $p_2 \in S$ in increasing distance $\|p_1 - p_2\|$ from $p_1$. Then scan over these points $p_2$ in sorted order, maintaining the points processed so far as inside the disk; that is we grow the disks outwards from the center. This process takes $O(s^2 \log s)$ time, and is called center.

Another version which considers radii points $p_1 \in N$ (a smaller "net" set of points) is called cNet. The analysis from Section 5 about the covering properties of nets does not apply to $\mathcal{D}_2$ in the same way it does for $\mathcal{D}_3$ and $\mathcal{R}$; therefore cNet and center do not have the same guarantees as moredisks, alldisks, and allrect even though it uses the net in the same way.

In our experiments in Section 4 both of algorithms (cNet and center) perform notably worse in finding the accurate range than the algorithms for $\mathcal{D}_3$ or $\mathcal{R}$; this likely is the result of considering far fewer ranges. However, they are much more efficient for the same number of points.

**Algorithm 3.3** center

```
Scan Statistic Score Φ = 0
for p1 in S do
  Sort p2 ∈ S increasing value ||p1 − p2||
  Set M_D = 0 and B_D = 0
  for p2 ∈ S in increasing ||p1 − p2|| do
    if (p2 ∈ S_a) then M_D = M_D + 1
    if (p2 ∈ S_b) then B_D = B_D + 1
    m_A = M_D/|S_a|; b_A = B_D/|S_b|; φ = φ(m_A, b_A)
    if (b_A ∈ [β_ρ, 1 − β_ρ] & m_A ∈ [β'_ρ, 1 − β'_r ho] & φ > Φ)
    then Φ = φ
return Φ
```

**Error Parameterized Runtime.** Finally, we introduce values of $n$ and $s$ so that we can guarantee additive error on our output $\Phi_{n,s}$. It is not possible to universally guarantee a desired bound of a $\Phi$ (the statistic on the full data) such that

Table 1: Runtime in terms of samples, assuming $n < s$.

| Ranges | Algorithm | Runtime |
|---|---|---|
| $\mathcal{D}_2$ | cNet | $O(ns \log s)$ |
| $\mathcal{D}_2$ | center | $O(s^2 \log s)$ |
| $\mathcal{D}_3$ | alldisks | $O(n^2 s \log n)$ |
| $\mathcal{D}_3$ | moredisks | $O(n^2 s \log s)$ |
| $\mathcal{R}$ | allrect | $O(n^4 + s\log(n))$ |

$\Phi - \varepsilon \leq \Phi_{n,s} \leq \Phi + \varepsilon$ since small ranges (e.g., containing a single point) are very sensitive to sampling error. Previous approximation bounds [2] (approximating $\Phi$, not the data as we do) assumed that each range contains at least an absolute constant number of points. However, it is known under this setting [1] that sampling cannot achieve additive error as desired.

Rather, we make a slightly stronger restriction, parameterizing forbidden ranges by $\rho \in (0, 1)$ and only allowing ranges with $b_A \in [\beta_\rho, 1 - \beta_\rho]$ and $m_A \in [\beta'_\rho, 1 - \beta'_\rho]$ where $\beta_\rho = \rho + \varepsilon$ and $\beta'_\rho = e^{-1/\rho} + \varepsilon$. This is easy to enforce in all of our algorithms, as seen in the psuedocode.

We set $s = O(\frac{1}{(\varepsilon\rho)^2} \log \frac{1}{\delta})$ and $n = O(\frac{1}{\varepsilon\rho} \log \frac{1}{\varepsilon\rho\delta})$ to achieve $\varepsilon\rho$-samples and $\varepsilon\rho$-nets for these large enough ranges with probability at least $1-\delta$. One can typically think of $\log \frac{1}{\delta}$ as a small constant and for intuition just ignore it. Section 5 will provide rigorous upper bounds for the error of our algorithms with these settings. Since these are asymptotic bounds, we will experiment directly with sampling parameters $s$ and $n$.

# 4. EXPERIMENTS

In this section we evaluate the stability of our sample-then-scan framework. We compare the various proposed scan algorithm algorithms against state-of-the-art techniques to evaluate their effectiveness and efficiency. We also explore how various parameter choices affect the performance. This includes parameters of the algorithms (sampling sizes of $n$ and $s$) and parameters of the data (measured rates $p$ and $q$, and planted cluster size $r$). For our comparisons we mainly use synthetic values generated at the real locations of 5 million geolocated tweets.

**Algorithm 4.1** Significance-Test$(X, \sigma, t)$

**for** $i = 1$ to $t$ (# permutations) **do**
  $X_i \sim \gamma$ % $\gamma :=$ null distribution
  $\Phi_i = \mathsf{Alg}(X_i)$
$\mu = \{\Phi_1, \Phi_2, \cdots, \Phi_t\}$
$(\Phi, \mathcal{D}) = \mathsf{Alg}(X)$
**if** $(\Pr[\Phi > \mu]) > 1 - \sigma)$ **then**
  **return** (TRUE, $\mathcal{D}$)
**else return** (FALSE, $\mathcal{D}$)

**Experimental Framework.** To evaluate our algorithms, we generate anomalous regions in several ways and then measure how consistently and efficiently our algorithms can detect them. Detecting an anomalous region has two components. First, the algorithm must return a region $\mathcal{D}$ (as the most anomalous) that is sufficiently close to the planted one. We consider a found region sufficiently close to a planted one if their Jaccard distance is less than $\tau = 0.4$. Our experiments justify this as a reasonable threshold. Second, the algorithm must recognize that the data in the found region occurs with statistically low probability. We consider it significant if its discrepancy score $\Phi(\mathcal{D})$ is larger than all but at most $\sigma = 0.05$ fraction of the most anomalous region found under permutation tests as shown in Algorithm 4.1. The null distribution here uses the same baseline parameters as the input data but without a planted region. The full procedure is outlined in Algorithm 4.2.

Table 2: Default values and range for parameters.

| variable | default | range |
|---|---|---|
| in-region rate: $q$ | .08 | $[.04, .08]$ |
| out-of-region rate: $p$ | .04 | $[.02, .08]$ |
| region size: $r$ | .05 | $[.01, .05]$ |
| net size: $n$ | 100 | $[20, 200]$ |
| sample size: $s$ | 4000 | $[100, 6000]$ |

**Algorithm 4.2** Power-Test$(X, \sigma = 0.05, \tau = 0.4)$

(Success, $\mathcal{D}$) = Significance-Test$(X, \sigma, t = 5000)$
**if** (Success and (Jaccard-Distance$(\mathcal{D}, \mathcal{D}^*) \leq \tau)$) **then**
  **return** TRUE
**else return** FALSE

We experiment using a real data set of $|X| = 5$ million geolocated tweets from North America. The latitude and longitude of these tweets represents the spatial coordinates of the data. Then we synthetically change whether the data points have some measured value. In particular, we plant a circular region from $\mathcal{D}_3$ where the measured points occur more frequently than outside that region. These regions are characterized by three parameters. The value $r \in (0, 1)$ represents the fraction of baseline points in the planted range. Points inside the region are assigned to have the measured effect with rate $q$ and points outside with rate $p$. We attempt to study the algorithms in regions where their accuracy begins to deteriorate. In fact, we observe fairly sharp *phase transitions* between when regions are detected as anomalous or not. The ranges and default values (chosen to highlight these phase transitions) are shown in Table 2. Figure 1 shows an example region and measured points from another data set.

We evaluate the effect of sampling using five of our algorithms (alldisks, center, cNet, moredisks, allrect). We also compare against two existing algorithms for which we could obtain code binaries: agarwal [1](Approx-Extents) and neill [18], which both scan $\mathcal{R}$ or a subset on a grid.

**Finding a Planted Region.** We first experiment by varying one parameter at a time while keeping others at their default. Then we measure the Jaccard distance between the found region and the planted one. We linearly varied 200 different parameter values across the range, and to smooth the noise, we plot the kernel regression of the Jaccard distance. We shade in a region of the average distance to the regressed value in Figure 3 (varying sampling parameters) to show the noise but omit these in Figure 4 (varying the data parameters) to make it less cluttered. Observe that alldisks, moredisks, and allrect consistently converge to a Jaccard distance less than $\tau = 0.4$ as the data signal becomes more clear or the sampling parameters increase. This threshold is drawn as a dashed line. Algorithms cNet and center do not consistently cross this threshold at this value, although center does often. This probably indicates that the window range $\mathcal{D}_2$ is too sparse as

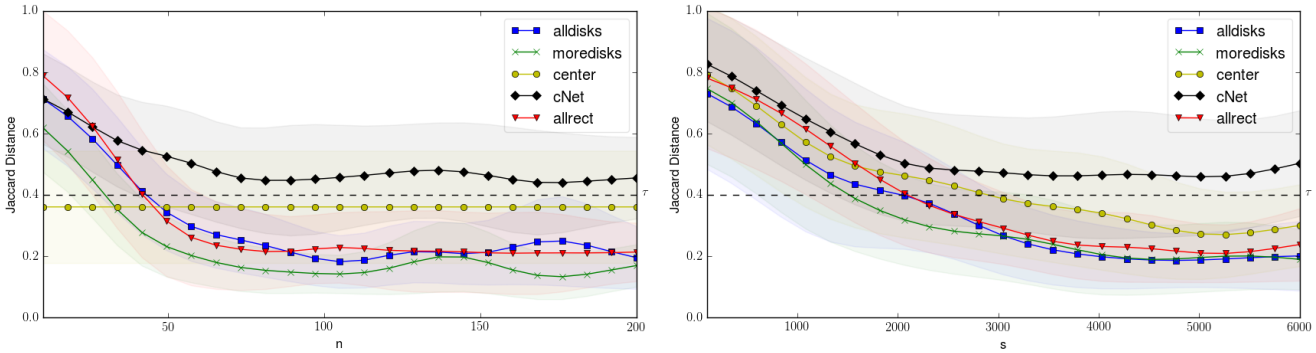


Figure 3: Expected Jaccard distance between found and planted range with default parameters values as $n$ (left) and $s$ (right) vary. Plotted as kernel regression of 200 tested parameter values, with shading at average distance to the interpolated value.

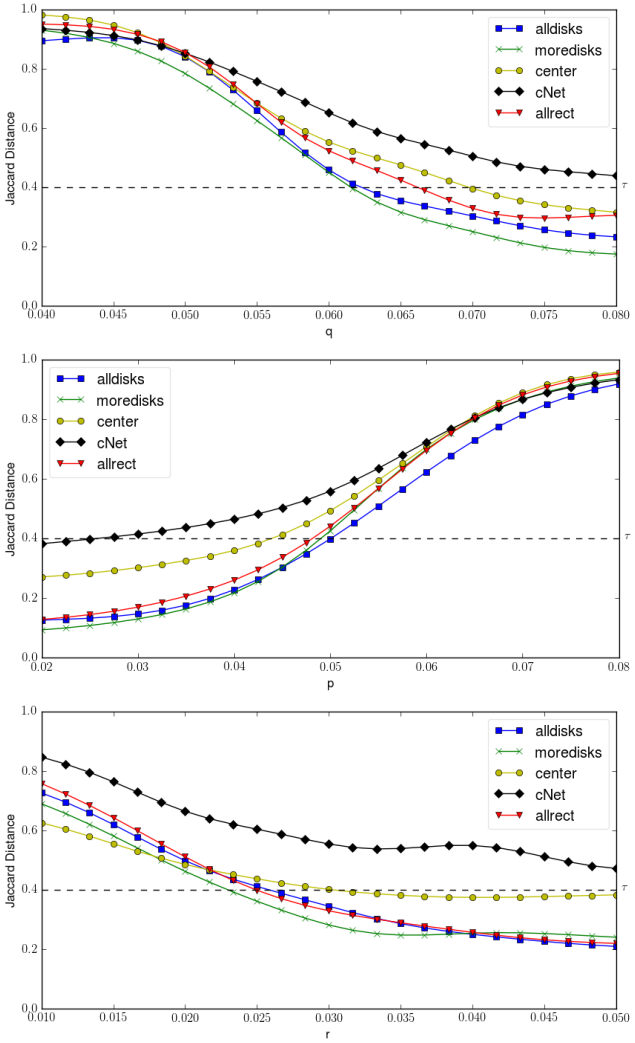


Figure 4: Expected Jaccard distance between found and planted range with default parameters values as we vary $q$ (top), $p$ (middle), and $r$ (bottom). Plotted as kernel regression of 200 tested parameter values.

a set of scan windows; and it is probably not just a mismatch in the shape of the planted range since allrect, which scans $\mathcal{R}$, has error nearly identical to alldisks and moredisks which scan $\mathcal{D}_3$.

We claim it is not necessary (or reasonable to expect) to find the planted region exactly. The optimal boundary is likely to fluctuate through the random process with which the data is generated (or observed). If we do find a close enough region (in this case overlapping most points) further investigation can interactively consider boundary cases [20].

Next observe that as we vary the data parameters, in Figure 4, that as $p$ and $q$ become closer, then all algorithms quickly degrade in their ability to find the planted region. This occurs with $q < 0.06$ (default $p = 0.04$) and with $p > 0.05$ (default $p = 0.08$). This is expected since at these values it is easy for even a single non-planted region to have a high fraction of measured points due to random variation. Also as the region size (measured as an $r$ fraction of points) becomes smaller, the algorithms have a harder time, starting around 2.5% of the data. This is due to random variation which is explained analytically in Section 5, and helps justify not considering regions with less than $\rho$ fraction of the points.

Similarly, Figure 3 shows that as the net size $n$ becomes greater than 50, the algorithms become quite stable (for a default range size of $r = 0.04$). Similarly as the sample size $s$ becomes greater than 3000 or 4000, the algorithms are able to consistently distinguish the planted region from a non-planted one. At the default values ($n = 100$ and $s = 4000$), and beyond, both converge to about a Jaccard distance of only 0.2 (for allrect, moredisks, alldisks), and not better due to boundary conditions in the data generation.

**Significance of the Found Cluster and Power Test.** Next, to completely assess the statistical power of the different algorithms we propose, we need to calculate the probability the correct region is found and deemed significant as in Algorithm 4.2. To do so we use 5000 permutation tests, Jaccard distance threshold $\tau = 0.4$ and a p-value of $\sigma = 0.05$. Figure 6 shows the effect as we vary the in-region rate $q$, the out-of-region rate $p$, and the region size $r$. As $q$ gets significantly larger than $p$, or the region gets large enough that the default difference become significant, then the power goes to 1. That means we always recover the right region *and* declare it significant. On the other hand, when the in-region and out-of-region rates become close, then power degrades. cNet is the most susceptible to this, and then center, stemming almost entirely from their issues in finding the right region as shown in Figure 4. When the region shrinks the

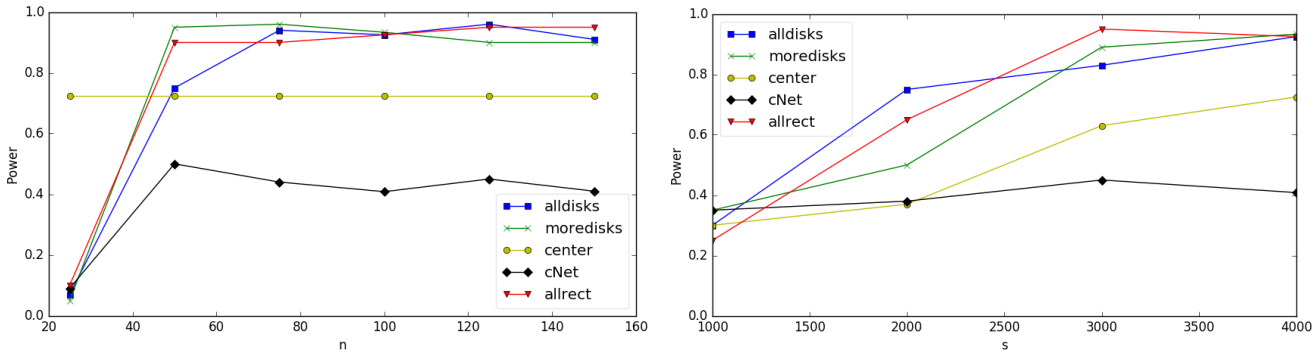


Figure 5: Power of sample-then-scan algorithms (see Algorithm 4.2) as the sampling parameters $n$ (left) and $s$ (right) vary.

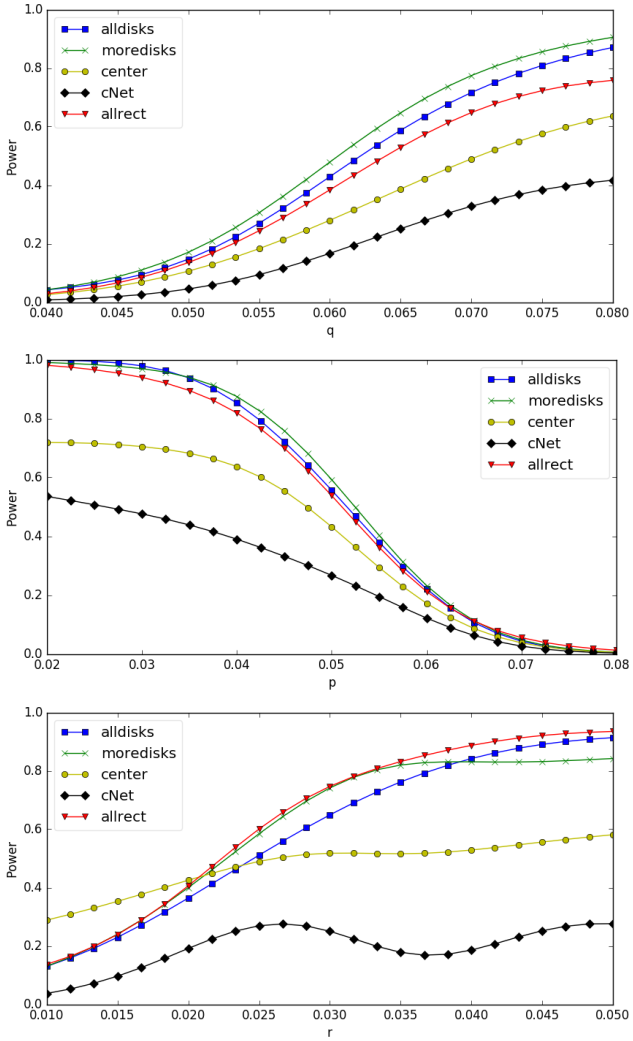


Figure 6: Power of sample-then-scan algorithms (see Algorithm 4.2) as the data parameters $q$ (top), $p$ (middle), and $r$ (bottom) and modified. Plotted as kernel regression of 200 tested parameter values.

power also decreases (as explained by our analysis in Section 5). It does so more rapidly for cNet because it lacks the guarantees that the other algorithms possess, and only scans a net of the smaller set $\mathcal{D}_2$.

Next in Figure 5 we show a line plot of varying the sampling parameters $n$ and $s$ in the algorithms. As $n$ and $s$ increase, the power predictably increases. At a critical point (around $n = 50$ and $s = 3000$) the power levels off indicating that increasing the number of regions by changing $n$ or by increasing the accuracy of the found regions by changing $s$ fails to significantly increase the overall accuracy. For cNet it requires a much larger number of $n$ and $s$ to get to such a state; however note that it has a much smaller runtime dependence in $n$, and actual runtime as we will see next. The techniques are quite resilient in that even with small sample sizes (using the default data settings) the statistical power of the methods with guarantees is always above about 0.9 after some critical threshold.

The curves track closely to the Jaccard distance curves found in Figure 4 and Figure 3. Thus, if the data in the region is anomalous enough to be consistently found, then it is also likely significant in the p-value sense.

**Efficiency.** We first plot our sample-then-scan algorithms in runtime as a function of $n$ and $s$ in Figure 7. More than the asymptotic bounds in Table 1, this shows the actual complexity induced by more complex data manipulation in scanning $\mathcal{D}_3$. Here we see that alldisks and moredisks are much slower than other simpler algorithms. As seen in the gap between moredisks and alldisks, just the constant time step of evaluating $\phi$ may be significant (repeated $s$ times in moredisks and $n$ times in alldisks). We see that cNet is incredibly fast (hard to even see on plot) as its asymptotic runtime would indicate. Also allrect is extremely efficient. A careful implementation and analysis detaches the $O(n^4)$ runtime for scanning $\mathcal{R}$ from the $O(s \log(n))$ runtime to maintaining counts in each range, but it appears the $O(n^4)$ penalty will start becoming problematic if we need $n > 200$.

In testing efficiency we also attempt to compare against some code from existing approaches which do not sample before scanning. This includes the Neill [18], Agarwal [1], and our implementation of SaTScan™ [14]. The Neill approach maps all data to grid cells (at a user defined size), and then finds the most anomalous rectangular union of grid cells using a branch-and-bound search. Given a grid of size $g \times g$, it requires $O(g^2 \log^2 g)$ time, but may fail to find reasonable regions if the grid cells are too large (for instance if the data set is state-wide or nation-wide, but the anomalous region is

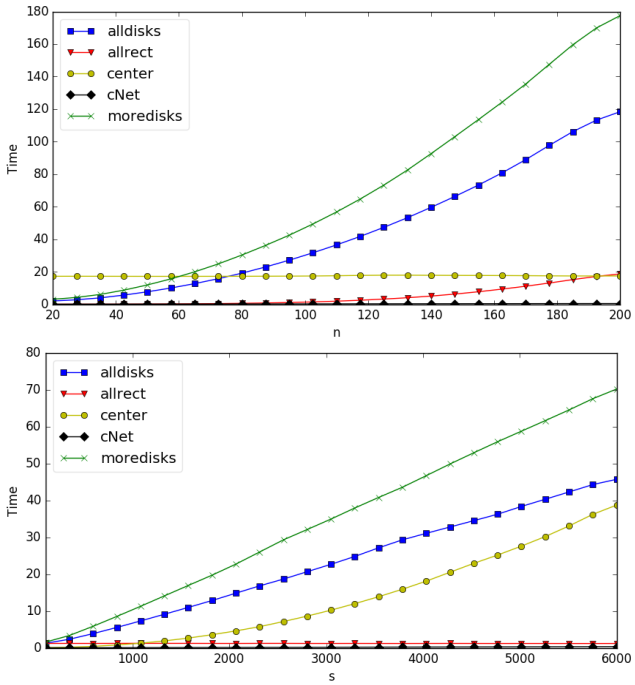


Figure 7: Time in seconds for sample-then-scan algorithms as function of sample parameters $n$ and $s$.

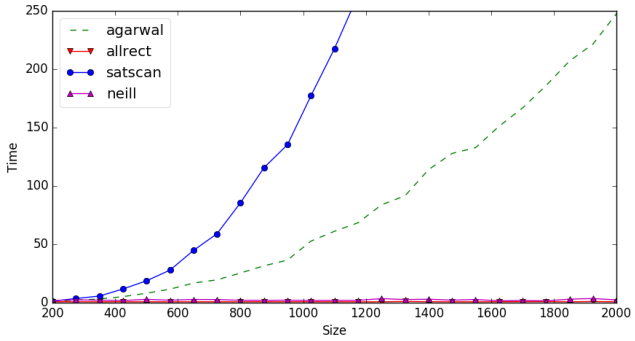


Figure 8: Time in seconds needed to find a region with Jaccard distance less than $\tau = 0.4$.

within only part of a densely populated city) or take at least $O(g^4)$ if certain conditions of the underlying data are not satisfied. The Agarwal approaches also consider rectangles (not necessarily on a grid), and improves beyond the naive approaches by approximating the discrepancy function in a way that allows all rectangles to be scanned faster. The SaTScan$^{\text{Tm}}$ algorithm considers all disks across the dataset. Since both Agarwal and SaTScan$^{\text{Tm}}$ have runtime that scales super-linearly with the data set size they quickly run into issues as the data set size gets large. Since all algorithms have various and different parameters, we provided a best effort to automatically tune the parameters to attain a Jaccard distance of $\tau = 0.4$. To match the parameters we choose $s = \frac{1}{2}n^2$ which is close to what the theory would predict the ratio between n and s should be. This adaptive approach, where we attempt to choose the best parameters for each algorithm was basically necessary to compare against Neill. Their code binary only returns a cluster if it is also deemed significant, and if we set $g$ smaller (more aggressively) then a large fraction of the time it does not return any cluster at all (even with a significant region that our algorithms find).

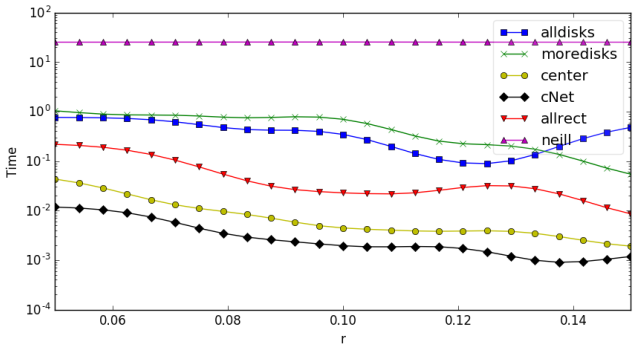


Figure 9: Time in seconds needed to find a region containing $r$ fraction of input points up to Jaccard distance 0.4.

Figure 8 shows that already at $|X| = 2000$, agarwal and SaTScan$^{\text{Tm}}$ are already significantly slower than both neill and allrect, which each scan $\mathcal{R}$. Hence we omit agarwal from further scalability tests. Note at this scale, neill is also slower than allrect since due to its required pre-processing time.

Next we plot the runtime as a function of $r$ in Figure 9, using $|X| = 5{,}000{,}000$. We see all of our algorithms are significantly faster than neill. As $r$ decreases, the sample-then-scan algorithms all take slightly longer (as predicted), but neill is roughly flat since it requires the same grid size $g$ to obtain this error, and since it does not suffer from the same sampling conditions as our algorithms do.

Interestingly, to achieve $\tau = 0.4$ Jaccard distance, cNet and center run significantly faster than the other algorithms. Thus although they perform worse for the same data and sampling parameters, they make up for it in this setting by scanning far fewer regions.

**On Grid Partitioning.** Spatial partitioning methods (e.g., gridding methods like neill) are not guaranteed to work well in situations where the anomalous region is spatially small compared to the entire region. For instance an anomaly (say autotheft crimes) within a densely populated city with a dataset spanning an entire state. A data-adaptive partitioning, such as the one based on our net $N$, will not encounter these same challenges, and could then be used in conjunction with branch-and-bound approaches [18, 23].

# 5. WHY THESE METHODS WORK

To demonstrate that our proposed methods are well-founded requires several steps. First, we need to show that the restricted family of ranges, those induced by our net $N$, gives an additive error bound for all ranges in the full data set.

Second, we need to show that the scan statistic $\phi$ is stable under random sampling; e.g., that our sample set $S$ is large enough to preserve the statistics for large ranges.

Finally, we combine these bound together to assess the accuracy of our procedures and argue that our sample-then-scan procedure is still a valid formulation and properly deals with multiple hypothesis testing.

## 5.1 Coverage Properties of Net Range Spaces

The goal of this section is to explain why and when a smaller set of sampled "net" points $N \subset X$ can be used to define which ranges to scan. Given a full data set $X \in \mathbb{R}^d$ and range space $(X, \mathcal{A})$, let $\mathcal{A}_{|N} = \{A \cap N | A \in \mathcal{A}\}$ be the restriction of $\mathcal{A}$ to the points in $N \subset X$. Recall that an

element $A \in (N, \mathcal{A})$ is a subset of $N$, and such a range does not automatically induce a subset of $X$. Hence, we need to define a geometric mapping $\psi(A) \subset \mathbb{R}^d$ and can then use $\psi(A)$ to define a subset $A' = X \cap \psi(A)$. We say a geometric mapping is *conforming to* $\mathcal{A}$ if for any $N \subset X$ it has the properties (i) any subset $A \in (N, \mathcal{A})$ that $\psi(A) \cap N = A$ and (ii) $\psi(A) \cap X \in (X, \mathcal{A})$. For instance, for $\mathcal{D}_3$ a conforming geometric mapping $\psi(A)$ could be the smallest enclosing disk. A similar mapping exists for $\mathcal{R}$ as the smallest enclosing rectangle, but there is not always one for $\mathcal{D}_2$.

Now we need to show how to construct a set $N \subset X$ and a range space $(N, \mathcal{A}')$ such that for each $\bar{A} \in (X, \mathcal{A})$ there exists a range $A \in (N, \mathcal{A}')$ such that $|\bar{A} \triangle (X \cap \psi(A))| \leq \varepsilon |X|$, and will also require that $X \cap \psi(A) \in (X, \mathcal{A})$. This extra requirement is what makes it difficult to work with $\mathcal{D}_2$.

For this we use symmetric difference range spaces. For a family of ranges $\mathcal{A}$, let $\mathcal{S}_\mathcal{A}$ be the family of ranges made up of the symmetric difference of ranges of the first type. Specifically $\mathcal{S}_\mathcal{A} = \{A_1 \triangle A_2 \mid A_1, A_2 \in \mathcal{A}\}$. If range space $(X, \mathcal{A})$ has VC-dimension $\nu$, then $(X, \mathcal{S}_\mathcal{A})$ has VC-dimension at most $O(\nu \log \nu)$ [16]. Thus for constant $\nu$ (as is the case for $\mathcal{D}_3$ and $\mathcal{R}$) we can use asymptotically the same size random sample as before.

LEMMA 5.1. *Given an $\varepsilon$-net $N$ over $(X, \mathcal{S}_\mathcal{A})$, a geometric mapping $\psi$ conforming to $\mathcal{A}$, then for any range $A \in (X, \mathcal{A})$, there exists a range $\psi(A') \cap X$ for $A' \in (N, \mathcal{A}_{|N})$ such that $|A \triangle (\psi(A') \cap X)| \leq \varepsilon |X|$.*

PROOF. Let $A' = A \cap N$, the part of $A$ in the net $N$, then we have both (i) $\psi(A') \cap N = A'$ and (ii) $\psi(A') \cap X \in (X, \mathcal{A})$, since $\psi$ is confirming. Now since $N$ is an $\varepsilon$-net of $(X, \mathcal{S}_\mathcal{A})$ then we know that if there is no point $x \in N$ in $A \triangle (\psi(A') \cap X)$ (see Figure 10), then $|A \triangle (\psi(A') \cap X)| \leq \varepsilon |X|$, as desired. So to finish the proof, we show that $A \cap N = (\psi(A') \cap X) \cap N$, which implies the condition for the $\varepsilon$-net. Since $N \subset X$, then $(\psi(A') \cap X) \cap N = \psi(A') \cap N = A' = A \cap N$. $\square$

This implies that we can select a large enough random sample $N$ (to satisfy an $\varepsilon$-net for $(X, \mathcal{S}_\mathcal{A})$) and then consider $(X, \mathcal{A}_{N,\psi})$, a subset of all ranges in $(X, \mathcal{A})$, instead of all in $(X, \mathcal{A})$, and incur only $\varepsilon |X|$ absolute counting error. And these correspond with the ranges in $(N, \mathcal{A}_{|N})$ that we scan over in our sample-then-scan algorithms.

Next, since $(X, \mathcal{A}_{N,\psi}) \subset (X, \mathcal{A})$, then an $\varepsilon$-sample of $(X, \mathcal{A})$ will also be an $\varepsilon$-sample of $(X, \mathcal{A}_{N,\psi})$. This implies the following theorem.

THEOREM 5.1. *Consider a range space $(X, \mathcal{A})$ with VC-dimension $\nu$ and conforming geometric mapping $\psi$, and consider random samples of $X$:*

- *$N$ of size $n = O(\frac{\nu \log \nu}{\varepsilon} \log \frac{\nu \log \nu}{\varepsilon \delta})$ and*
- *$S$ of size $s = O(\frac{1}{\varepsilon^2}(\nu + \log \frac{1}{\delta}))$.*

*Then, with probability at least $1 - \delta$, any $A \in (X, \mathcal{A})$ induces a range $A' \in (X, \mathcal{A}_{N,\psi})$ so that $\left| \frac{|A \cap X|}{|X|} - \frac{|A' \cap S|}{|S|} \right| \leq \varepsilon$.*

## 5.2 Stability of Spatial Scan Statistics

Recall that $S_m \subset X$ is a random sample of measured points in $X$ and $S_b \subset X$ is a random sample of the baseline points in $X$. With $X$, $S_m$, and $S_b$ fixed and for a fixed region $A \in \mathcal{A}$ simplify notation as $b_A \triangleq b_X(A)$, $m_A \triangleq m_X(A)$, $\hat{b}_A \triangleq b_{S_b}(A)$, and $\hat{m}_A \triangleq m_{S_m}(A)$. Now our scan statistic becomes

$$\phi(m_A, b_A) = m_A \ln \frac{m_A}{b_A} + (1 - m_A) \ln \frac{1 - m_A}{1 - b_A}.$$

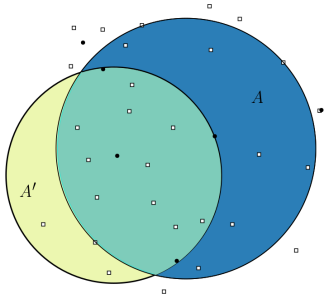


Figure 10: Yellow region is $\psi(A')$, blue region is $\psi(A)$, filled points are in $N$, and empty points are in $X$. The symmetric difference region, $\psi(A) \triangle \psi(A')$, corresponds to the yellow region and the blue region, but excludes the green region. It does not contain any points from $N$ and therefore $|A \triangle (\psi(A') \cap X)| \leq \varepsilon |X|$.

LEMMA 5.2. *If $\hat{m}_A$ and $\hat{b}_A$ are bounded such that for some $\alpha \in (0, 1)$ and $\rho \in (0, 1)$ we have that $\alpha + \varepsilon_1 \leq \hat{m}_A \leq 1 - \alpha - \varepsilon_1$ and $\rho + \varepsilon_2 \leq \hat{b}_A \leq 1 - \rho - \varepsilon_2$ then if:*

$$\varepsilon_1 \geq |m_A - \hat{m}_A| \quad \text{and} \quad \varepsilon_2 \geq |b_A - \hat{b}_A|$$

*we have that*

$$|\phi(m_A, b_A) - \phi(\hat{m}_A, \hat{b}_A)| \leq \varepsilon_1 \ln \frac{1}{\alpha} + \varepsilon_2 \frac{1}{\rho}.$$

PROOF. For simplicity we denote $E_1 = m_A - \hat{m}_A$ and $E_2 = b_A - \hat{b}_A$. The statistic $\phi$ is a convex function and therefore by the first order condition of convexity

$$\phi(m_A, b_A) - \phi(\hat{m}_A, \hat{b}_A) \leq \left\langle \nabla \phi(m_A, b_A), (E_1, E_2) \right\rangle \quad (1)$$

$$\phi(m_A, b_A) - \phi(\hat{m}_A, \hat{b}_A) \geq \left\langle \nabla \phi(\hat{m}_A, \hat{b}_A), (E_1, E_2) \right\rangle, \quad (2)$$

where

$$\nabla \phi(m_A, b_A) = \left( \ln \frac{m_A}{b_A} - \ln \frac{1 - m_A}{1 - b_A}, \frac{b_A - m_A}{(1 - b_A) b_A} \right).$$

The gradient of $\phi$ blows up around the boundary, but since $\alpha + \varepsilon_1 \leq \hat{m}_A \leq 1 - \alpha - \varepsilon_1$ and $\rho + \varepsilon_2 \leq \hat{b}_A \leq 1 - \rho - \varepsilon_2$, then $\alpha \leq m_A \leq 1 - \alpha$ and $\rho \leq b_A \leq 1 - \rho$. Then equation (1) is upper bounded if we choose $E_1$ and $E_2$ to be of opposite sign from the gradient term and of maximum magnitude

$$\left\langle \nabla \phi(\hat{m}_A, \hat{b}_A), (E_1, E_2) \right\rangle \leq \varepsilon_1 \left| \ln \frac{m_A (1 - b_A)}{b_A (1 - m_A)} \right| + \varepsilon_2 \left| \frac{m_A - b_A}{b_A (1 - b_A)} \right|.$$

The first term is maximized by setting $m_A = 1 - \alpha$ and $b_A = \rho$ and the second term is maximized when $m_A = 1 - \alpha$ and $b_A = \rho$ or when $m_A = \alpha$ and $b_A = 1 - \rho$. These two settings are equivalent so we can upper bound the above by

$$\begin{aligned} \varepsilon_1 \ln \frac{(1 - \alpha)(1 - \rho)}{\alpha (1 - \rho)} + \varepsilon_2 \left| \frac{1 - \alpha - \rho}{\rho (1 - \rho)} \right| & \\ &= \varepsilon_1 \ln \frac{(1 - \alpha)}{\alpha} + \varepsilon_2 \frac{1 - \alpha - \rho}{\rho (1 - \rho)} \\ &\leq \varepsilon_1 \ln \frac{1}{\alpha} + \varepsilon_2 \frac{1 - \rho}{\rho (1 - \rho)} \\ &= \varepsilon_1 \ln \frac{1}{\alpha} + \varepsilon_2 \frac{1}{\rho}. \end{aligned}$$

We can repeat the above using equation (2), loosen the constraints and choose $E_1$ and $E_2$ to be of opposite direction to the gradient to get the same bound and therefore

$$|\phi(m_A, b_A) - \phi(\hat{m}_A, \hat{b}_A)| \leq \varepsilon_1 \ln \frac{1}{\alpha} + \varepsilon_2 \frac{1}{\rho}. \quad \square$$

If we set $\alpha = \exp(-1/\rho)$ and $\varepsilon_1 = \varepsilon_2 = \varepsilon\rho/2$, then we obtain $|\phi(m_A, b_A) - \phi(\hat{m}_A, \hat{b}_A)| \leq \varepsilon$.

## 5.3 Combined Statistical Error Bounds

We can replace $\Phi = \max_{A \in \mathcal{A}} \phi(m_{X_m}(A), b_{X_b}(A))$ with a new statistic $\Phi_{n,s} = \max_{A \in \mathcal{A}_{N,\psi}} \phi(m_{S_m}(A), b_{S_b}(A))$ where $N$ and $S$ are random subsets of $S$ of size $n$ and $s$, respectively, both chosen from measured and baseline points separately. Here we will assume both statistics only consider ranges $A$ so $\beta_\rho \leq \frac{|S \cap A|}{|S|} \leq 1 - \beta_\rho$, as enforced by our algorithms.

By Theorem 5.1 we can bound $n = |N|$ and $s = |S|$ to obtain $\varepsilon\rho/2$ error in $\varepsilon_1$ and $\varepsilon_2$, which applies to searching over the restricted "net" range space $(X, \mathcal{A}_{N,\psi})$. Combining this with the stability results for $\phi$ in Lemma 5.2 we can obtain our main result.

THEOREM 5.2. *Consider range space $(X, \mathcal{A})$ with VC-dimension $\nu$ and conforming geometric mapping $\psi$, and parameters $\varepsilon < \rho \leq 1$. Consider random samples of $X$:*

- *$N$ of size $n = O(\frac{\nu \log \nu}{\varepsilon} \log \frac{\nu \log \nu}{\varepsilon\delta})$ and*
- *$S$ of size $s = O(\frac{1}{\varepsilon^2}(\nu + \log \frac{1}{\delta}))$.*

*Then with probability at least $1 - \delta$,*

$$\Phi - \varepsilon \leq \Phi_{n,s} \leq \Phi + \varepsilon.$$

This theorem applies to our sample-then-scan algorithm allrect over $\mathcal{R}$, and algorithms moredisks and alldisks over $\mathcal{D}_3$.

Let $\eta$ be the critical value for the original scan statistic corresponding to a size $|X|$ and significance level $\sigma$, then if

$$\phi(\hat{m}_A, \hat{b}_A) > \eta + \varepsilon,$$

with probability at least $1 - \delta$, we can reject the null hypothesis at significance level $\sigma$. Or, the test on sample data $S$ has critical value $\eta + \varepsilon$ with significance level $\sigma(1 - \delta)$.

Alternatively, as we advocate in Algorithm 4.1, we can estimate the critical value $\eta$ for $\Phi_{n,s}$ directly (by sampling $N$ and $S$ each permutation). This is a valid scan statistic test and its power is well-defined, in particular at the $\sigma = 0.05$ significance level under the method we evaluate it.

# 6. CONCLUSIONS

We introduce the sample-then-scan method for scaling spatial scan statistics to unlimited data sizes. We show both empirically and analytically that these methods are effective, efficient, and have high statistical power. Our method of creating and analyzing nets of ranges to scan is data adaptive, and we believe orthogonal to other efficient branch-and-bound speed ups for these problems. We plan to release a publicly available version of our code on github, and also building a clean simple interface so many practitioners can scale their analysis to much larger sizes.

# 7. REFERENCES


[1] D. Agarwal, A. McGregor, J. M. Phillips, S. Venkatasubramanian, and Z. Zhu. Spatial scan statistics: Approximations and performance study. In *KDD*, 2006.

[2] D. Agarwal, J. M. Phillips, and S. Venkatasubramanian. The hunting of the bump: On maximizing statistical discrepancy. In *SODA*, January 2006.

[3] S. Agarwal, B. Mozafari, A. Panda, H. Milner, S. Madden, and I. Stoica. Blinkdb: Queries with bounded errors and bounded response times on very large data. In *ACM EuroSys*, 2013.

[4] Census. 2010 census by the numbers, 2010. https://www.census.gov/newsroom/releases/pdf/cb10-ffse01.pdf.

[5] R. Christensen, L. Wang, F. Li, K. Yi, J. Tang, and N. Villa. STORM: Spatio-temporal online reasoning and management of large spatio-temporal data. In *SIGMOD*, 2015.

[6] E. Gardner, T. Kimpel, and Y. Zhao. American community survey user guide, 2015.

[7] A. Gelman and E. Lokken. The garden of forking paths: Why multiple comparisons can be a problem, even when there is no "fishing expedition" or "p-hacking" and the research hypothesis was posited ahead of time. http://www.stat.columbia.edu/˜gelman/research/unpublished/p_hacking.pdf, 2014.

[8] S. Har-Peled and M. Sharir. Relative (p,eps)-approximations in geometry. *Discrete & Computational Geometry*, 45:462–496, 2011.

[9] D. Haussler and E. Welzl. epsilon-nets and simplex range queries. *Discrete and Computational Geometry*, 2:127–151, 1987.

[10] L. Huang, M. Kulldorff, and D. Gregorio. A spatial scan statistic for survival data. *BioMetrics*, 63:109–118, 2007.

[11] J. P. A. Ioannidis. Why most published research findings are false. *PLoS Medicine*, 2:124, 2005.

[12] J. Krumm. Inference attacks on location tracks. In *ICPC*, 2007.

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

[14] M. Kulldorff. *SatScan User Guide*. http://www.satscan.org/, 7.0 edition, 2006.

[15] Y. Li, P. M. Long, and A. Srinivasan. Improved bounds on the samples complexity of learning. *J. Comp. and Sys. Sci.*, 62:516–527, 2001.

[16] J. Matoušek. *Lectures in Discrete Geometry*. (Graduate Texts in Mathematics). Springer, 2002.

[17] D. B. Neill. Expectation-based scan statistics for monitoring spatial time series data. *International Journal of Forecasting*, 25:498–517, 2009.

[18] D. B. Neill and A. W. Moore. Rapid detection of significant spatial clusters. In *KDD*, 2004.

[19] D. B. Neill, A. W. Moore, and G. F. Cooper. A Bayesian spatial scan statistic. In *NIPS*, 2006.

[20] T. Tango and K. Takahashi. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 4(1):1–15, 2005.

[21] V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theo. of Prob and App*, 16:264–280, 1971.

[22] C. E. Woodcock and A. H. Strahler. The factor of scale in remote sensing. *Remote Sensing of Environment*, 21:311–332, 1987.

[23] M. Wu, X. Song, C. Jermaine, S. Ranka, and J. Gums. A LRT framework for fast spatial anomaly detection. In *KDD*, 2009.