SCALABLE SPATIAL SCAN STATISTICS

by

Raghvendra Singh

A thesis submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computing

School of Computing

The University of Utah

December 2015

The University of Utah Graduate School

STATEMENT OF THESIS APPROVAL

The thesis of **Raghvendra Singh**

has been approved by the following supervisory committee members:

Jeffrey Phillips , Chair		**07/31/2015**
		Date Approved
Suresh Venkatasubramanian , Member		**07/21/2015**
		Date Approved
Vivek Srikumar , Member		**07/21/2015**
		Date Approved

and by **Ross T. Whitaker** , Chair of

the **School of Computing** and by

David B. Kieda , Dean of The Graduate School.

ABSTRACT

We present algorithms for detecting spatial anomaly in a time efficient manner. There are many other approaches to solve the same problem but they face a serious issue of very huge computational time. We came up with some novel algorithms which help us to solve the problem in a time efficient manner for very large data sets. We tried to show, by executing experiments on both synthetic and real world data set, that the results obtained from the original data set and the sampled data set are very similar and therefore we executed all our approaches on sampled data set rather than on the original data set. Thus we saved a lot of computational time by using sampled data set as an input to our approaches.

CONTENTS

ACKNOWLEDGMENTS

I am extremely thankful and indebted to my thesis advisor, Prof. Jeff M. Phillips, for sharing expertise, sincere and valuable guidance and encouragement extended to me. He helped me to have a clear vision towards my research and guided me towards the completion of my thesis. His expert advice and support throughout my master's program will always be appreciated.

I would like to thank my committee member Prof. Suresh Venkatasubramanian for his clustering class which has totally changed my perception towards clustering. This class helped me to understand that there is much more than conventional methods like K-means and hierarchical clustering.

I would also like to thank my committee member Prof. Vivek Srikumar for his precious time and advice on some of the very cool research areas like Machine Learning and Natural Language Processing which helped me to solve some of the problems in my thesis.

Last but not least, it was a wonderful experience to work with my advisor and committee members and the knowledge which I received during my master's program will be very helpful for my future success.

CHAPTER 1

INTRODUCTION

Statistical spatial anomaly detection has become an important tool for many problems ranging from biosurveillance (detecting disease outbreaks) [1, 2] to crowd control [3] to weather monitoring [4] to pinpointing influential players in a social network [5]. But as these topics have become more pertinent, the scale of the data has grown rapidly, making many of the standard approaches to these problems infeasible. This has led to either ad hoc approaches which preprocess the data in ways which may affect the underlying statistics in unpredictable ways, or restrict any algorithm to run on a subset of the data, again missing out on the anomalies sought.

Another issue often quite evident in anomaly detection is the multiple comparisons problem. In this scenario, there are many possible hypotheses tested, and if any one of them is deemed significant (e.g., an anomaly is found), it is reported as such. However, if such a significance criteria is based on a fixed threshold, then as more or richer sets of hypothesis are considered, it is more likely that some will be reported even with a fixed data set, and even if there is no underlying structure. The way around such an approach is to adapt the significance threshold to the set of hypothesis considered. However, yet again, this typically adds to the computational complexity of the problem, again limiting how this can scale to large data.

1.1 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 measure ϕ to score the likelihood of an anomaly in a chosen region.

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

(S3) Assess whether the score $\phi(C)$ indicates that C is a significant anomaly, either directly raising an alarm, or investigating further [6].

The first step (S1) is by now fairly well understood. Kulldorff introduced the Spatial Scan Statistic [7] for Poisson data, and this has since been extended to other models by Kulldorff [8] in the extensive SatScan software, and more generally by Agarwal et al. [9]. There are many recently proposed variants such as the expectation-based Poisson [10], Gaussian [11] and exponential [1] 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 ϕ in all regions [9]. Another popular approach is to map the data to a discretized grid [12] or a set of predefined regions such as counties or zip codes [12]. However, Agarwal et al. [13] demonstrated such mappings can introduce large errors due to boundary issues.

Moreover (S3), when dangerously relying on a fixed threshold, typically involves permutation testing [2]. That is repeating step (S2) on many random inputs that should not intentionally give rise to a large valued $\phi(C)$ region, but might be due to peculiarities of randomness. This further amplifies the efficiency bottlenecks present in (S2).

1.2 Big Spatial Data

Despite discretization concerns, many papers have considered data limited to a fixed discretization [12]. In many cases this is because available data due to collection resolution [14] or privacy concerns [15, 16] 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 geolocation encodings. Detecting an anomalous event, perhaps indicating an interesting social event or uprising, is infeasible with current statistical anomaly detection approaches.

Furthermore, Agarwal et al. [13] provided a proof that sublinear approaches such as streaming cannot provide strong approximation guarantees to the function $\Phi = \max_C \phi(C)$. So standard approaches for large data problems seem hopeless.

1.3 Dangers of Multiple Comparisons Testing

Step (S2) is a classic example of multiple comparisons testing, in that by design, it considers every possible region as a potential anomaly. Most spatial anomaly approaches which may avoid such an explicit scan to find some interesting region may, however, also be considering such a scan implicitly or approximately. Basically, unless the study starts with some prespecified zone which one suspects to be the location of an anomaly *without considering the data*, then it is by default considering many such regions.

Moreover, several recent studies [17, 18] have documented the dangers of the multiple comparisons problem: how it can result in false discovery, and its relationship with the noise in the data. As science in general is shifting towards a data science view where large corpuses of data are used for many parallel studies, understanding in this area is of pressing importance.

1.4 Our Approach and Results

We address the scalability of the spatial anomaly detection problem while also considering its effect on avoiding a problem based on multiple comparisons. We do so by considering data reduction approaches. Basically, starting with an enormous data set, can we reduce it in size so that well-studied but less scalable approaches can be applied. Can this, and how does this, affect the certainty with which we can detect outliers? Can this be used within a scalable and efficient anomaly detection pipeline?

We show that *yes*, we can still find many types of spatial anomalies, efficiently and robustly. However, through novel analysis and experimentation, we also show that as data size grows, certain types of spatial anomalies which might have been detectable at small scale, are no longer possible at large scale. This is not just a computational effect either. As the number of data points increases, so does the number of scanning windows and hence the likelihood of rare event also increases, thus incorrectly rejecting the null hypothesis indicating the presence of a cluster when actually there is none. The presence of such a cluster is no longer a reliable indicator of a true underlying phenomenon. This is exactly an instance of a multiple testing problem. We note that this does not contradict prior space lower bounds (from a streaming context) since we analyze the *difference* in measures ϕ, not just their values, and we restrict our study to more realistic data settings.

CHAPTER 2

BACKGROUND

In this section we review the general pipeline needed to use statistical spatial anomaly detection.

2.1 Scan Statistics

Consider a data set $X \in \mathbb{R}^d$. 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, and $m(x) \in \{0, 1\}$, and only 1 if it represents some reading that would contribute towards an anomalous event.

Given a region $C \subset X$ define $b_X(C) = \sum_{x \in C} b(x)/B_X$ and $m_X(C) = \sum_{x \in C} m(x)/M_X$ 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 C to a subset $\mathbb{R}^d$ (as opposed to combinatorially to a subset of X), and often there are restrictions on the subset of $\mathbb{R}^d$ which can define C, as in it is a disk or a rectangle?

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

$\phi_X(C) = m_X(C) \log \frac{m_X(C)}{b_X(C)} + (1 - m_X(C)) \log \frac{1 - m_X(C)}{1 - b_X(C)}$.

$\phi_X(C)$ is also called the discrepancy score of the region C.

2.2 Permutation Tests

A permutation test randomizes the functions m and perhaps b, then recalculates Φ. By repeating this process some number (e.g., 99 times) then we can estimate the fraction of random functions m that would have a Φ score as high as the input data. Often if the data's Φ value is larger than 95% (or for some p-value other than $p = 0.05$) of the randomized trials, then we may consider the found region C to be an anomaly. The underlying goal of this step is to calculate a distribution on the values Φ under random m that otherwise aligns with the input data, and then compare the Φ obtained from the real data to this distribution.

2.3 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.

There is another way to determine the performance of a cluster detection algorithm by calculating the statistical measures like Sensitivity and Specificity. In our setting sensitivity directly corresponds to power which gives fraction of times an algorithm is able to detect a cluster at 95 percentile significance level when actually there is a cluster present inside the data set. By keeping the significance level at the 95th percentile of the distribution of data under null hypothesis we are in a way fixing the specificity also. Specificity gives the fraction of times an algorithm detects a cluster when actually there is no cluster in the data set. Since we fixed the significance level at the 95th percentile we can say that there is 5 % chance of detecting a cluster when actually there is none in the data set.

CHAPTER 3

OUR APPROACHES

For all the following three approaches, we sample $S \subset X$ points, where $|S| = \frac{1}{\epsilon^2}(v + \log \frac{1}{\delta})$ and v is the VC dimension of the the region which is a circle in our approaches and hence $v = 3$. Sample set size S works mainly for numerical discrepancy function [13] but can be transferred to Poisson discrepancy in bounded range. However, outside this bounded range sample set size S does not work.

3.1 Sample and Run

We use only the regions defined by these S points. In particular, we only consider the set of regions $\mathcal{D}_S$ as disks D centered at some $c_D \in S$ and radius defined as $r_D = \|c_D - s\|$ for another $s \in S$. Since every $D \in \mathcal{D}_S$ is uniquely defined by two points from S, there are at most $O(s^2)$ such regions. Furthermore we can calculate $m(D)$ and $b(D)$ over S for each region in $O(s^2 \log s)$ time. For each $s_1 \in S$, we consider it in turn as a center point. The we calculate and sort all other points $s_2 \in S$ in order of distance from s_1. Let $B(D) = \sum_{x \in D \cap S} b(x)$ and $M(D) = \sum_{x \in D \cap S} m(x)$. Starting with $r_D = \|s_1 - s_1\| = 0$, the $b(D) = b(s_1)$ and $m(B) = m(s_1)$. Then these values are incremented as we scan over the other points in S in sorted order from s_1. At each step $m(D)$ and $b(D)$ can be derived as $M(D)/M$ and $B(D)/B$, respectively, and $\phi(D)$ can be calculated in constant time.

We set $\Phi = \max_{D \in \mathcal{D}_S} \phi(D)$, run permutation tests, and return D if it has a p-value less than 0.05.

3.2 Sample Data to Limit Ranges

In this approach we will only focus on the red points because the density of red points in the cluster should be more as compared to outside and hence there are high chances of finding a cluster near red points. The radius is considered as a distance between red point and other red points because if we had considered the radius as a distance between red point and blue point then we would have added one more blue point in the region and

therefore the density of red points inside the region will be less. Let R_p be a set of all the red points in sampled data. We consider every red point as center and consider only those circles as valid ranges for which the radius is equal to distance between the center and some other red point. In particular, we only consider the set of regions $\mathcal{D}_{R_p}$ as disks D centered at some $c_D \in R_p$ and radius defined as $r_D = \|c_D - r\|$ for another $r \in R_p$. Since every $D \in \mathcal{D}_{R_p}$ is uniquely defined by two points from R_p, there are at most $O(r^2)$ such regions. Thus by reducing the number of ranges from $O(s^2)$(in first approach) to $O(r^2)$, $r \in R_p$, the time complexity of this approach is greatly reduced. Time complexity for this approach is $O(rs\log(s))$ where r is the total number of the red points in the sampled data set and s is the size of the sampled data set. For every red point $r \in R_p$ we sort all other points in order of distance from r. We then find the discrepancy score $\phi(D)$ for each circular region D in $\mathcal{D}_{R_p}$ and find the maximum discrepancy score Φ where $\Phi = \max_{D \in \mathcal{D}_{R_p}} \phi(D)$. The circular region with discrepancy score Φ will be the most anomalous region and then we can check the statistical significance of this circular region by calculating the statistical power. If $power > 0.95$ then we can say that the region detected is the most anomalous region.

3.3 Limit Ranges to Neighborhoods

This approach is first trying to find a red point in the cluster region and then applying the second approach to this red point only. The main idea is that if we choose sufficiently large value of K then the ratio of number of red to blue point in the neighborhood set is maximized for a point in cluster region only.

Let R_p be a set of all the red points in sampled data S. Let $R_p \in S$ be a set of all the red points. We consider every red point $r \in R_p$ as the center and a neighborhood set containing k nearest points to r. For every red point r as the center we then calculate α which is equal to the ratio of number of red points and number of blue points in the neighborhood set. We consider that red point as a candidate point whose α value is maximum. We then start with candidate point as the center and consider all the circular regions with radius as distance between candidate point and other red points. In this approach we have only one red point as a candidate point and hence the number of circular regions are reduced to $O(r)$. We then find the discrepancy score $\phi(D)$ for the circular region $D \in \mathcal{D}_{R_p}$ for which the candidate point is the center then. We can check the statistical significance of this circular region by calculating the statistical power. If $power > 0.95$ then we can say that the region detected is the most anomalous region. The time complexity of this algorithm is $O(rsk + s\log(s))$ where r is total number of red points in sampled data set and s is size of sampled data

set. $O(rsk)$ is to find a neighbourhood set for each red point $r \in R_p$ and $O(s\log(s))$ is for finding the discrepancy score of the most anomalous region.

CHAPTER 4

EXPERIMENTS

We executed our algorithm 1000 times on a uniformly distributed data set of size 40,000 with p = 0.1 as fraction of red points. This will give us 1000 max discrepancy scores. We then create a distribution of these 1000 values and calculate the 95th percentile value of the discrepancy score which will act as a threshold. We then took the original input data size=40,000 of uniformly distributed points in a range. We then implanted a cluster in the data set. The cluster size is fixed for all the experiments. Fraction of red points inside the cluster is q=0.2 and fraction of red points in the entire original data set is p = 0.1. We will then execute our algorithm 100 times. This will give us 100 max discrepancy scores. Let m be the number of discrepancy scores out of 100 which are greater than the threshold. Then $power = \frac{m}{100}$. We performed a few experiments to check how the statistical power of a particular approach changes with sample size. The results are shown in Figure 4.1.

We also did some experiments to see which approach takes less time to achieve the same power. Figure 4.2 gives the power vs time taken graph for all the three approaches. Time is measured in seconds and sample size set contains sizes in multiple of 100 from 100 to 2800 data points. We then performed some experiments for showing how power varies with cluster size, fraction of red points in the sampled data set and fraction of red points inside the cluster. The corresponding results are shown in Figure 4.3, Figure 4.4 and Figure 4.5, respectively. In Figure 4.4 and Figure 4.5 there is a dip at p=0.25 and q=0.1, respectively, because in the default setting we have p=0.1 and q=0.25 and therefore we are observing a dip in these two figures because p and q become very close and hence the discrepancy score gets very low. Table 4.1 compares the run time of different approaches we came up with in the thesis. We experimented with everything in the default setting which is as follows: a)Fraction of red points in the cluster denoted by q = 0.25. b)Fraction of red points in the entire original data set denoted by p = 0.1. c)Sample size = 2500. d)Cluster size denoted by c = 0.1 % of original data set. e)Original data size = 40000.

We also compared our results with the results produced by software written by Daniel

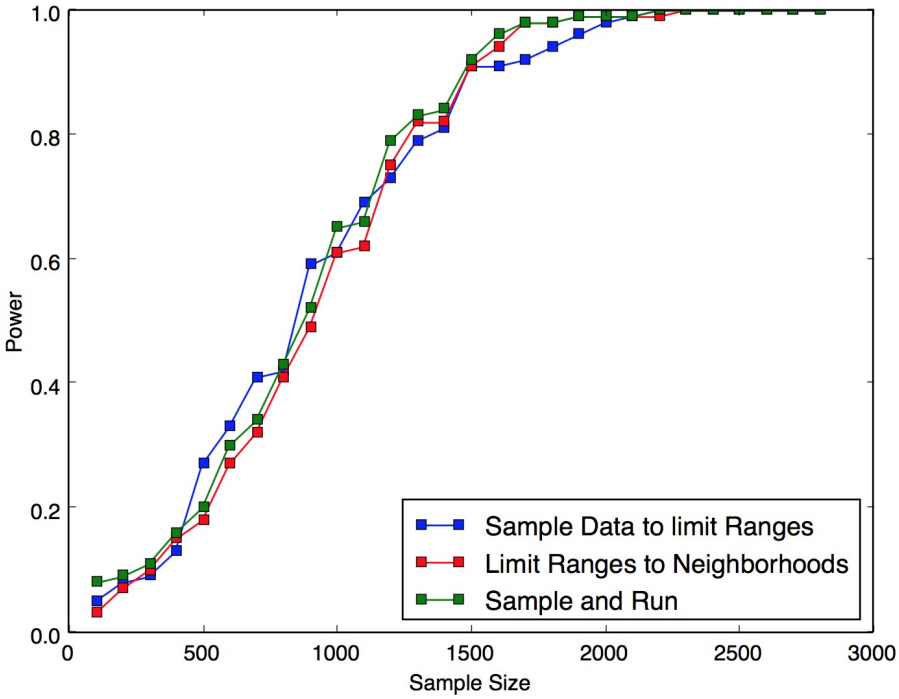

Figure 4.1: Graph of Power vs Sample size for all the approaches

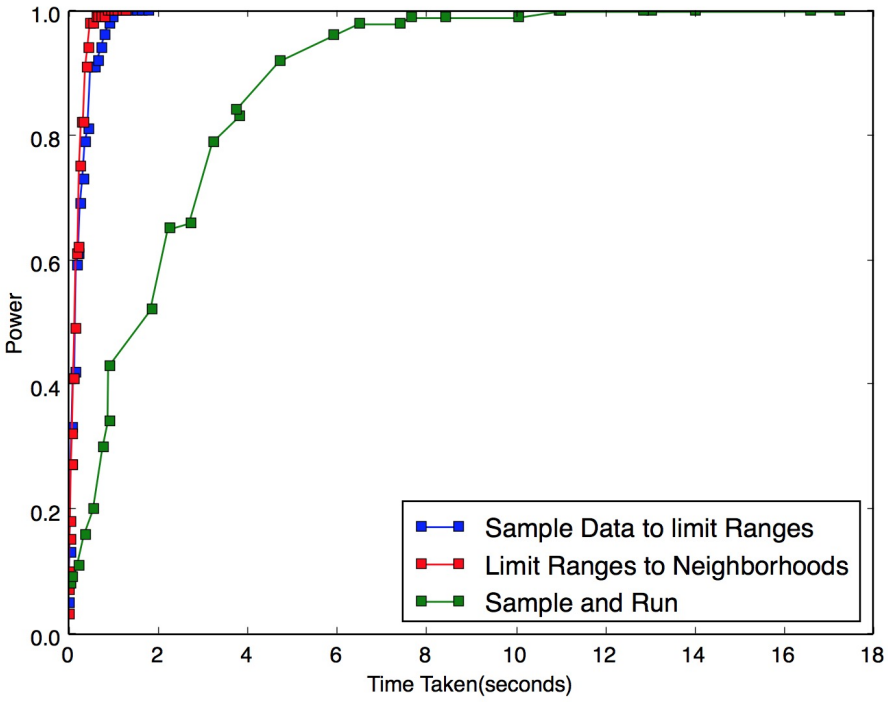

Figure 4.2: Graph of power vs Time taken(seconds) for all the approaches

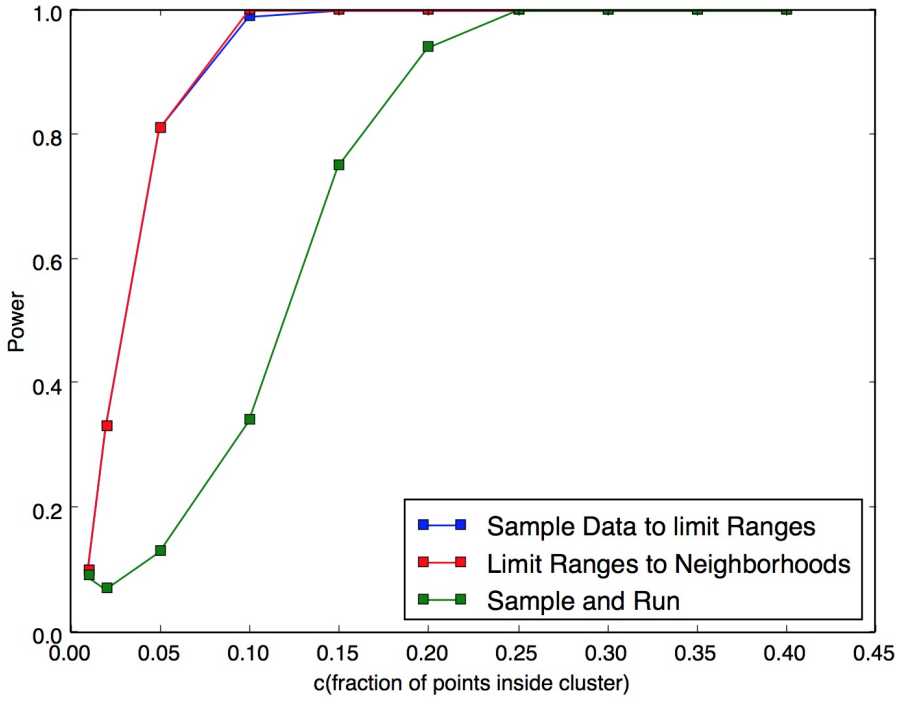

Figure 4.3: Graph of Power vs fraction of points inside the cluster for all the approaches

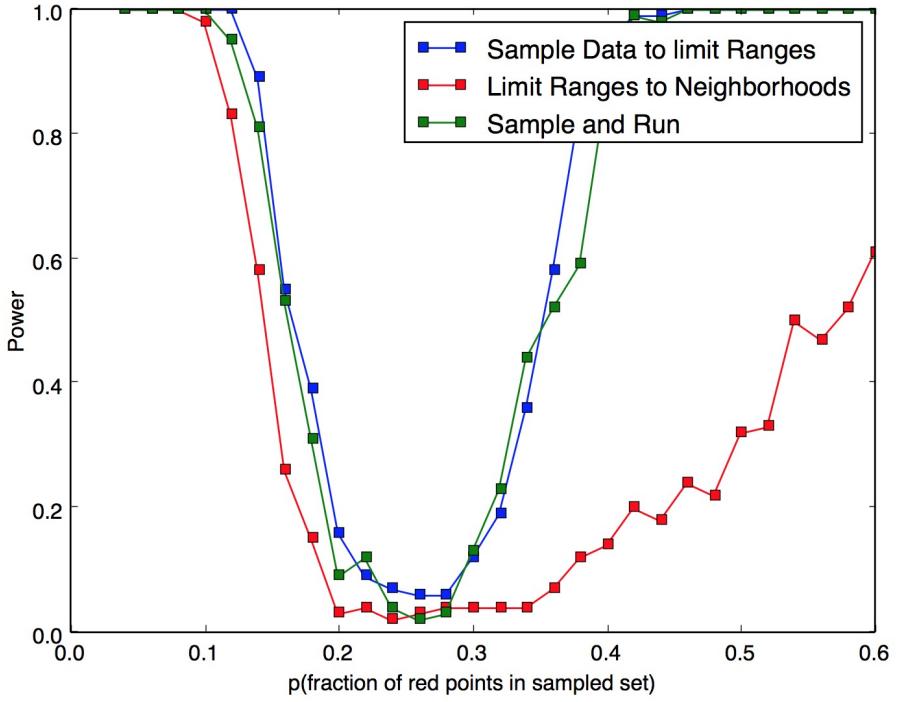

Figure 4.4: Graph of Power vs fraction of red points in sampled set for all the approaches

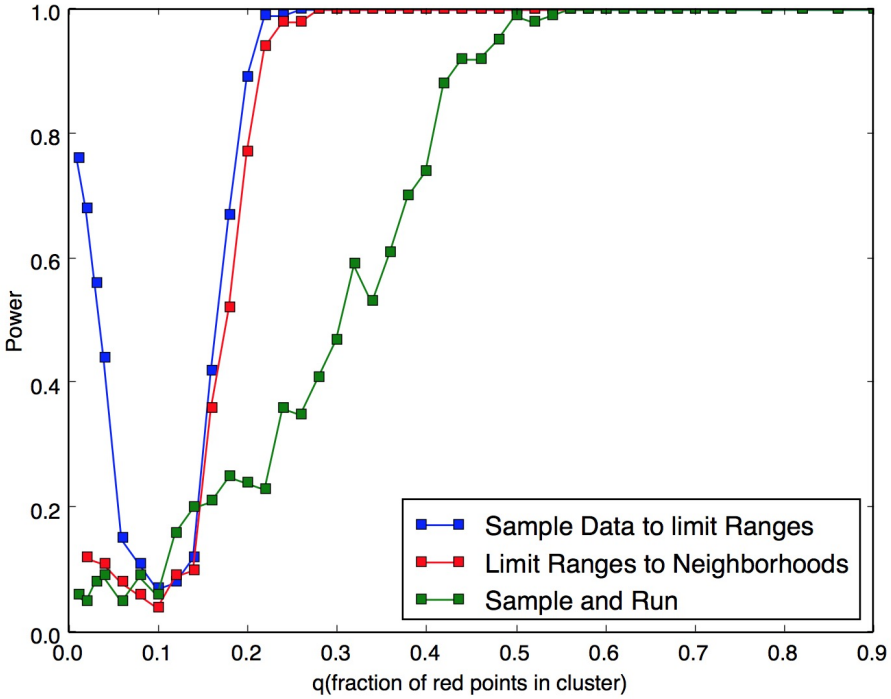

Figure 4.5: Graph of Power vs fraction of red points in the cluster for all the approaches

Table 4.1: Comparison of run time of all the three approaches for default setting.

Approach	Run Time(seconds)
Limit Ranges to Neighborhoods	0.69
Sample Data to Limit Ranges	1.13
Sample and Run	11.71
Run Without Sampling	5130.91

B. Neill and Andrew W. Moore. Their software is built on the algorithm given in [12]. Their algorithm tries to find the rectangular region with the highest density and calculate its significance by randomization. The algorithm considers an $N \times N$ grid of squares, where each square has a count c_{ij} and an underlying population p_{ij}. Their algorithm partitions the grid into overlapping regions using an overlap-kd tree data structure which basically prunes regions which cannot contain the maximum density region. Their algorithm takes $O((N \log N)^2)$ to detect a cluster provided the regions are sufficiently dense. For more accurate results the number of cells in both x and y dimension of the grid should be sufficiently large but this will have an adverse affect on the time complexity. Therefore, in order to detect a cluster more accurately their algorithm needs a greater number of cells in the grid and also the regions should be sufficiently dense whereas there are no such requirements in our algorithms.

The comparison of results produced by executing our algorithms and the Neill et al. [12] algorithm on the Road data set is given in Table 4.2. The Entire Road data set is present in a $[-96.7956289, -90.14749] \times [40.29526, 43.67256]$ rectangular box. The Road data set contains more than a million data points with default parameter settings. Each data point represents the latitude and longitude of a position on a particular road. We have used the latitude and longitude of a point as x and y coordinate, respectively.

We then compared our results with the results in Agarwal et al. [13]. Agarwal et al. try to approximate the discrepancy measure by linear functions. They also executed their algorithms on gridded regions. We compared our results with the results produced by Approx Grid algorithm in Agarwal et al. [13] for different error rates ϵ. The comparison is shown in Table 4.2. We can observe that for Agarwal et al. [13] as the grid size decreases and epsilon value increases the position of the cluster becomes less accurate. Even when the grid size is 128 for epsilon value 0.1 the cluster is not determined as accurate as compared to the Neill et al. [12] algorithm and our.

Figure 4.6 and Figure 4.7 represent the true cluster, observed cluster detected by

Table 4.2: Comparison of results between Neill et al. [12] and our algorithm on Road data set. Figure 4.6 and Figure 4.7 shows the position of true and observed cluster.

	Obs. Cluster Center	True Cluster Center	Run Time(sec)
Neill et al. [12]	(-94.82,41.65)	(-94.85,41.65)	27.200 [128]
Neill et al. [12]	(-94.87,41.66)	(-94.85,41.65)	13.419[64]
Neill et al. [12]	(-94.92,41.61)	(-94.85,41.65)	8.575 [32]
Agarwal et al. [13]	(-95.28,41.455)	(-94.85,41.65)	151.348[128] ($\epsilon = 0.1$)
Agarwal et al. [13]	(-95.28,41.455)	(-94.85,41.65)	46.103[128] ($\epsilon = 1$)
Agarwal et al. [13]	(-95.28,41.63)	(-94.85,41.65)	20.157[128] ($\epsilon = 5$)
Agarwal et al. [13]	(-95.28,41.455)	(-94.85,41.65)	14.233[128] ($\epsilon = 10$)
Agarwal et al. [13]	(-95.28,41.285)	(-94.85,41.65)	17.634[64] ($\epsilon = 0.1$)
Agarwal et al. [13]	(-95.28,41.97)	(-94.85,41.65)	5.331[64] ($\epsilon = 1$)
Agarwal et al. [13]	(-95.28,41.97)	(-94.85,41.65)	2.353[64] ($\epsilon = 5$)
Agarwal et al. [13]	(-92.26,41.285)	(-94.85,41.65)	1.693[64] ($\epsilon = 10$)
Agarwal et al. [13]	(-96.8,40.94)	(-94.85,41.65)	1.819[32] ($\epsilon = 0.1$)
Agarwal et al. [13]	(-96.80,0)	(-94.85,41.65)	0.541[32] ($\epsilon = 1$)
Agarwal et al. [13]	(-96.8,40.94)	(-94.85,41.65)	0.249[32] ($\epsilon = 5$)
Agarwal et al. [13]	(-92.26,40.26)	(-94.85,41.65)	0.169[32] ($\epsilon = 10$)
Approach 3.2	(-94.83,41.65)	(-94.85,41.65)	3.42
Approach 3.3	(-94.87, 41.65)	(-94.85,41.65)	1.15 [k=50]

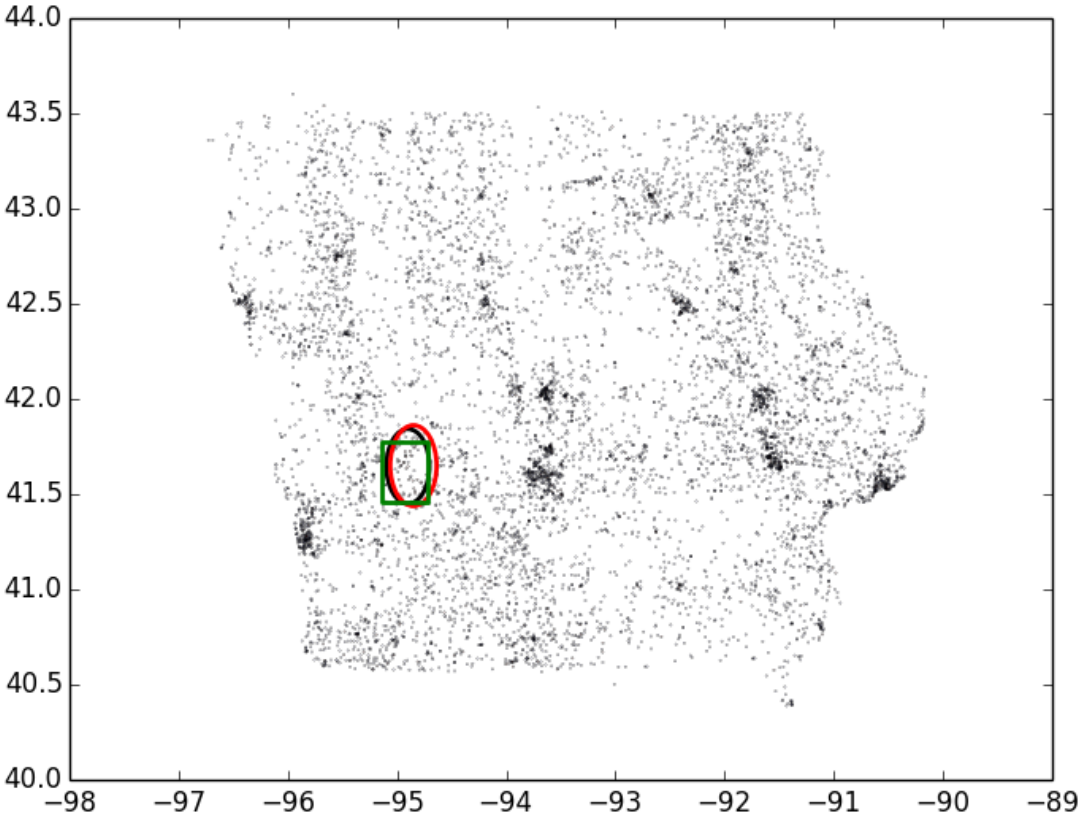

Figure 4.6: True cluster(red), Observed cluster(black) using Limit Ranges to Neighborhoods approach and Observed cluster(green) using the Neill et al. [12] algorithm for grid size of 32 x 32.

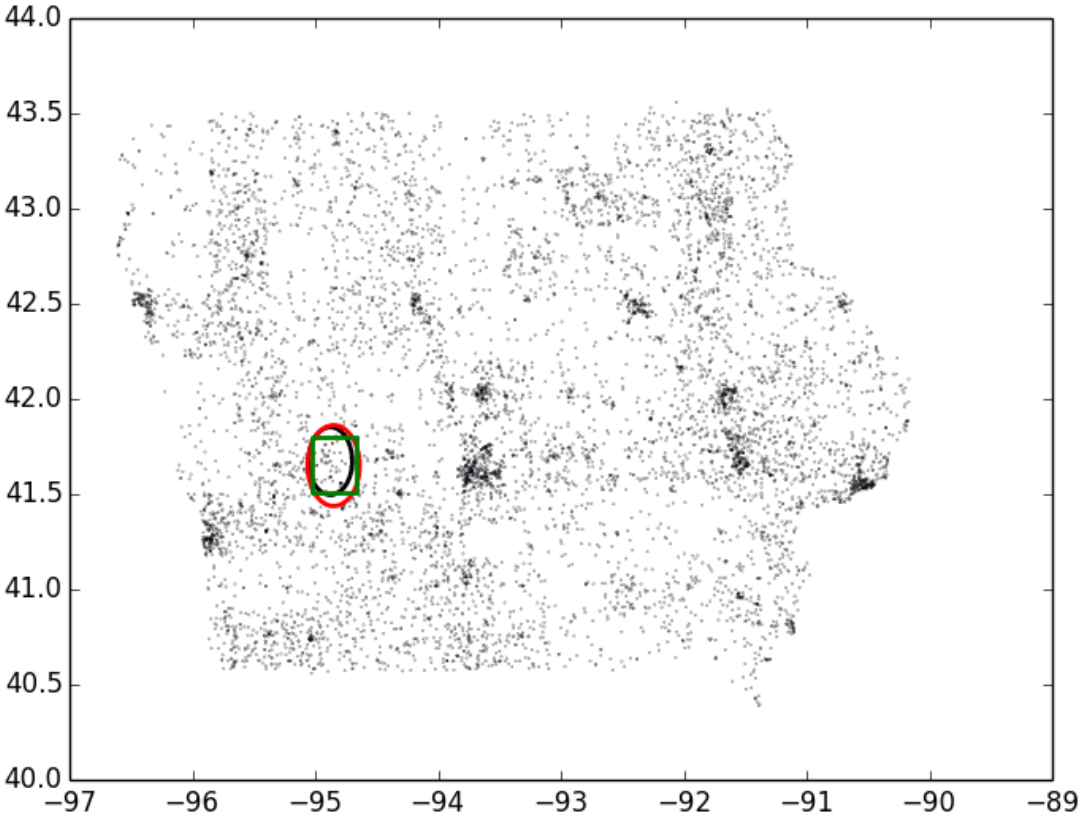

Figure 4.7: True cluster(red), Observed cluster(black) using Limit Ranges to Neighborhoods approach and Observed cluster(green) using the Neill et al. [12] algorithm for grid size of 128 x 128.

Limit Ranges To Neighbourhood approach and observed cluster detected by the Neill et al. [12] algorithm for grid size of 32 x 32 and 128 x 128, respectively, for the Road dataset. Figure 4.8 and Figure 4.9 represent the true cluster, observed cluster detected by *Sample Data to Limit Ranges* approach and observed cluster detected by the Neill et al. [12] algorithm for grid size of 32 x 32 and 128 x 128, respectively, for the Road dataset. The circle in red represents the true cluster, the circle in black represents the observed cluster detected by our algorithm and the circle in green represents the observed cluster detected by the Neill et al. [12] algorithm.

We then performed experiments on the Crime and Surgeon data set in United States. For both the datasets ground truth is the circular region with maximum discrepancy score obtained by executing the algorithm on all the circular regions defined over entire data set. The Crime dataset contains latitude, longitude and crime type data from April 2015 to May 2015. This dataset contains 0.34 million points. The latitudes and longitudes are scaled in [0,1] range and considered as x and y coordinate for corresponding points. We are searching the cluster for crime type such as Drugs. Hence all those points which have crime type Drugs will be considered as red points and all other points as blue.

The Surgeon dataset contains only around 15,000 points. Each point contains latitude, longitude and gender of surgeon. The latitudes and longitudes are scaled in [0,1] range and considered as x and y coordinate for corresponding points. We are searching the cluster for gender as Female(F). Hence all those points which have gender type as F will be considered as red points and all other points as blue.

The comparison of results produced by executing our algorithms and the Neill et al. algorithm on the Crime data set is given in Table 4.3. Table 4.4 presents the result of comparison of the Neill et al. algorithm and our algorithms on Surgeon data set.

The number inside the bracket [] in *Run Time(seconds)* column of Table 4.2, Table 4.3 and Table 4.4 represents the grid size. It is quite evident from the results that as we increase the grid size the accuracy of detected cluster center increases but at the cost of increased time complexity.

4.1 Avoiding Significance Testing

One way to test the significance of a detected cluster is to compare the maximum discrepancy score with the 95th percentile value in the distribution of maximum discrepancy scores obtained by executing the algorithm multiple times say 1000 on the uniformly distributed point data set with the probability of a point being red

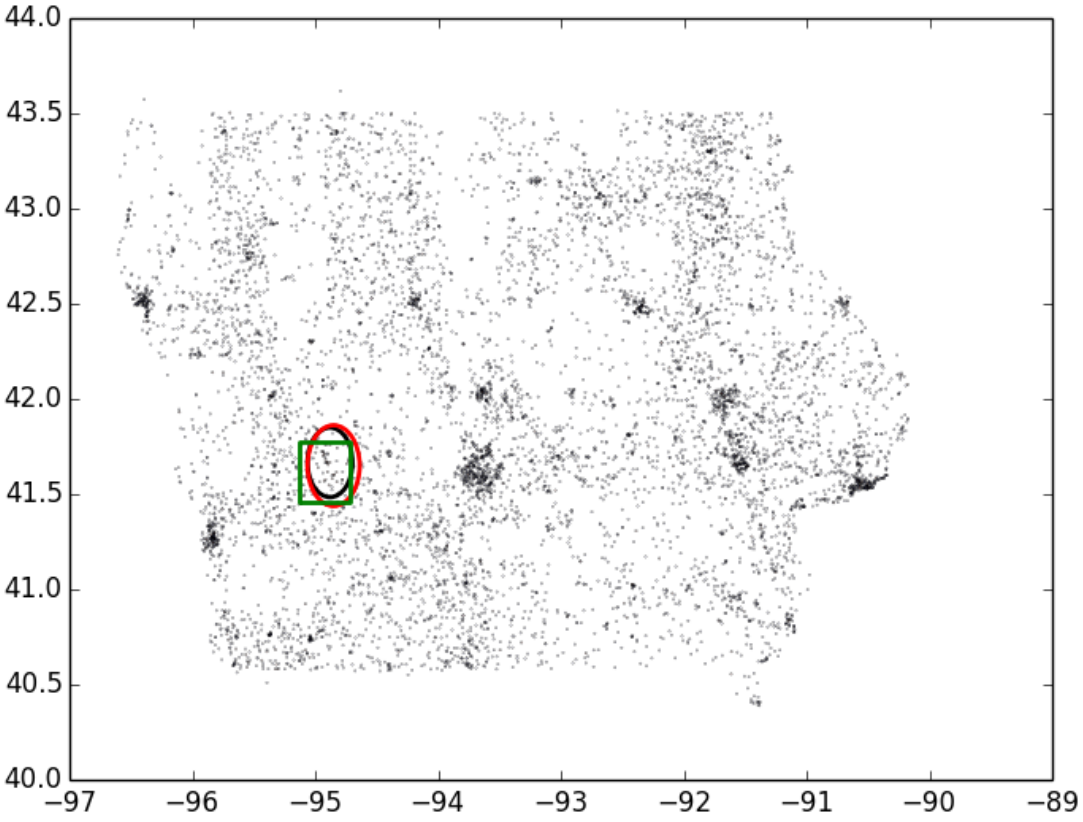

Figure 4.8: True cluster(red), Observed cluster(black) using Sample Data To Limit Ranges approach and Observed cluster(green) using the Neill et al. [12] algorithm for grid size of 32 x 32.

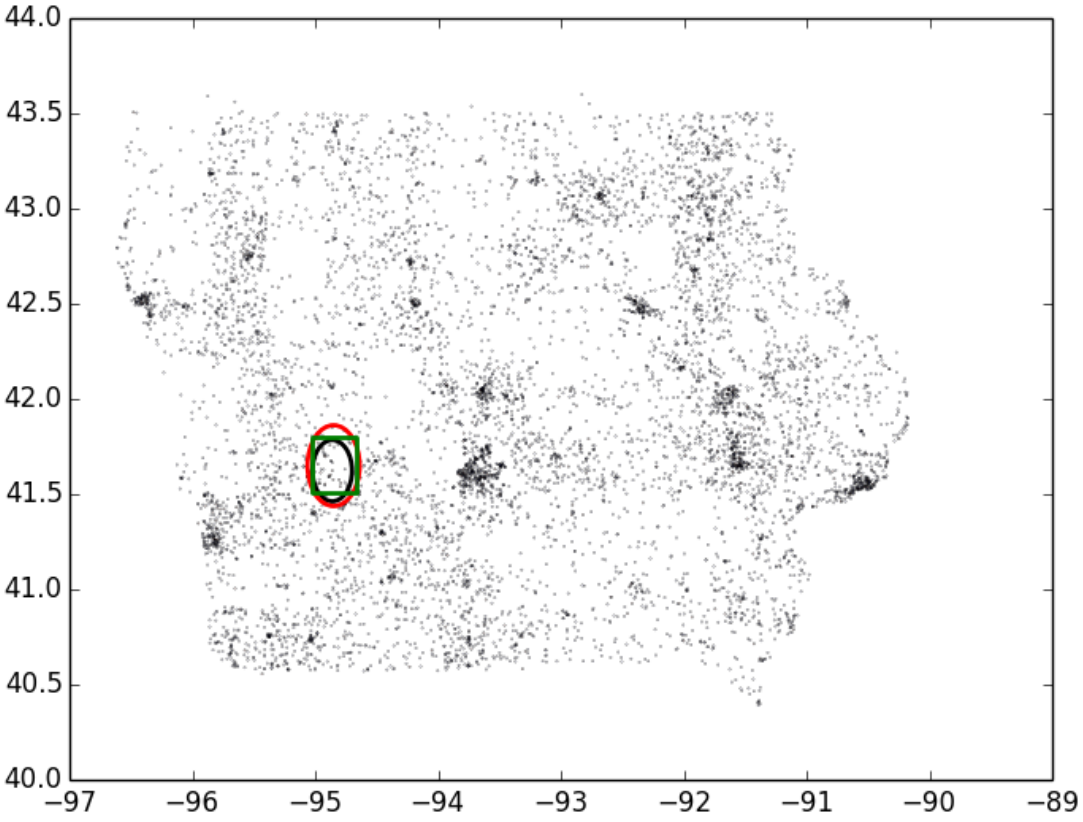

Figure 4.9: True cluster(red), Observed cluster(black) using Sample Data To Limit Ranges approach and Observed cluster(green) using the Neill et al. [12] algorithm for grid size of 128 x 128.

Table 4.3: Comparison of results between Neill et al. [12] and our algorithm on Crime data set scaled to fit in a box of [0,1] x [0,1]

	Observed Cluster Center	Run Time(seconds)
Neill et al. [12]	(0.1915,0.7815)	45.116 [128]
Neill et al. [12]	(0.1875,0.75)	9.774 [64]
Neill et al. [12]	(0.1875,0.594)	3.643 [32]
Sample Data to limit Ranges	(0.1954,0.7922)	4.492
Limit Ranges to Neighborhoods	(0.1856,0.8026)	2.0239 [k=50]

Table 4.4: Comparison of results between Neill et al. [12] and our algorithm on Surgeon data set scaled to fit in a box of [0,1] x [0,1]

	Observed Cluster Center	Run Time(seconds)
Neill et al. [12]	(0.637,0.8204)	58.093 [128]
Neill et al. [12]	(0.6095,0.8675)	11.1384 [64]
Neill et al. [12]	(0.7185,0.8215)	5.3668 [32]
Sample Data to limit Ranges	(0.628,0.823)	11.538
Limit Ranges to Neighborhoods	(0.6236,0.8425)	26.933 [k=50]

equal to p. In every execution of algorithm the position of a point will remain same but it can be red or blue depending on probability p. The significance test takes a lot of time which we can avoid if we were able to prove that the distribution of maximum discrepancy scores on the real world data set with every point is equally likely of being red (say with probability p) is very similar to the distribution of maximum discrepancy scores on the synthetic data set with every point having the same probability p of being red. In each trial we executed *Sample Data to Limit Ranges* algorithm 1000 times on synthetic data set containing 40,000 points and the Road data set containing more than a million points and then created distribution for both the data sets. For both data sets we kept the probability of a point being red equal to 0.05. We executed 5 such trials and calculated different statistics on both the data sets. We then took the Mean and Standard Deviation of all the statistics over all the 5 trials. The results are shown in Table 4.5.

We can see from Table 4.5 that the statistics of distribution for Synthetic and Road data set are quite similar and hence if we perform significance testing once for a dataset with fixed value of n and p, then we are probably safe using those values for future data sets with same (or similar) values of n and p where n is the total number of data points and p is the probability of a point being red.

Table 4.5: Comparison of distribution of Synthetic and Road data set averaged over multiple trials.

Statistic	Mean Synthetic	Std. Dev. Synthetic	Mean Road	Std. Dev. Road
Mean	0.042	0.000155	0.042	0.000094
Std. Dev.	0.00945	0.00014	0.0093	0.000061
90th percentile	0.085	0.00256	0.0836	0.00136
95th percentile	0.0904	0.00287	0.088	0.00167
99th percentile	0.1032	0.00483	0.1068	0.00747

CHAPTER 5

POISSON STATISTICAL SCAN MODEL AND LIPSCHITZ CONTINUITY

Poisson discrepancy function has already been derived in Kulldorff [7] . There is a derivation in Agarwal et al. [9](see also [19], Chapter 4). We have included the derivation in the appendix for completeness.

5.1 Lipschitz Properties of Poisson Discrepancy

We want to preserve the region with maximum discrepancy and its score under random sampling because then we can say that the value of discrepancy function for the observed cluster is very close to the value of discrepancy function for the true cluster and then the observed cluster detected by our algorithm would be much closer to the true cluster. It is known that random sampling preserves the density up to some error which can be derived using random sampling theory [20]. Thus for any range we consider, the values m and b are preserved up to a bounded error. Because the discrepancy function is a nonlinear function of the values m and b, it is hard to assess exactly how much error is caused in the discrepancy function by sampling. However, in the following paragraph we show that in a bounded range of m and b values, the discrepancy function is Lipschitz. That is the absolute value of its derivative is bounded by some real constant, and can thus be approximated well by a single linear function. In other words, in the bounded range, as m and b change, the change in the discrepancy function is affected in a way that it is bounded by a linear function in these changes to m and b.

Let the discrepancy function is $L(m, b)$ where:

$L(m, b) = m \log(\frac{m}{b}) + (1 - m) \log(\frac{1-m}{1-b})$

$\frac{\partial L}{\partial m} = \log\left[\frac{m(1-b)}{b(1-m)}\right]$ and $\frac{\partial L}{\partial b} = \frac{b-m}{b(1-b)}$ $-2 \leq \frac{\partial L}{\partial m} \leq 3$ and $-10 \leq \frac{\partial L}{\partial b} \leq 2$ for all the values of m ranging from 0.01 to 0.2 and for all the values of b ranging from 0.02 to 0.5. Outside this range of m and b the derivatives may not be bounded, that is, change in discrepancy

score can be huge for a small change in values of m and b and thus the region of maximum discrepancy may not be preserved under sampling outside of this range of m and b.

Under the more realistic setting we chose m ranging from 0.01 to 0.2 and b ranging from 0.02 to 0.5 and then we have shown that $\frac{\partial L}{\partial b}$ and $\frac{\partial L}{\partial m}$ are bounded for m ranging from 0.01 to 0.2 and b ranging from 0.02 to 0.5 and hence the discrepancy function $L(m, b)$ is Lipschitz continuous.

Figure 5.1 shows the graph of $\frac{\partial L}{\partial m}$ for b=0.02. Figure 5.2 shows the graph of $\frac{\partial L}{\partial b}$ for m=0.2. From Figure 5.1 it is evident that $\frac{\partial L}{\partial m}$ is bounded for $0.01 \leq m \leq 0.2$ and similarly from Figure 5.2 we can see that $\frac{\partial L}{\partial b}$ is bounded for $0.02 \leq b \leq 0.5$.

$\frac{\partial L}{\partial m}$ is a log function of m having b as a constant and since it is a log function it does not change much as compared to $\frac{\partial L}{\partial b}$. So we really need to find the range of values of m for which $\frac{\partial L}{\partial b}$ is bounded within a small range because $\frac{\partial L}{\partial b}$ is not a log function and thus changes more rapidly as compared to $\frac{\partial L}{\partial m}$. Once we find out the range of values of m for which $\frac{\partial L}{\partial m}$ is bounded we then can try to find out the bounds for $\frac{\partial L}{\partial b}$ and the corresponding range for the values of b. For having the bounds for $\frac{\partial L}{\partial m}$ in the range from -2 to 3 we started from b=0.02 and we found out that the lower bound for $\frac{\partial L}{\partial m}$ holds for m starting from 0.01. If we make b any smaller than 0.02 the lower bound for $\frac{\partial L}{\partial m}$ becomes tighter but the upper bound does not hold because the log function shifts around whatever value we choose for b. If we choose b=0.01 then the $\frac{\partial L}{\partial m}$ becomes 0 at m=0.01. It then decreases for values of m less than 0.01 and increases for values of m more than 0.01. Hence the lower bound for m becomes tighter whereas the upper bound goes up more and more. We can however take values of b smaller than 0.02 by keeping $\frac{\partial L}{\partial m}$ in a bounded range from -2 to 3 but then we have to shift the range of m to 0.001 to 0.17 which is not a realistic range because a real world data set does not have a fraction of all red points inside the cluster in the range from 0.1 % to 17 %. It should be more than that. Therefore the value of b=0.02 is more practical for getting a range of m values giving a better real world sense.

We chose the upper value of m as 0.2 because if we go for any higher value the lower bound for $\frac{\partial L}{\partial b}$ will become less. Therefore in order to have a lower bound of $\frac{\partial L}{\partial b}$ as -10 we need to choose the upper value of m=0.2. Hence $\frac{\partial L}{\partial m}$ ranges from -2 to 3 for values of m ranging from 0.01 to 0.2 keeping b=0.02. We now know the lower value of b as 0.02 and then we choose the upper value of b as 0.5 in order to make $\frac{\partial L}{\partial b}$ bounded in the range from -10 to 2. Thus both $\frac{\partial L}{\partial m}$ and $\frac{\partial L}{\partial b}$ are bounded in a small range and hence the function $L(m, b)$ is Lipschitz continuous in a bounded range.

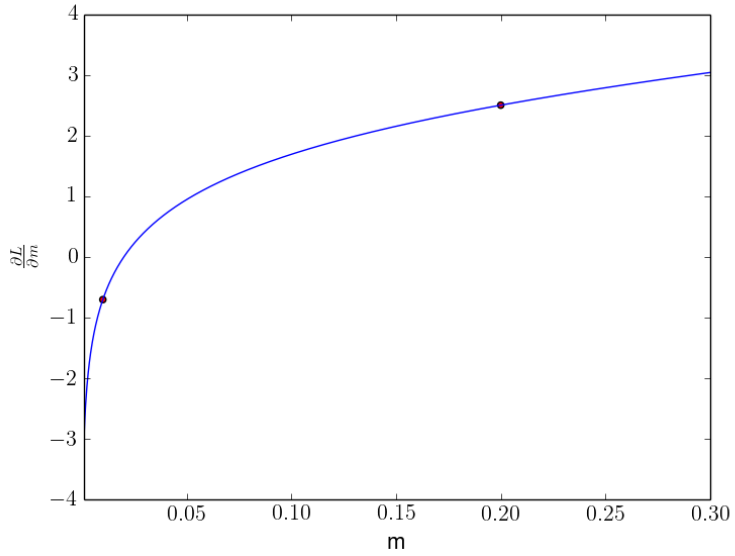

Figure 5.1: Graph of $\frac{\partial L}{\partial m}$ for b=0.02. The graph can be considered flat between the two red dots as the value of $\frac{\partial L}{\partial m}$ does not change much. Therefore the discrepancy function shows Lipschitz property between the two red dots.

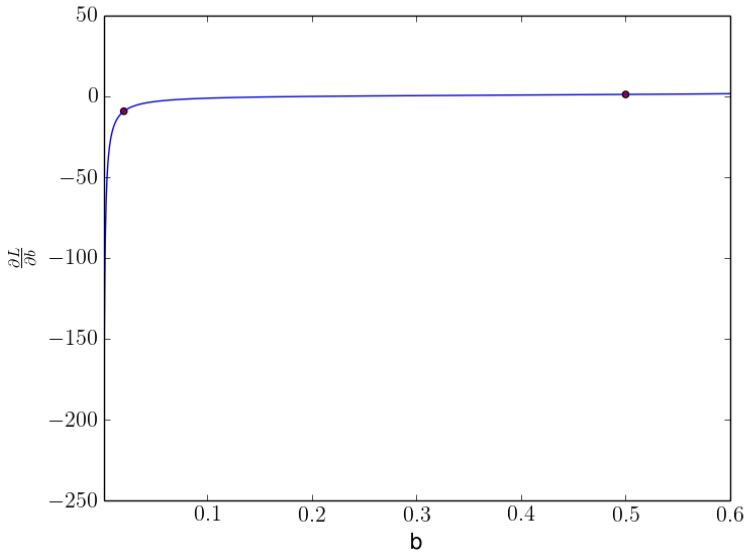

Figure 5.2: Graph of $\frac{\partial L}{\partial b}$ for m=0.2. The graph can be considered flat between the two red dots as the value of $\frac{\partial L}{\partial b}$ does not change much. Therefore the discrepancy function shows Lipschitz property between the two red dots.

CHAPTER 6

RELATED WORK

In this section we are going to review some of the related work done in the field of spatial scan statistics. Some of the approaches worked on reducing the number of scanning regions while others focussed on detecting irregular shaped clusters.

Besag and Newell (1991) [21] only considered the circles having a red point as the center and then increasing the size of these regions until a minimum number of red points are included. Here cases are the red points. Instead of considering every point as a center, this paper considered only the cases as the center of the circle. They then determine the optimal radius by starting at the smallest possible radius, and slowly increasing the size until a minimum number of cases are included in the circle. This method has the advantage of faster running times, especially in large datasets with only a few cases. Also, it does not waste time searching for clusters in areas that have no cases. Therefore, this method has an improved time complexity given by: $O(k(rk + n))$, where k is the number of case points, r is the average number of radii tested before the minimum case number is met, and n is the number of data points. At each case point, an average of r circles are created. A point-in-circle test for the k case points is performed for each of these r circles to determine whether or not the circle contains the minimum number of cases, and a point-in-circle test for the n data points is performed on the final circle. Their approach is very similar to ours except the time complexity of our algorithm is better because we are only dealing with sampled data set of size $s < n$.

Tango and Takahashi (2005) [22] presented a flexibly shaped spatial scan statistic (FlexScan) where the neighboring regions are aggregated to the cluster during the scan to detect flexibly shaped clusters. In order to put a limit on the size of the cluster they chose $10 \sim 15$ % of the entire study area as the probable cluster size. This method extends spatial scan statistics for detecting irregular shaped clusters but only detects clusters of moderate sizes.

Zhijun et al. [23] proposed *maxima-likelihood-first (MLF)* and *non-greedy growth (NGG)*

algorithm for detecting irregular shaped clusters. In MLF algorithm Log Likelihood Ratio (LLR) of every area is calculated and sorted to get maximum Log Likelihood Area as a candidate region. Then the candidate region is aggregated with its neighbours and forms a group of new regions. The LLRs of these new regions are also calculated and combined with the LLRs of previous regions and sorted again to get one with the new maximum LLR as the new candidate. The algorithm repeats itself until a threshold is reached. NGG algorithm allows not only neighboring area with the local maximum to be included but also includes many other neighboring areas in the search procedure. Specifically, a threshold M is set for the maximum number of potential clusters generated at each step. Then all areas are put into a temporary list and the LLRs of these areas are calculated. In the next step, the average number of neighbors (L) of each region is calculated. The approximate number of candidates (N) for the next iteration is given by equation N = M/L. N areas with the highest LLRs are chosen from the temporary list and the list is emptied afterward. New regions created from the candidates and their neighbors are put into the emptied list. These steps are repeated until either the aggregated area covers half of the study area or has half of the total population.

Neill et al. [24] proposed a fast subset scan method called Linear Time Subset Scan (LTSS). They only perform search over N subsets rather than an exhaustive search over 2^N subsets where N is the total number of points in the data set. For a given data set D, the score function $F(S), S \subseteq D$ and priority function $G(s), s \in S$ satisfy the LTSS property if and only if $\max_{S \subseteq D} F(S) = \max_{j=1...N}[F\{s_1...s_j\}]$. The dataset D consists of spatial locations s_i having statistics count c_i and b_i. They proved that the commonly used spatial scan statistics, including Kulldorff's original spatial scan statistic [7] and many recently proposed variants such as the expectation-based Poisson [25], Gaussian [11] and exponential [1] scan statistics, satisfy the LTSS property. The authors proposed that if $F(S) = F(X, Y)$ be a quasi convex function of X and Y where $X(S) = \sum_{s_i \in S} x_i$ and $Y(S) = \sum_{s_i \in S} y_i$ and if $F(S)$ is monotonically increasing with $X(S)$, and that all y_i values are positive then $F(S)$ satisfies the LTSS property with priority function $G(s_i) = \frac{x_i}{y_i}$. For the Poisson Statistics $F(S) = m \log\left[\frac{m}{b}\right] + (1-m)\log\left[\frac{1-m}{1-b}\right]$ where $m(S) = \sum_{s_i \in S} x_i$ and $b(S) = \sum_{s_i \in S} y_i$. $F(S)$ monotonically increases with $m(S)$ and all b_i are assumed to be positive. Since $F(S)$ is a quasi convex function as it is a sum of two convex functions $m \log\left[\frac{m}{b}\right]$ and $(1-m)\log\left[\frac{1-m}{1-b}\right]$. Therefore $F(S)$ satisfies LTSS property. Now a subset of data records which maximizes F(S) can be found by ordering the regions according to some priority function and searching over groups consisting of the top k highest priority regions, requiring only a linear rather than

exponential number of subsets to be evaluated. The authors proposed a fast localized scan approach where they consider every point in the data set as the center s_i of some region and S_i as its local neighborhood containing the center s_i itself and k-1 nearest neighbours to s_i. Then they use LTSS to maximize $F(S)$ for the given neighborhood by evaluating only $O(k)$ of the $O(2^k)$ subsets. The running time of their algorithm is $O(Nk + N \log N)$ for fixed k assuming that the nearest k points are already known. This fast localized scan approach is very similar to our *Limit Ranges to Neighborhoods* approach but the time complexity of our approach is better because we are considering only red points as the center of the region and if we keep the same assumption that the nearest k points are already known then the time complexity of our algorithm is $O(rk + s \log s)$ where r is the total number of red points and s is the total number of sampled points. Since $r < N$ and $s < N$. Therefore the time complexity of our algorithm is better.

Neill and Moore [2] proposed a fast multiresolution method for detecting the maximum discrepancy region by applying branch and bound method of pruning to the data present in a grid. Counts are aggregated to a square grid G of size $N \times N$ where each square $s_{ij} \in G$ is associated with count c_{ij} and underlying population p_{ij}. The naive approach for finding the maximum discrepancy region takes $O(N^3)$ as there can be total $O(N^3)$ squares in an $N \times N$ grid. This can be very expensive. Therefore the authors came up with an ingenious multiresolution data structure called overlap-mutires tree where each region S is divided into four overlapping subregions of size k. Overlapping is required because it can be possible that a dense region may get split into two or more subregions none of which is as dense as the original region. On careful examination of the tree one can see that at level i there are 2^i regions. There are total $O(N^2)$ gridded regions to be searched and therefore if only the gridded regions are to be searched the time complexity is reduced to $O(N^2)$. The significance testing can be made fast by using simple tricks like stop examining a replica G' immediately if a region with density greater than the density of maximum discrepancy region for G is found. The significance testing can be stopped early if after a number of replications $R' < R$, we can conclude with high confidence that the region is not significant. However the regions with significant spatial overdensity are still computationally expensive to search and hence there is a need of multiresolution data structure. Top down pruning is done first by deriving an upper bound $D_{max}(S, k)$ on the density of subregions of minimum size k contained in a given region S. Then if $D_{max}(S, k) < D(S^*)$, S^* is the maximum density region, we know that no subregion of S with size k or more can be the maximum

density region. We have already compared our approach to this fast multiresolution method in section 4.

CHAPTER 7

CONCLUSION

We tried to solve the problem of finding the most anomalous region for large scale data sets. There are many other approaches to trying to find the most anomalous region but they lack either in accuracy or take much more time to do so. We came up with three novel approaches for solving this problem for large data sets. We made our approaches scalable by first sampling from the original data set and then executing our approaches on the sampled data set. We proved that the results obtained from original and sampled data set are very similar. We also compared the results obtained by our algorithms with the others. Though our approaches work only for the circular region, we can extend them to detect clusters of arbitrary shape and size as a future work. We can also make our algorithms detect multiple significant clusters.

APPENDIX

REDERIVATION OF THE POISSON DISCREPANCY

Let N denote a spatial point process where $N(A)$ is the random number of points in the set $A \subset G$. As the circular window moves over the geographical area under study it defines a collection of zones $Z \subset \mathbb{Z}$. Under the Poisson model points are generated by an inhomogeneous Poisson process. We define a measure μ such that $\mu(A)$ is an integer for all subsets $A \subset G$. Each unit of measure corresponds either to a red or blue point and the location of these points constitute a point process. In the model there is exactly one zone $Z \subset G$ such that each point within this zone has probability p of being a red point while the probability of points being red outside the zone is q. The probability for any one individual point is independent of others. The null hypothesis is $H_o : p = q$. The alternative hypothesis is $H_1 : p > q, Z \in \mathbb{Z}$. Under $H_o, N(A) \sim Po(p\mu(A))\ \forall A$. Under $H_1, N(A) \sim Po(p\mu(A \cap Z) + q\mu(A \cap Z^c))\ \forall A$.

According to Kulldorff [7], the probability for n_G number of red points, following Poisson distribution, in the study area is given by $\frac{e^{-p\mu(Z)-q(\mu(G)-\mu(Z))}[p\mu(Z)+q(\mu(G)-\mu(Z))]^{n_G}}{n_G!}$. The equation says that all the n_G red points are distributed according to Poisson distribution with $[p\mu(Z) + q(\mu(G) - \mu(Z))]$ as the expected number of occurrences of red points in the study area. If we observe more carefully we can see that $p\mu(Z)$ is the expected number of red points inside the region Z because there are total $\mu(Z)$ points(blue+red) inside the region Z and each point has probability p of being red. Therefore the expected number of red points inside the region Z is $p\mu(Z)$. Similarly, the expected number of points outside the region Z is $q(\mu(G)-\mu(Z))$ because there are total $(\mu(G)-\mu(Z))$ number of points(blue+red) outside the region Z where $\mu(Z)$ is the number of points(blue+red) inside the region Z and $\mu(G)$ is the total number of points(blue+red) inside the entire study region G. The density function f(x) of a red point being observed at location x is given by:

$$\begin{cases} \frac{p\mu(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))} & \text{if } x \in Z \\ \frac{q\mu(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))} & \text{if } x \notin Z \end{cases}$$

$p\mu(x)$ is the expectation of point being red and $p\mu(Z) + q(\mu(G) - \mu(Z))$ is the expected number of red points inside the entire study region G. Therefore the probability of a point being red inside region Z is $\frac{p\mu(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))}$. Similarly, the probability of a point being red outside the region Z is $\frac{q\mu(x)}{p\mu(Z)+q(\mu(G)-\mu(Z))}$.

Since the probability for any individual point is independent of all other points, the likelihood function for an alternate hypothesis can then be written as the product of probability for n_G number of points in the study area, product of probabilities of all the red points being observed inside region Z and the product of probabilities of all the red points being observed outside region Z. We can write the likelihood function for alternate hypothesis as:

$$\begin{aligned} L_1 &= \frac{e^{-p\mu(Z)-q(\mu(G)-\mu(Z))}[p\mu(Z)+q(\mu(G)-\mu(Z))]^{n_G}}{n_G!} \times \prod_{x_i \in Z} \frac{p\mu(x_i)}{p\mu(Z)+q(\mu(G)-\mu(Z))} \\ &\quad \times \prod_{x_i \notin Z} \frac{q\mu(x_i)}{p\mu(Z)+q(\mu(G)-\mu(Z))} \\ &= \frac{e^{-p\mu(Z)-q(\mu(G)-\mu(Z))}}{n_G!} p^{n_Z} q^{(n_G-n_Z)} \prod_{x_i} \mu(x_i) \end{aligned}$$

For L_1 we first take the supremum over all p and q for a fixed Z. L_1 takes its maximum when $p = \frac{n(Z)}{\mu(Z)}$ and $q = \frac{n_G-n_Z}{\mu(G)-\mu(Z)}$. So

$$L_1 = \frac{e^{-n_G}}{n_G!} \left(\frac{n_Z}{\mu_Z}\right)^{n_Z} \left(\frac{n_G-n_Z}{\mu(G)-\mu(Z)}\right)^{n_G-n_Z} \prod_{x_i} \mu(x_i)]$$

Likelihood function for null hypothesis is:

$$L_o = \frac{e^{-p\mu(G)} p^{n_G}}{n_G!} \prod_{x_i} \mu(x_i)$$

For L_o we first take the supremum over all p. L_o takes its maximum when $p = \frac{n_G}{\mu(G)}$.

$$L_o = \frac{e^{-n_G}}{n_G!} \left(\frac{n_G}{\mu_G}\right)^{n_G} \prod_{x_i} \mu(x_i)$$

The test statistic λ of the likelihood ratio test can now be written as:

$$\lambda = \sup_{Z\in\mathbb{Z}} \frac{L_1}{L_o}$$
$$= \begin{cases} \sup\limits_{Z\in\mathbb{Z}} \frac{\left(\frac{n_Z}{\mu(Z)}\right)^{n_Z}\left(\frac{n_G-n_Z}{\mu(G)-\mu(Z)}\right)^{n_G-n_Z}}{\left(\frac{n_G}{\mu(G)}\right)^{n_G}}, & \text{if } \frac{n_Z}{\mu(Z)} > \frac{n_G-n_Z}{\mu(G)-\mu(Z)} \\ 1 & \text{otherwise} \end{cases}$$

Taking log on both sides:

$$\log(\lambda) = \sup_{Z\in\mathbb{Z}} n_Z \log\left[\frac{n_Z}{\mu(Z)}\right] + (n_G - n_Z)\log\left[\frac{n_G - n_Z}{\mu(G)-\mu(Z)}\right] - n_G \log\left[\frac{n_G}{\mu(G)}\right]$$
$$= \sup_{Z\in\mathbb{Z}} n_G \frac{n_Z}{n_G} \log\left[\frac{n_Z}{n_G\frac{\mu(Z)}{\mu(G)}}\frac{n_G}{\mu(G)}\right] + n_G(1-\frac{n_Z}{n_G})\log\left[\left(\frac{1-\frac{n_Z}{n_G}}{1-\frac{\mu(Z)}{\mu(G)}}\right)\frac{n_G}{\mu(G)}\right]$$
$$- n_G \log\left[\frac{n_G}{\mu(G)}\right]$$

Let $m = \frac{n_Z}{n_G}$ and $b = \frac{\mu(Z)}{\mu(G)}$

$$\log(\lambda) = \sup_{Z\in\mathbb{Z}} n_G m \log\left[\frac{m}{b}\right] + m\log\left[\frac{n_G}{\mu(G)}\right] + (1-m)\log\left[\frac{1-m}{1-b}\right] + (1-m)\log\left[\frac{n_G}{\mu(G)}\right]$$
$$- \log\left[\frac{n_G}{\mu(G)}\right]$$
$$= \sup_{Z\in\mathbb{Z}} n_G \left[m\log\left[\frac{m}{b}\right] + (1-m)\log\left[\frac{1-m}{1-b}\right]\right]$$

we define the Poisson discrepancy, *dp*, as

$dp(Z) = m\log\left[\frac{m}{b}\right] + (1-m)\log\left[\frac{1-m}{1-b}\right]$

$log(\lambda) = n_G \max_{Z\in\mathbb{Z}} dp(Z)$

REFERENCES

[1] L. Huang, M. Kulldorff, and D. Gregorio, "A spatial scan statistic for survival data," *Biometrics*, vol. 63, no. 1, pp. 109–118, 2007.

[2] D. B. Neill and A. W. Moore, "A fast multi-resolution method for detection of significant spatial disease clusters," in *Advances in Neural Information Processing Systems*, 2003, p. None.

[3] Y. Sun, H. Fan, M. Helbich, and A. Zipf, "Analyzing human activities through volunteered geographic information: Using flickr to analyze spatial and temporal pattern of tourist accommodation," in *Progress in Location-Based Services*. Springer, 2013, pp. 57–69.

[4] G. Patil, V. Patil, S. Pawde, S. Phoha, V. Singhal, and R. Zambre, "Digital governance, hotspot geoinformatics, and sensor networks for monitoring, etiology, early warning, and sustainable management," *Geoinformatics for Natural Resource Management*, pp. 1–98, 2008.

[5] X. Wang, M. S. Gerber, and D. E. Brown, "Automatic crime prediction using events extracted from twitter posts," in *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 2012, pp. 231–238.

[6] M. Wu, X. Song, C. Jermaine, S. Ranka, and J. Gums, "A lrt framework for fast spatial anomaly detection," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2009, pp. 887–896.

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

[8] ——, "Satscan user guide for version 9.0," 2011.

[9] D. Agarwal, J. M. Phillips, and S. Venkatasubramanian, "The hunting of the bump: On maximizing statistical discrepancy," in *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*. Society for Industrial and Applied Mathematics, 2006, pp. 1137–1146.

[10] D. B. Neill, "Expectation-based scan statistics for monitoring spatial time series data," *International Journal of Forecasting*, vol. 25, no. 3, pp. 498–517, 2009.

[11] D. B. Neill, A. W. Moore, and G. F. Cooper, "A bayesian spatial scan statistic," *Advances in Neural Information Processing Systems*, vol. 18, p. 1003, 2006.

[12] D. B. Neill and A. W. Moore, "Rapid detection of significant spatial clusters," in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2004, pp. 256–265.

[13] D. Agarwal, A. McGregor, J. M. Phillips, S. Venkatasubramanian, and Z. Zhu, "Spatial scan statistics: Approximations and performance study," in *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2006, pp. 24–33.

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

[15] J. Krumm, "Inference attacks on location tracks," in *5th International Conference on Pervasive Computing*, 2007.

[16] M. Decker, "Location privacy – an overview," in *7th International Conference on Mobile Business*, 2008.

[17] J. P. A. Ioannidis, "Why most published research findings are false," *PLoS Medicine*, vol. 2, p. 124, 2005.

[18] A. Gelman and E. Loken, "The garden of forking paths: Why multiple comparisons can be a problem, even when there is no fishing expeditionor p-hackingand the research hypothesis was posited ahead of time," *Downloaded January*, vol. 30, p. 2014, 2013.

[19] J. M. Phillips, "Small and stable descriptors of distributions for geometric statistical problems," Ph.D. dissertation, Duke University, 2009.

[20] V. N. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory of Probability & Its Applications*, vol. 16, no. 2, pp. 264–280, 1971.

[21] J. Besag and J. Newell, "The detection of clusters in rare diseases," *Journal of the Royal Statistical Society. Series A (Statistics in Society)*, vol. 154, no. 1, pp. pp. 143–155. [Online]. Available: http://www.jstor.org/stable/2982708

[22] T. Tango and K. Takahashi, "A flexibly shaped spatial scan statistic for detecting clusters," *International Journal of Health Geographics*, vol. 4, no. 1, p. 11, 2005.

[23] Z. Yao, J. Tang, and F. B. Zhan, "Detection of arbitrarily-shaped clusters using a neighbor-expanding approach: A case study on murine typhus in south Texas," *International journal of health geographics*, vol. 10, no. 1, p. 23, 2011.

[24] D. B. Neill, "Fast subset scan for spatial pattern detection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 74, no. 2, pp. 337–360, 2012. [Online]. Available: http://dx.doi.org/10.1111/j.1467-9868.2011.01014.x

[25] D. B. Neill and A. W. Moore, "Anomalous spatial cluster detection," in *Proceedings of the KDD 2005 Workshop on Data Mining Methods for Anomaly Detection*, 2005.