A Nondeterministic Normalization based Scan Statistic (NN-scan) towards Robust Hotspot Detection: A Summary of Results

Yiqun Xie* Shashi Shekhar*

Abstract

Hotspot detection aims to find sub-regions of a space that have higher probability density of generating certain events (e.g., disease, crimes) than the other regions. Finding hotspots has important applications in many domains including public health, crime analysis, transportation, etc. Existing methods of hotspot detection rely on test statistics (e.g., likelihood ratio, density) that do not consider spatial nondeterminism, leading to false and missing detections. We provide theoretical insights into the limitations of related work, and propose a new framework, namely, Nondeterministic Normalization based scan statistic (NN-scan), to address the issues. We also propose a DynamIc Linear Approximation (DILA) algorithm to improve NN-scan's efficiency. In experiments, we show that NN-scan can significantly improve the precision and recall of hotspot detection and DILA can greatly reduce the computational cost.

1 Introduction

Given a collection of geo-located instances of an event (e.g., disease or crime cases) in a spatial domain, hotspot detection aims to find geographic regions with higher probability density of generating instances of such event than the rest of the study area.

As a core topic in spatial data mining [11], hotspot detection has important societal applications in many domains. For example, in public health, researchers use hotspot detection to find regions of disease outbreak, including childhood leukemia, Legionnaires' disease, cancer [1, 7, 8], etc. The hotspot found in the 1854 London cholera map ended the spread of the disease, saved numerous lives and became a major milestone in the development of the Germ Theory, a turning point of the modern science [10]. The National Cancer Institute has included hotspot detection as a standard method for its surveillance research program [1]. In crime analysis, ring-shaped hotspot detection is used to locate serial criminals (e.g., arsonists) whose crime zones are often ring-shaped areas [4]. In transportation, linear hotspots on road networks help authorities identify outbreaks of road accidents or pedestrian fatalities (e.g., caused by deteriorated road conditions) [12].

In these societal applications, the price of making a mistake is normally very high (e.g., losing control of a real disease outbreak or wasting resources on false positives). Thus, the robustness of a hotspot detection approach (e.g., statistical significance) is critical. Besides the above examples, hotspot detection is also used in many other domains, including agriculture, forestry, astronomy, geology, etc [7, 1].

The spatial scan statistic [7, 6] is the most widely used approach in hotspot detection. Its corresponding software, SaTScan [2], has become a standard tool in a variety of research fields (e.g., National Cancer Institute [1]). Spatial scan statistics provide a hotspot detection framework based on a likelihood ratio test. Extending this framework, many techniques have been proposed in the field of data mining to enable hotspot detection for a richer set of geometric shapes, including rectangular [8, 9], circular [7], ellipsoidal, ring [4], linear [12] and arbitrary [3] in both Euclidean space and network space (e.g., road and river networks). While these approaches mainly focus on enumeration strategies of candidate regions of different shapes, the most widely used test statistic for candidate evaluation – likelihood ratio – itself still has well-known issues in practice (e.g., favoring tiny hotspots, example in Sec. 3) [13]. We provide detailed theoretical analyses to show that these issues are caused by ignoring spatial nondeterminism (formally defined in Sec. 4.1). Besides spatial scan statistic based methods, traditional clustering methods (e.g., DBSCAN[5], k-means, EM) can also be used to find high-density regions. However, since these methods do not involve statistical significance by design, they are prone to yield many false positives (Sec. 5). Thus, they are not commonly adopted in hotspot detection.

We propose a novel framework, namely a Nondeterministic Normalization based scan statistic (NN-scan) to address the limitations of spatial scan statistics by explicitly modeling spatial nondeterminism. We also propose a DynamIc Linear Approximation (DILA) algorithm to improve the computational efficiency.

*Department of Computer Science and Engineering, University of Minnesota. {xiexx347, shekhar}@umn.edu

Through detailed experiments under a variety of controlled parameters, we show that the proposed NN-scan framework can significantly improve the precision and recall of hotspot detection. A real-world example is also presented through a case study. In addition, we show that the DILA algorithm can greatly reduce execution time of the baseline NN-scan.

2 Problem Formulation

2.1 Key Concepts

Point distribution: A collection of N geo-located instances (points) of an event (e.g., crime, disease) in a spatial domain.

Point process: A statistical process that generates a point distribution. It determines the probability of each point being located at each location in the study area. A homogeneous point process has identical probability across locations.

Hotspot: A sub-region within the study area that has a higher probability density of generating certain instances (e.g., disease or crime cases) than its outside. Existence of hotspots means that the point process is not homogeneous and is biased towards hotspot regions.

2.2 Formal Problem Definition

Inputs:

(1) A distribution of N geo-located points;
(2) Candidate (sub)region enumeration scheme (e.g., circular, rectangular or linear regions).

Output: Hotspots (if they exist).

Objectives:

(1) Solution quality (e.g., by precision and recall);
(2) Computational efficiency.

There are two key building blocks of hotspot detection. The first is to enumerate candidate regions (e.g., rectangular [8], circular [2], ring-shaped [4], or linear [12]) inside the study area, and the second is to evaluate if a candidate region is a true hotspot.

The scope of the present study focuses on the second building block. Since our work is applicable to general candidate region enumeration algorithms, we consider the choice of enumeration algorithms as an input.

The solution quality of hotspot detection is evaluated via controlled simulations of point processes. For example, a point distribution with true hotspots can be generated using a biased point process (Sec. 2.1), with which we can assess how well the result of a hotspot detection algorithm can match the true hotspots.

3 Limitations of Current Framework

3.1 Spatial scan statistic (SSS) In SSS, the shape of candidate regions is often chosen based on domain science. For example, the most popular circular shape is based on the diffusion theory in epidemiology and ring-shape is from criminology. Given the high popularity of circular shaped hotspots in research and applications, we use circles as the default shape of candidate regions in the rest of the illustrations for simplicity.

Key steps: For a given N point distribution, the spatial scan statistic (SSS) enumerates a list of candidate regions using a region enumeration algorithm (e.g., circular regions of different sizes at different locations). For each candidate region, its score is evaluated using a test statistic (e.g., density, likelihood ratio). The scores represent the quality of the candidates.

Then, to confirm if the score of a candidate is high enough to be a hotspot, SSS uses statistical significance testing. Because hotspots are defined as regions with a higher probability density of generating certain event instances than the outside, the null hypothesis H_0 claims that the point distribution is generated by a homogeneous point process (i.e., no higher probability for any region) and the alternative hypothesis H_1 claims a biased point process (i.e., probability is higher inside hotspots). Since the exact distribution of a test statistic is often unknown, SSS runs Monte Carlo simulation trials to estimate the significance level. In each trial, it generates a random N point distribution under H_0, and uses the same region enumeration algorithm to list all candidates but only records the best test statistic score achieved by the candidates. After a large number of trials (e.g., 1000), it sorts all recorded best scores from the trials (i.e., 1000 of them). Finally, for each candidate from the real (observed) data, if its score is among the top 1% (i.e., corresponding to significance level of 0.01) of the simulated best scores, we are confident that it is very unlikely for a homogeneous point process to form such a candidate region and thus it is a hotspot. In other words, non-hotspots are filtered out by using the minimum of the top 1% scores as a **threshold**.

3.2 Test statistic: Idea of normalization A critical component of SSS is the test statistic, as it is used to score and rank candidates, and directly determines result quality. Current methods use algebraic or exponential functions (e.g., density, density ratio and likelihood ratio) as the test statistic. Among them, likelihood ratio, which is the key contribution in the original SSS [7], remains the most foundational and standard test statistic and is used in the vast majority of SSS methods.

The key idea of the test statistics in SSS is normalization, which makes candidates with different areas and counts (number of points in a candidate region) comparable. This is the basis for significance testing in SSS. For example, denote two candidates as C_1 and C_2,

their areas as a_1 and a_2, and their counts as n_1 and n_2. Without a normalizing test statistic, it is difficult to tell which candidate is better (e.g., when $a_1 > a_2$ and $n_1 > n_2$). The simplest test statistic, density, normalizes the candidates using area, i.e., $d = n/a$. Likelihood ratio (LR) is more statistically advanced by normalizing with the null hypothesis (final form in Eq. (3.1)).

$$LR = \frac{Likelihood(H_1)}{Likelihood(H_0)} = \left(\frac{n}{e}\right)^n \left(\frac{N-n}{N-e}\right)^{N-n} I \tag{3.1}$$

where N and n are the number of points in the entire study area and the candidate region, respectively; $e = N \cdot (a/A)$ is the expected number of points in the candidate region under the null hypothesis (A and a are the area of the study area and the candidate region, respectively); H_1 and H_0 are alternative and null hypotheses; and $I = 1$ if $\frac{n}{a} > \frac{N-n}{A-a}$ and 0 otherwise, is an added indicator function enforcing that a candidate's density is higher than its outside (i.e., not sparse).

Comparing these normalizations, density based test statistics, although intuitive, are known to be biased towards smaller candidate regions [9]. Here we formally show the limit of density in hotspot detection. Denote d^* as the density threshold returned by significance testing (e.g., separating the top 1% and the rest 99% for the significance level of 0.01), and N as the total number of points in the distribution. We have:

THEOREM 3.1. *Density, if used with SSS framework, cannot detect hotspots with an area greater than N/d^*.*

Proof. The proof is straightforward. Given the density definition $d = n/a$, $d > d^*$ means $d^* < n/a \leq N/a$.

While Thm. 3.1 is easy to prove, its implication is significant in practice. For example, for a 10×10 study area with $N = 200$ points, d^* returned by 1000-trial Monte Carlo simulation is $d^* = 40012.1$. This means SSS, when using density as the test statistic, cannot detect hotspots of an area larger than 0.005, which is only 0.005% of the study area. With such bias, it either detects no hotspot or returns extremely small hotspots that are not interesting in applications.

To mitigate this, likelihood ratio based SSS incorporates probabilistic models. Its goal is to assess how much more likely a candidate is formed by the alternative hypothesis H_1 rather than the null hypothesis H_0 (denominator). With such normalization, likelihood ratio is expected to favor candidates that are actual hotspots. This also means it needs to be fair about candidate regions of different areas because actual hotspots could potentially have any area $a \in (0, A]$. In other words, candidate regions of different areas need to have

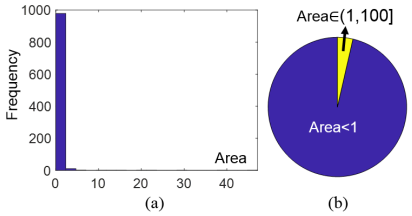

Figure 1: Areas of candidate regions with highest LR.

equal chance of having high likelihood ratios in a completely random point distribution. However, experiment results reveal that candidate regions of very small areas have a much greater chance of having high likelihood ratios. We generated 1000 random distributions of 200 points in a 10×10 space. For each distribution, we recorded the area of the candidate region that achieved the highest likelihood ratio. Fig. 1 shows the frequency of the areas of best candidate regions and over 95% of them are smaller than 1, out of a maximum area of 100. In fact, 90% are smaller than 0.1 and 69% smaller than 0.01. This shows likelihood ratio still has a bias towards smaller candidates (a concrete example in Sec. 4.1).

4 Proposed: Nondeterministic Normalization based Scan Statistic (NN-scan)

4.1 Spatial Nondeterminism First, we examine the theoretical reason behind likelihood ratio's bias. The analysis will be illustrated through a simple example, namely Pattern TINY:

DEFINITION 1. ***Pattern TINY:*** *A region whose area is $0^+ = 1/\infty$ and has only a single point inside.*

As a degenerate case, Pattern TINY is neither an interesting pattern nor hotspot. However, its likelihood ratio ($+\infty$) indeed dominates those of all other candidates. In fact, this likelihood ratio definition leads to a seemingly "**probabilistic paradox**", as shown by the "contradiction" between Propositions 4.1 and 4.2.

PROPOSITION 4.1. *According to the likelihood ratio in Eq. (3.1), the likelihood (i.e., probability) of having Pattern TINY under a homogeneous point process is 0.*

Proof. The likelihood ratio (Eq. (3.1)) of Pattern TINY is $+\infty$. As a probability, the likelihood of $H_1 \in [0, 1]$. Thus, the likelihood of H_0 is 0.

PROPOSITION 4.2. *In any random point distribution, the probability of observing Pattern TINY is 1.*

Proof. One can simply create an infinitely small region around any point in a distribution.

To resolve this "probabilistic paradox", we define two key concepts: best candidate region and spatial nondeterminism. Denote CR_a as the set of all candidate regions of area a and $|CR_a(i)|$ as the number of points inside the i^{th} candidate region in CR_a.

DEFINITION 2. ***Best candidate region:*** *The best candidate region of area a is $CR_a(i^*)$, where $i^* = \underset{i}{\text{argmax}}\,|CR_a(i)|$.*

DEFINITION 3. ***Spatial nondeterminism:*** *The phenomenon that the location of $CR_a(i^*)$ is nondeterministic in point distributions generated by a homogeneous point process (null hypothesis).*

While Proposition 4.1 and 4.2 seem contradictory, there is a critical difference in how the null hypothesis is considered in each. In the current definition of likelihood ratio, the likelihoods are computed assuming that the candidate region is at a fixed location (i.e., a fixed bi-partition of the study area). In other words, after a dense region is found at location loc_x in our observed data, the likelihood of the null hypothesis is computed as the probability of recreating this dense region at loc_x under a homogeneous point process.

If we only consider a region at a fixed location, then indeed, as its area goes to zero, the probability of randomly placing a point in it is zero (Proposition 4.1). However, according to spatial nondeterminism, we do not necessarily have the best candidate region at a fixed location (in fact, it is very unlikely). So this probability going to zero does not mean that this pattern is not from a homogeneous point process.

In contrast, Proposition 4.2 considers spatial nondeterminism (Def. 3). As long as a random point distribution is valid (not empty), we can always find Pattern TINY although its location is nondeterministic.

Spatial nondeterminism explains why Pattern TINY is not statistically meaningful or interesting, which conforms to our intuition. It reveals the core issue in the current use of likelihood ratio in hotspot detection, which is the ignorance of spatial nondeterminism.

4.2 Nondeterministic Normalization The bias in current test statistics mainly leads to issues of incorrect ranking among candidate regions. For example, a non-hotspot may have a higher test statistic score than an actual hotspot. Since in significance testing a threshold of the score (i.e., splitting point of the top 1% scores and the rest 99% from Monte Carlo trials, Sec. 3.1) is used to determine if a candidate region is significant (i.e., true hotspot), incorrect ranking among candidate regions will result in a situation where we either miss true hotspots or include false ones.

The goal of nondeterministic normalization is to incorporate spatial nondeterminism into the test statistic to eliminate such situations and guarantee correctness of ranking. Essentially, significance testing aims to select hotspot candidates that are very unlikely to be formed by chance under a homogeneous point process (i.e., null hypothesis). In this context, we formally define the correctness of ranking in Def. 4. Denote C_1 and C_2 as two candidate regions, a_1 and a_2 as their areas, and n_1 and n_2 as their contained number of points. Denote p_1 as the probability of observing any candidate of area a_1 with at least n_1 points under the null hypothesis, and similarly p_2 as the probability of observing any candidate of area a_2 with at least n_2 points. The correctness of ranking is defined by:

DEFINITION 4. ***Correctness of ranking*** *A test statistic must give a higher score for C_1 than C_2 if $p_1 < p_2$.*

In significance testing, we only need to select a single threshold of test statistic score using the given significance level α (e.g., 0.01). All candidates satisfying it will be returned as hotspots. Thus, based on Def. 4, we define a slightly relaxed version of correctness:

DEFINITION 5. ***Correctness of ranking (relaxed)*** *A test statistic must give a higher score for C_1 than C_2 **only** if $p_1 < \alpha$ and $p_2 \geq \alpha$ (implying $p_1 < p_2$).*

To guarantee correct ranking (Def. 5), our Nondeterministic Normalization Index (NNI) directly evaluates if a candidate region can be formed under the null hypothesis with spatial nondeterminism:

DEFINITION 6. ***Nondeterministic Normalization Index (NNI):***

$$NNI = \frac{f(n, a, N, A)}{f(n^*, a, N, A)} \tag{4.2}$$

$$n^* = \max_x x, \ \textbf{\textit{s.t.}}\ p(x, a) \geq \alpha \tag{4.3}$$

where n and a are the number of points and area of a candidate region, and N and A are the total number of points and the entire area of the study area, α is the significance level, and $p(x, a)$ is the probability of observing at least one candidate region of area a with x or more points (may appear at any locations) in a point distribution under H_0, n^* is the best we can get under the null hypothesis and significance level α, and f is an inner-level statistic (e.g., density) that will be normalized by spatial nondeterminism (i.e., using n^*).

The **key parameter** and difference maker in the NNI is n^*, which gives the best we can get with the null

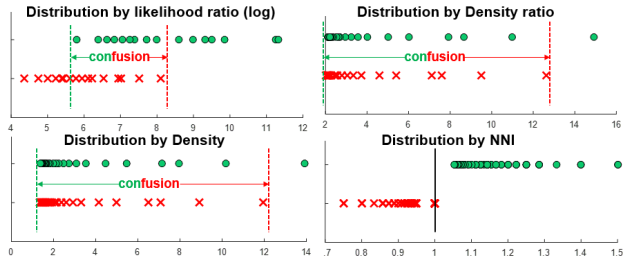

Figure 2: Ranking of candidate regions (best in color).

hypothesis under a desired significance level α. Function f can be any traditional test statistic (e.g., density) but must satisfy a monotonicity property: with fixed (a, N, A), f must increase monotonically as n increases. With this necessary property, we can conclude that:

$$\begin{cases} p(n,a) < \alpha \text{ (e.g., 0.01)}, & \text{if } NNI > 1 \\ p(n,a) \geq \alpha, & \text{otherwise} \end{cases} \tag{4.4}$$

From Eq. (4.4) we can see that the correctness of ranking (Def. 5) is guaranteed by the NNI through n^*. In addition, the threshold of the NNI is always 1 for different significance levels, because significance level α determines the value of n^*. In the rest of the paper, we will by default use $f(n,a,N,A) = n$ for function f (i.e., $NNI = n/n^*$) for illustration purposes; this does not affect our theorems and proofs.

To show the robustness of NNI, Fig. 2 shows a ranking example of four test statistics. The green points represent a set of candidate regions (different areas and counts) with $p(n,a) < 0.01$ and the red crosses represent those with $p(n,a) \geq 0.01$. Each probability is computed experimentally with 1000 Monte Carlo trials. In each trial, we generate a random point distribution under the null hypothesis and then see if a candidate region with area a and at least n points exists. The X-axis in Fig. 2 shows the test statistic scores from low to high. The Y-axis is just for a clear visual separation of the two groups of candidates so that the green points are vertically above the red crosses. The exact values on the Y-axis should be ignored. Based on Def. 5, incorrect ranking happens if the test statistic scores of the red crosses are better (higher) than those of green points.

We can see that current test statistics all contain incorrect rankings, which create confusion zones among candidates. Looking at the extent of the confusion zones, likelihood ratio does have an improvement over density (n/a) and density ratio ($\frac{n/a}{(N-n)/(A-a)}$). Finally, we can see that with NNI the incorrect rankings are eliminated by addressing spatial nondeterminism.

4.3 NN-scan: Baseline Algorithm While NNI maintains ranking correctness, computing it is challenging as there still does not exist a closed-form approach to directly compute or evaluate the statistical distribution of the key parameter n^* (more specifically, $p(x,a)$ in Eq. (4.3), which involves spatial nondeterminism). Thus, we use Monte-Carlo simulation to estimate n^*.

Algorithm 1 shows the Monte Carlo framework for n^* of candidate regions with an area a. In each trial, we generate a random point distribution under H_0 and use the candidate region enumeration scheme (Sec. 2.1) to yield a list of candidate regions of area a across the study area (lines 3, 4). Among them, we find the maximum number of points contained in a single region, and insert this maximum number into a descendingly ordered list $nList$. After all M trials, n^* is estimated as the $(\alpha M)^{th}$ largest value in $nList$, where α is the significance level.

Algorithm 1: Monte Carlo estimation of n^*

Require: • candidate region area a • number of points N • candidate region enumeration scheme enumCR() • significance level α • number of Monte Carlo trials M

1: $nList$ = new List(M)
2: **for** $i = 1$ to M **do**
3: $data_r$ = RandomPointDistribution(N)
4: **for** cr in enumCR($data_r$, a) **do**
5: n_{cr} =cr.getInsidePoints($data_r$).count()
6: $nList(i)$ = max($nList(i)$, n_{cr})
7: **end for**
8: **end for**
9: $nlist$.sort('DESC')
10: **return** $n^* = nList(ceil(\alpha \cdot M))$

Since we can generate an infinite number of different region areas (i.e., continuous variable) of a certain shape (e.g., circles), for computability of NNI, we use a finite-length area vector V_{area} to represent all the areas to be considered. The total number of areas k is given as a user-input. In addition, since the Monte Carlo trials (line 2, Alg. 1) are independent, the baseline algorithm parallelizes them across CPU cores to reduce time cost.

4.4 Acceleration: the DynamIc Linear Approximation (DILA) Algorithm Monte Carlo simulation is the most time-consuming task in NNI computation, since the number of trials M often needs to be large (e.g., 1000, 10000) to accurately estimate the distribution. The goal of DILA is to create tight lower and upper bounds of n^* (Eq. (4.3)) to directly filter out some candidates without Monte Carlo trials. To get tight bounds on n^*, we first need to study the relationship

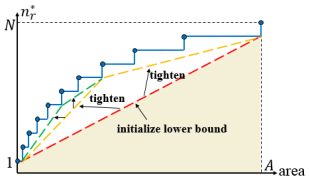

Figure 3: Trend of n^* and its lower bounds.

between n^* (discrete) and region area a (continuous).

DEFINITION 7. ***Break points of area:*** *A break point is an area a, which if we reduce any infinitely small area Δa from it, its corresponding n^* will reduce by 1.*

LEMMA 4.1. *Given an ordered (ascending) list of break points of area $\{a_1, a_2, ..., a_Z\}$, the value of $(a_i - a_{i-1})$ monotonically increases as i increases $\forall i = 2, ..., Z$. (**proof** in Appx. A.1)*

Based on Lemma 4.1, we know n^* increases more slowly as region area a increases, as shown in Fig. 3. This "slower-growing" trend of n^* also conforms to the intuition that, no matter how large area a is, n^* is at maximum N, which is the total number of points in the study area. This means n^* reaches a growing speed of zero at the end. Based on Lemma 4.1, we further develop the lower and upper bounds for n^* of area a:

THEOREM 4.1. *Given two areas a_i, a_j (not necessarily break points) and their corresponding n^* values n_i^*, n_j^*, the n^* value of any area $a \in [a_i, a_j]$ is lower bounded by: $LB(n^*) = \lfloor n_i^* + \frac{a-a_i}{a_j-a_i}(n_j^* - n_i^*) \rfloor$. (**proof** in Appx. A.2)*

THEOREM 4.2. *Given two areas a_i and a_j (not necessarily break points) satisfying conditions: (1) $sign(a_i - a) = sign(a_j - a)$, (2) $|a_i - a| > |a_j - a|$, and (3) $|n_i^* - n_j^*| \geq 1$, we have n^* upper bounded by: $UB(n^*) = n_i^* + \frac{a-a_i}{a_j-a_i}(n_j^* + 1 - n_i^*)$. (**proof** in Appx. A.3)*

For the upper bound, we evaluate $UB(n^*)$ of area a from two sides (i.e., $a_i > a_j > a$ and $a_i < a_j < a$) and take the minimum of the two. With upper and lower bounds, exact n^* values only need to be computed if the observation $n \in (LB(n^*), UB(n^*))$.

DynamIc Linear Approximation (DILA): The idea is to dynamically tighten the bounds as more exact n^* values are computed. In the following we use lower bound initialization and update rules to illustrate DILA; the same strategy also applies to upper bounds.

To initialize the lower bounds for all areas, one can compute exact n^* values of the smallest and largest areas, and then estimate the lower bounds for other areas using Thm. 4.1. This is shown by the red dash-line in Fig. 3. While candidates in the region below can be filtered out, there is still a large gap with the exact values. To tighten this initial bound, DILA starts with a **three-point lower bound construction**. Compared to the previous strategy, this construction involves the n^* value of an extra area: the median area of the area-vector (i.e., the vector with k areas to be enumerated). This three-point based lower bound is approximated by the yellow dash-line (Fig. 3), which greatly narrows the gap with the exact values (i.e., blue line).

As DILA progresses, whenever a new exact n^* value is computed for a new area a_{new}, the algorithm finds a_{new}'s two nearest areas a_1 and a_2 whose exact n^* values are already computed. Formally, denote S_a as the set of areas with exact n^* values computed. We have:

$$\begin{cases} a_1 = \max_{a_x}\{a_x | a_x < a_{new}, a_x \in S_a\} \\ a_2 = \min_{a_x}\{a_x | a_x > a_{new}, a_x \in S_a\} \end{cases} \tag{4.5}$$

Then, DILA updates the lower bound $LB(n_x^*)$ of each area a_x ($a_x \notin S_a$) by $LB(n_x^*) =$:

$$\begin{cases} \lfloor n_1^* + \frac{a_x - a_1}{a_{new} - a_1}(n_{new}^* - n_1^*) \rfloor, & \text{if } a_x \in (a_1, a_{new}) \\ \lfloor n_{new}^* + \frac{a_x - a_{new}}{a_2 - a_{new}}(n_2^* - n_{new}^*) \rfloor, & \text{if } a_x \in (a_{new}, a_2) \end{cases}$$

where n_1^*, n_2^*, n_{new}^* are the n^* values of a_1, a_2 and a_{new}.

In Fig. 3, the green dash-lines show one update after the lower bound initialization. For this example, the lower bound is already very close to the exact values. Updates on upper bounds apply the same idea.

Time complexity: Denote N as total number of points, k as area vector size, M as the number of Monte-Carlo simulation trials and λ as the number of CPU cores. Since the candidate region enumeration scheme is an input, we denote its time complexity for enumerating all regions of an area as $r(N)$. Assuming f in Eq. (4.2) can be computed in $O(1)$ time (e.g., simple density), we have the time complexity of the baseline algorithm as $O((M \cdot r(N)/\lambda + M \log M + k) \cdot k)$. Since the core of DILA is to reduce the number of exact n^* computations on different areas, we denote the number of pruned areas as k'. The time complexity of DILA is thus $O((M \cdot r(N)/\lambda + M \log M + k)(k - k'))$.

5 Validation

We evaluated NN-scan through detailed controlled experiments and a real-world example.

5.1 Controlled experiments of solution quality

The solution quality of hotspot detection can be experimentally evaluated using controlled synthetic data: (1) point distributions generated by biased point processes, in which we artificially insert true hotspots (i.e., higher probability density than outside); (2) point distributions by a homogeneous point process, which does

not contain any hotspot. The former evaluates how well an algorithm can detect the true hotspots, and the latter tests if an algorithm is robust against false positives (i.e., dense regions created by random chance under H_0).

Parameter setting: We considered a variety of parameters in synthetic data generation: (1) Number of points N; (2) Effect size es: for a hotspot, this represents how many times the probability density inside is higher than outside; (3) Radius of hotspot r (circular shape is used in the experiments since it is based on diffusion theory and is used in most real-world applications); (4) Number of hotspots h. To evaluate the effect of a single parameter, we kept the rest fixed at their default values ($N = 400$, $es = 3$, $r = 1$ (dimension of study area is 10×10) and $h = 3$) and varied that single parameter through a series of experiments.

We implemented and included five candidate methods for comparison: (1) Proposed NN-scan with significance level $\alpha = 0.01$, (2) Spatial scan statistic (SSS) with likelihood ratio and $\alpha = 0.01$ (performances of density and density ratio based versions were also evaluated but their results were very poor), (3) DB-6: DBSCAN [5] with density threshold $\epsilon = 0.6$, (4) DB-9: DBSCAN with $\epsilon = 0.9$ and (5) DB-12: DBSCAN with $\epsilon = 1.2$. For DBSCAN, the minimum number of points $MinPts$ was set to 5% of the total number of points, which is the best threshold that we found with the above ϵ values in our experiments (additional results in Appendix C).

Fig. 4 shows the precision ($\frac{|detections \cap true|}{|detections|}$), recall ($\frac{|detections \cap true|}{|true|}$) and F1-scores ($\frac{2}{precision^{-1}+recall^{-1}}$) of the candidate methods under a variety of scenarios. Each performance statistic (e.g., precision) value was computed using 100 repeated runs (i.e., a summary of detection results from 100 point distributions generated by the same parameter set (N, es, r, d)). Since the geometric shape of hotspots used in the experiments was circular, we converted DBSCAN results to circles by using the mean coordinates of a cluster as the center and then generating a minimum bounding circle.

The **general trend** is that NN-scan overall maintained the highest F1-scores under different parameter settings except in very few cases (e.g., very small effect size). With significance testing, NN-scan and SSS kept the highest precision among the methods. While DB-9 and DB-12 were able to reach the best recall, they mostly had very low precision, leading to low F1-scores. In addition, the rank of DBSCAN methods changed across experiments, showing its sensitivity to the thresholds. Compared to SSS, NN-scan achieved much higher recall in general (e.g., above 20% in many scenarios).

Effect of number of points N (Fig. 4(a)): The solution quality of NN-scan and SSS gradually improved

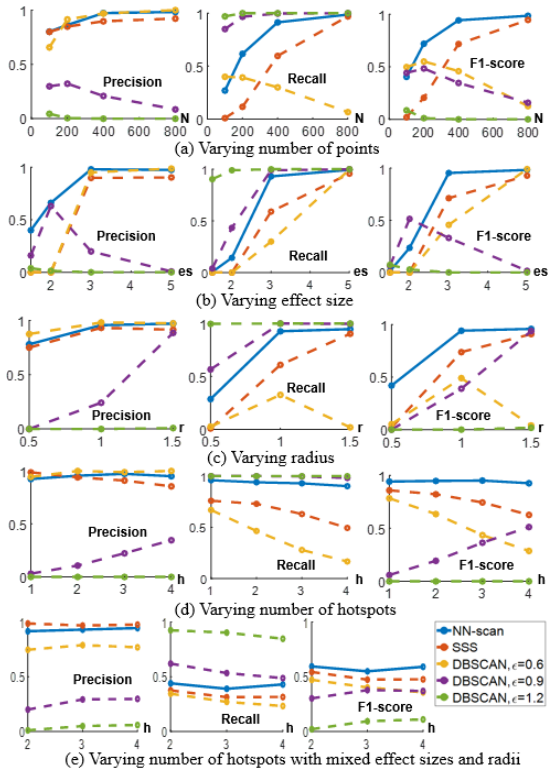

Figure 4: Precision, recall and F1-scores. (best in color)

as the number of points increased. This is expected statistically as it is easier to confirm hotspots with more observed samples. The major dominance zone of NN-scan is at smaller numbers of points where we can see a $30\% - 40\%$ difference. This is meaningful for many societal applications because it is better to confirm outbreaks of disease, crime etc. at an earlier stage to reduce their effects. In addition, many real world applications do not always have large numbers of samples, such as transportation-related fatalities and major crimes (e.g., arson) within a city. For DBSCAN methods, the performance dropped with more samples, because more false positives were detected when the density of data increased. This again shows that DBSCAN's performance is very sensitive to thresholds.

Effects of effect size es **and hotspot size** r (Fig. 4(b) and (c)): The two parameters determine how likely a point will emerge within a hotspot. Note that es must be greater than 1 for hotspots to exist; otherwise the probability density is not biased in the study area. For hotspot size (i.e., radius r), when it is small its corresponding probability mass may also be small despite good effect sizes. This means it can be very difficult to detect hotspots with low es or r values and confirm their statistical significance. On the **bright**

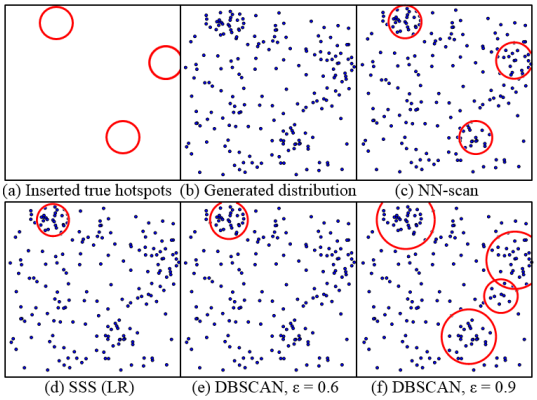

Figure 5: Example results when true hotspots exist.

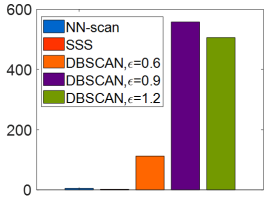

Figure 6: Number of mistakenly detected hotspots.

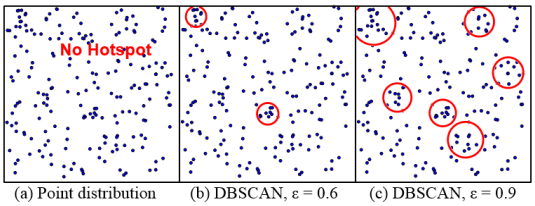

Figure 7: Detection results on data generated by homogeneous point process (i.e., no hotspot). Both NN-scan and SSS did not output any hotspot (same as (a)).

side, if NN-scan or SSS find hotspots of small size, we can have higher confidence on them as the effect size is likely to be very large (especially when N is small). The results of DBSCAN methods were not robust (e.g., DB-9 worked well for large r but poor for large es).

Effect of number of hotspots h (Fig. 4(d)): NN-scan maintained high solution quality consistently for different h in the experiment. SSS faced difficulty for larger h values. DB-6 and DB-9 only worked okay here for specific values of h. DB-12 did not work well.

Mixed parameters (Fig. 4(e)): The mixed setup was implemented for each different number of hotspots h. For each h, the biased point process uses different es and r (randomly chosen from $es \in \{1.5, 2, 3, 5\}$ and $r \in \{0.5, 1, 1.5\}$) for different hotspots. In a mixed scenario, it is more difficult to detect hotspots with relatively smaller probability mass, which led to a decrease in general solution quality. NN-scan was able to maintain the best performance consistently in the experiment.

Besides quantitative metrics, Fig. 5 visualizes how the methods compare on an example point distribution (more in Appendix B) with artificially inserted true hotspots. DB-12's result was poor and skipped.

Robustness against false positives: Due to the high-cost nature of false alarms in real world applications (e.g., disease outbreak), it is critical for hotspot detection methods to filter out "dense" regions formed by random chance under a homogeneous point process (i.e., no hotspot). Fig. 6 shows the number of false positives detected by the methods across 100 point distributions generated by a homogeneous point process. Both NN-scan and SSS had very few false positives with significance testing. In contrast, DBSCAN based methods all resulted in hundreds of false detections. This limits the use of DBSCAN in real world hotspot detection. Fig. 7 visualizes an example point distribution without hotspot. Both NN-scan and SSS did not return any hotspot whereas DBSCAN gave multiple false positives.

5.2 Real-world Example: Crime Hotspots We also evaluated NN-scan on a real-world Motor-Part Theft dataset from the Minneapolis (USA) Police Department. Fig. 8(a) shows the 124 instances displayed on top of the city map. We picked a small dataset since one advantage of NN-scan over SSS is on finding significant hotspots with small data. We can see there is a dense region around the middle of the map (i.e., covering downtown), and another at the bottom right (next to a lake, a river park and a major airport). As we can see in Fig. 8(c), NN-scan was able to identify the two hotspots under significance level 0.01. The NNI values of both hotspots indicates that they cannot be recreated under the null hypothesis. By contrast, using SSS (Fig. 8(b)), we were able to find the larger hotspot around downtown but missed the other hotspot. This result is consistent with our earlier experiments (Sec. 5.1). We also expect the smaller hotspot to have a fairly large effect-size based on our analysis with synthetic data.

5.3 Execution-time Experiments The baseline algorithm includes a multi-core parallelization of the Monte-Carlo simulation. The same is applied to DILA. The experiments were performed on a 16-core node (Intel Haswell E5-2680v3 processor). Fig. 9 shows the time comparison between baseline and DILA algorithms. The top row shows the performance on data with true hotspots (more difficult to prune), and the bottom on data without hotspots. Within each row, the two charts show the effects of total number of points N

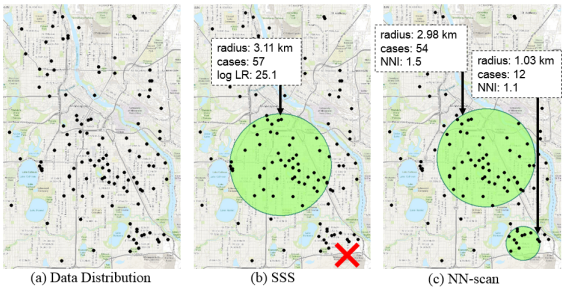

Figure 8: Statistically significant crime hotspots.

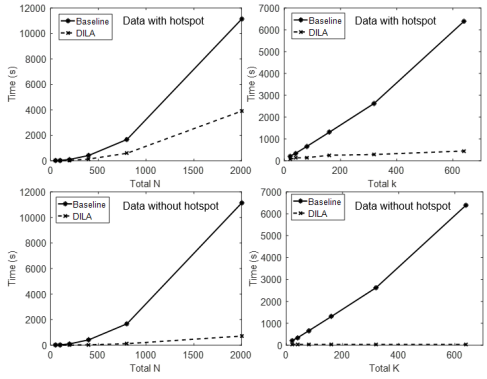

Figure 9: Execution time: Baseline vs. DILA.

(left) and the size k of the area-vector (right), respectively. When evaluating effect on N, k was fixed to 40. For effect on k, N was fixed to 400. The number of Monte-Carlo trials was 1,000 for both. The overall trend is that DILA ran much faster than the baseline algorithm on both data (with and without hotspots). In addition, the improvements were greater for data without hotspots (bottom row). When there is no hotspot, it is less likely for a region to have a number of points exceeding the tight lower bound. Also, while the core of DILA is pruning, its execution time did not increase linearly as k increased. An insight is that, in practice, DILA may only need a small number of exact n^* values to tighten the lower bounds (e.g., Fig. 3).

6 Conclusions and Future Work

We showed the limitations of current theoretical foundations of hotspot detection, and proposed a NN-scan framework to address them by explicitly modeling spatial nondeterminism. We also proposed a DILA algorithm to improve its efficiency. Experiments showed that NN-scan greatly improved solution quality and DILA greatly reduced time cost. In future work, we will explore the use of NN-scan with DBSCAN to leverage its ability to enumerate arbitrarily shaped zones while maintaining statistical robustness. We will also study the use of NN-scan with polygon input data and the effects of the f function in NNI. Other new opportunities include temporal, network and Poisson NN-scan, etc.

Appendices and code are shared at: https://www-users.cs.umn.edu/%7exiexx347/nnscan.html.

7 Acknowledgments

This work is supported by the US NSF under Grants No. 1737633, 1541876, 1029711, IIS-1320580, 0940818 and IIS-1218168, the USDOD under Grants HM0210-13-1-0005, ARPA-E under Grant No. DE-AR0000795, USDA under Grant No. 2017-51181-27222, NIH under Grant No. UL1 TR002494, KL2 TR002492 and TL1 TR002493 and the OVPR U-Spatial and Minnesota Supercomputing Institute at the University of Minnesota.

References

[1] National Cancer Institute, Surveillance Research Program. https://surveillance.cancer.gov/satscan/, 2017.
[2] SaTScan. https://www.satscan.org/, 2017.
[3] R. Assuncao, M. Costa, A. Tavares, and S. Ferreira. Fast detection of arbitrarily shaped disease clusters. *Statistics in medicine*, 25(5):723–742, 2006.
[4] E. Eftelioglu et al. Ring-shaped hotspot detection: a summary of results. In *Data Mining (ICDM), 2014 IEEE Intl. Conf. on*, pages 815–820. IEEE, 2014.
[5] M. Ester et al. A density-based algorithm for discovering clusters a density-based algorithm for discovering clusters in large spatial databases with noise. In *Proc. of the 2nd Intl. Conf. on Knowledge Discovery and Data Mining*, KDD'96, pages 226–231, 1996.
[6] I. Jung et al. A spatial scan statistic for multinomial data. *Stats. in Med.*, 29(18):1910–1918, 2010.
[7] M. Kulldorff. A spatial scan statistic. *Comm. in Statistics-Theory and methods*, 26(6):1481–1496, 1997.
[8] 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 (NIPS)*, pages 651–658, 2004.
[9] D. B. Neill and A. W. Moore. Rapid detection of significant spatial clusters. In *Proc. ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining*, pages 256–265, 2004.
[10] S. K. Prasad et al. Parallel processing over spatial-temporal datasets from geo, bio, climate and social science communities: A research roadmap. In *Big Data, 2017 IEEE Intl. Congress on*, pages 232–250.
[11] S. Shekhar, S. Feiner, and W. Aref. Spatial computing. *Communications of the ACM*, 59(1):72–81, 2015.
[12] X. Tang et al. Significant linear hotspot discovery. *IEEE Trans. on Big Data*, 3(2):140–153, 2017.
[13] Y. Xie et al. Transdisciplinary foundations of geospatial data science. *ISPRS International Journal of Geo-Information*, 6(12):395, 2017.