Computing Approximate Statistical Discrepancy

Michael Matheny
University of Utah, Salt Lake City, USA
mmath@cs.utah.edu

Jeff M. Phillips[1]
University of Utah, Salt Lake City, USA
jeffp@cs.utah.edu

Abstract

Consider a geometric range space $(X, \mathcal{A})$ where X is comprised of the union of a red set R and blue set B. Let $\Phi(A)$ define the absolute difference between the fraction of red and fraction of blue points which fall in the range A. The maximum discrepancy range $A^* = \arg\max_{A \in (X,\mathcal{A})} \Phi(A)$. Our goal is to find some $\hat{A} \in (X, \mathcal{A})$ such that $\Phi(A^*) - \Phi(\hat{A}) \leq \varepsilon$. We develop general algorithms for this approximation problem for range spaces with bounded VC-dimension, as well as significant improvements for specific geometric range spaces defined by balls, halfspaces, and axis-aligned rectangles. This problem has direct applications in discrepancy evaluation and classification, and we also show an improved reduction to a class of problems in spatial scan statistics.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Scan Statistics, Discrepancy, Rectangles

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2018.32

Related Version A full version of the paper is available at [17], `https://arxiv.org/abs/1804.11287`.

Acknowledgements Meg Rosales for her editing and Praful Agarwal for many discussions.

1 Introduction

Let X be a set of m points in $\mathbb{R}^d$ for constant d. Let $X = R \cup B$ be the union (possibly not disjoint) of two sets R, the red set, and B, the blue set. Also consider an associated range space $(X, \mathcal{A})$; we are particularly interested in range spaces defined by geometric shapes such as rectangles in $\mathbb{R}^d$ $(X, \mathcal{R}_d)$, disks in $\mathbb{R}^2$ $(X, \mathcal{D})$, and d-dimensional halfspaces $(X, \mathcal{H}_d)$.

Let $\mu_R(A) = |R \cap A|/|R|$ and $\mu_B(A) = |B \cap A|/|B|$ be the fraction of red or blue points, respectively, in the range A. We study the discrepancy function $\Phi_X(A) = |\mu_R(A) - \mu_B(A)|$, when for brevity is typically write as just $\Phi(A)$. A typical goal is to compute the range $A^* = \arg\max_{A \in \mathcal{A}} \Phi(A)$ and value $\Phi^* = \Phi(A^*)$ that maximizes the given function Φ. Our goal is to find a range $\hat{A}_\varepsilon$ that satisfies $\Phi(\hat{A}_\varepsilon) \geq \Phi(A^*) - \varepsilon$.

The exact version of this problem arises in many scenarios, formally as the classic discrepancy maximization problem [3, 7]. The rectangle version is a core subroutine in algorithms ranging from computer graphics [8] to association rules in data mining [9]. Also, for instance, in the world of discrepancy theory [20, 6], this is the task of evaluating how large the discrepancy for a given coloring is. For the halfspace setting, this maps to the minimum disagreement problem in machine learning (i.e., building a linear classifier) [16]. When Φ is

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

replaced with a statistically motivated form [12, 13], then this task (typically focusing on disks or rectangles) is the core subroutine in the GIScience goal of computing the spatial scan statistic [11, 22, 2, 1] to identify spatial anomalies. Indeed this statistical problem can be reduced the approximate variant with the simple discrepancy maximization form [2].

The approximate versions of these problems are often just as useful. Low-discrepancy colorings [20, 6] are often used to create the associated ε-approximations of range spaces, so an approximate evaluation is typically as good. It is common in machine learning to allow ε classification error. In spatial scan statistics, the approximate versions are as statistically powerful as the exact version and significantly more scalable [19].

While the exact versions take super-linear polynomial time in m, e.g., the rectangle version with linear functions takes $\Omega(m^2)$ time conditional on a result of Backurs *et al.* [3], we show approximation algorithms with $O(m + \mathsf{poly}(1/\varepsilon))$ runtime. This improvement is imperative when considering massive spatial data, such as geotagged social media, road networks, wildlife sightings, or population/census data. In each case the size m can reach into the 100s of millions.

While most prior work has focused on improving the polynomials on the exact algorithms for various shapes [14, 25] or on using heuristics to ignore regions [28, 22], little work exists on approximate versions. These include [1] which introduced generic sampling bounds, [19] which showed that a two-stage random sampling can provide some error guarantees, and [27] which showed approximation guarantees under the Bernoulli model. In this paper, we apply a variety of techniques from combinatorial geometry to produce significantly faster algorithms; see Table 1.

Our results. Our work involves constructing a two-part coreset of the initial range space $(X, \mathcal{A})$; it approximates the ground set X *and* the set of ranges $\mathcal{A}$. This needs to be done in a way so that ranges can still be effectively enumerated and $\mu_R(A)$ and $\mu_B(A)$ values tabulated. We develop fast coreset *constructions*, and then extend and adapt exact scanning algorithms to the sparsified range space.

We develop notation and review known solutions in Section 2; also see Table 1. Then we describe a general sampling result in Section 3 for ranges with bounded VC-dimension. In particular, many of these results can be seen as formalizations and refinements (in theory and practice) of the two-stage random sampling ideas introduced in [19].

In Section 3.1 we describe improvements for halfspaces and disks. The details, defer to the full version [17], first improve upon the sampling analysis to approximate ranges $\mathcal{H}_2$. By carefully annotating and traversing the dual arrangement from the approximate range space, we improve further upon the general construction.

Then in Section 4 we describe our improved results for rectangles. We significantly extend the exact algorithm of Barbay *et al.* [4] and obtain an algorithm that takes $O(m + \frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})$. This is improved to $O(m + \frac{1}{\varepsilon^2}\log\log\frac{1}{\varepsilon})$ with some more careful analysis in the full version [17]. This nearly matches a new conditional lower bound of $\Omega(m + \frac{1}{\varepsilon^2})$, assuming current algorithms for APSP are optimal [3].

In Section 5 we show how to approximate a *statistical discrepancy function* (SDF, defined in Section 5) Φ, as well as any *general* function Φ. These require altered scanning approaches and the SDF-approximation requires a reduction to a number of calls to the generic ("linear") Φ. We reduce the number of needed calls to generic Φ functions from $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ [2] to $O(\frac{1}{\sqrt{\varepsilon}})$.

Finally, in Section 6 we show on rectangles strong *empirical* improvement over state of the art [19].

Table 1 Algorithm times for (ε-approximately) maximizing different range spaces. Here dimension d, VC-dimension ν, and probability of failure are all constants. For $(X, \mathcal{R}_2)$ we show it takes $\Omega(m + 1/\varepsilon^2)$ time, assuming hardness of APSP.

		Known Exact	Known Approx [19]	**New Runtime Bounds**
General Range Space		$O(m^{\nu+1})$	–	$O\left(m + \frac{1}{\varepsilon^{\nu+2}} \log^{\nu} \frac{1}{\varepsilon}\right)$
Halfspaces	$\mathbb{R}^d$	$O(m^d)$ [8]	–	$O\left(m + \frac{1}{\varepsilon^{d+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Disks	$\mathbb{R}^2$	$O(m^3)$ [8]	$O(m + \frac{1}{\varepsilon^4} \log^3 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{3+1/3}} \log^{2/3} \frac{1}{\varepsilon}\right)$
Rectangles	$\mathbb{R}^2$	$O(m^2)$ [4]	$O(m + \frac{1}{\varepsilon^4} \log \frac{1}{\varepsilon})$ [2, 1]	$O(m + \frac{1}{\varepsilon^2} \log\log \frac{1}{\varepsilon})$
Rectangles (SDF)	$\mathbb{R}^2$	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$	$O\left(m + \frac{1}{\varepsilon^{2.5}}\right)$
Rectangles (General)	$\mathbb{R}^2$	$O(m^4)$	$O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon}))$	$O\left(m + \frac{1}{\varepsilon^4}\right)$

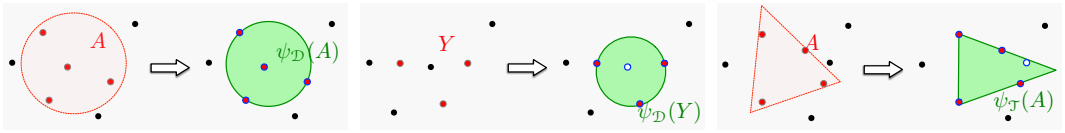


Figure 1 First two panels show that $(\mathbb{R}^2, \mathcal{D})$ has a conforming map $\psi_{\mathcal{D}}$ defined by the smallest enclosing disk. The last panel shows a range space $(X, \mathcal{T})$ corresponding to triangles, and that a mapping $\psi_{\mathcal{T}}$ defined by minimum area triangle is not conforming; it does not recover A.

2 Background on Geometric Range Spaces

To review, a range space $(X, \mathcal{A})$ is composed of a ground set X (for instance a set of points in $\mathbb{R}^d$) and a family of subsets $\mathcal{A}$ of that set. In this paper we are interested in geometrically defined range spaces $(X, \mathcal{A})$, where $X \subset \mathbb{R}^d$. We formalize the requirements of this geometry via a conforming geometric mapping ψ; see Figure 1. Specifically, it maps from a subset $Y \subset X$ to subset of $\mathbb{R}^d$. Typically, the result is a Lebesgue measureable subset of $\mathbb{R}^d$, for instance $\psi_{\mathcal{D}}(Y)$, defined for disk range space $(X, \mathcal{D})$, could map to the smallest enclosing disk of Y.

We say this mapping $\psi_{\mathcal{A}}$ is *conforming to* $\mathcal{A}$ if for any $N \subset X$ it has the properties:

- for any subset $A \in (N, \mathcal{A})$ then $\psi_{\mathcal{A}}(A) \cap N = A$ [*the mapping recovers the same subset*]
- for any subset $Y \subset X$ then $\psi_{\mathcal{A}}(Y) \cap X \in (X, \mathcal{A})$ [*the mapping is always in* $(X, \mathcal{A})$]

2.1 Basic Combinatorial Properties of Geometric Range Spaces

We highlight two general combinatorial properties of geometric range spaces. These are critical in sparsification of the data and ranges, and enumeration of the ranges.

Sparsification. An *ε-sample* $S \subset X$ of a range space $(X, \mathcal{A})$ preserves the density for all ranges as $\max_{A \in \mathcal{A}} |\frac{|X \cap A|}{|X|} - \frac{|S \cap A|}{|S|}| \leq \varepsilon$. An *$\varepsilon$-net* $N \subset X$ of a range space $(X, \mathcal{A})$ hits large ranges, specifically for all ranges $A \in \mathcal{A}$ such that $|X \cap A| \geq \varepsilon |X|$ we guarantee that $N \cap A \neq \emptyset$. Consider range space $(X, \mathcal{A})$ with VC-dimension ν. Then a random sample $S \subset X$ of size $O(\frac{1}{\varepsilon^2}(\nu + \log \frac{1}{\delta}))$ is an ε-sample with probability at least $1 - \delta$ [26, 15]. Also a random sample $N \subset X$ of size $O(\frac{\nu}{\varepsilon} \log \frac{1}{\varepsilon\delta})$ is an ε-net with probability at least $1 - \delta$. For our ranges of interest, the VC-dimensions of $(X, \mathcal{H}_d)$, $(X, \mathcal{D})$, and $(X, \mathcal{R}_d)$ are d, 3, and $2d$.

Enumeration. For the ranges spaces we will consider that each range can be defined by a *basis* B; where B is a point set. Given a geometric conforming map ψ and subset Y, a range space's basis $B \subset Y$ is such that $\psi(B) = \psi(Y)$, but on a strict subset $B' \subset B$, then $\psi(B')$

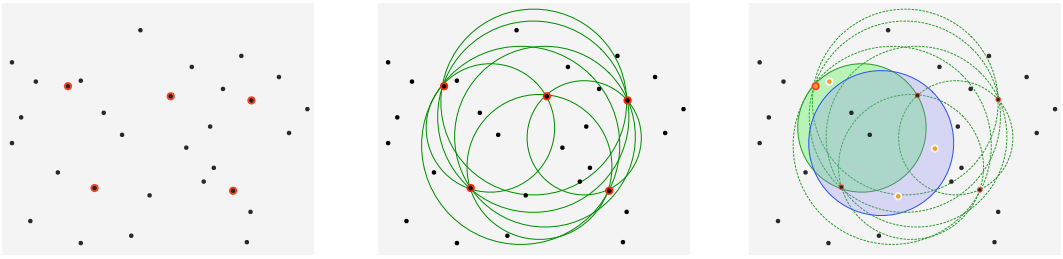


Figure 2 First panel shows $N \subset X$. Second panel shows the set of disks $\{\psi_{\mathcal{D}}(A) \mid A \in (N, \mathcal{D}_{|N})\}$ induced by N. The third panel shows a range $Y \subset X$ (defined by disk in blue). It has symmetric difference over X (in orange) of size 4 with the one defined by the disk $\psi_{\mathcal{D}}(A)$ (in green) induced by a subset $A \subset (N, \mathcal{D}_{|N})$.

is different (and usually smaller under some measure) than $\psi(B)$. We will use β to denote the maximum size of the basis for any subset $Y \subset X$. For instance for $\psi_{\mathcal{D}}$ then $\beta = 3$, for $\psi_{\mathcal{R}_d}$ then $\beta = 2d$, and for $\psi_{\mathcal{H}_d}$ then $\beta = d$. Recall, by Sauer's Lemma [23], if a range space $(X, \mathcal{A})$ has VC-dimension ν, then $\beta \leq \nu$.

This implies that for $m = |X|$ points, there are at most $\binom{m}{\beta} = O(m^\beta)$ different ranges to consider. We assume β is constant; then it is possible to construct $\psi(Y)$ in $O(|Y|)$ time, and to determine if $\psi(Y)$ contains a point $x \in X$ in $O(1)$ time. This means we can enumerate all $O(m^\beta)$ possible bases in $O(m^\beta)$ time, construct their maps $\psi(B)$ in as much time, and for all of them count which points are inside, and evaluate each $\Phi(A)$ to find A^*, in $O(m^{\beta+1})$ time.

For the specific range spaces we study, the time to find $A^* \in \mathcal{A}$ can be improved by faster enumeration techniques. For $\mathcal{H}_d$, Dobkin and Eppstein [7] reduced the runtime to find A^* from $O(m^{d+1})$ to $O(m^d)$; this implies for $\mathcal{D}$ the runtime is reduced from $O(m^4)$ to $O(m^3)$. For $\mathcal{R}_d$, Barbay *et al.* [4] show how to find A^* in $O(m^d)$ time; this was recently shown tight [3] in $\mathbb{R}^2$, assuming APSP takes cubic time.

2.2 Coverings

Our main approach towards efficient approximate range maximization, is to sparsify the range space $(X, \mathcal{A})$. This will have two parts. The first is simply replacing X with an ε-sample. The second is sparsifying the ranges $\mathcal{A}$, using a concept we refer to as an ε-covering.

Recall that the symmetric difference of two sets $A \triangle B$ is $(A \cup B) \setminus (A \cap B)$. Define an *$\varepsilon$-covering* $(X, \mathcal{A}_\triangle)$ of a range space $(X, \mathcal{A})$ where $(X, \mathcal{A}_\triangle) \subset (X, \mathcal{A})$, so that for any $A \in \mathcal{A}$ there exists a $A' \in \mathcal{A}_\triangle$ such that $|A \triangle A'| \leq \varepsilon |X|$. See Figure 2 for an illustration of this concept. If a range space satisfies the above condition for any one specific range A, but not necessarily all ranges $A \in \mathcal{A}$ simultaneously, then it is a *weak ε-covering* of $(X, \mathcal{A})$.

We will use subsets of the ground set to define subsets of the ranges. For a subset $N \subset X$, let $\mathcal{A}_{|N} = \{A \cap N \mid A \in \mathcal{A}\}$ be the restriction of $\mathcal{A}$ to the points in N. We will define $(X, \mathcal{A}_\triangle)$ using $\mathcal{A}_{|N}$ or a subset thereof. However, as each $A \in \mathcal{A}_{|N}$ is a subset of N, which itself is a subset of X, we need a conforming map $\psi_{\mathcal{A}}$ to take a region $A \in \mathcal{A}_\triangle$ and map it back to some region in $\mathcal{A}$, a subset of X. Given $\mathcal{A}'_{|N}$ (which is $\mathcal{A}_{|N}$ or a subset) we define $(X, \mathcal{A}_\triangle)$ as

$$(X, \mathcal{A}_\triangle) = \{X \cap \psi_{\mathcal{A}}(A) \mid A \in (N, \mathcal{A}'_{|N})\}.$$

A small sized ε-covering is implied by a result of Haussler [10]. For every range space $(X, \mathcal{A})$ of VC-dimension ν, with $m = |X|$, there always exist a maximal set of ranges $A_\triangle$ of size $O((\frac{m}{k+\nu})^\nu)$ where for every pair of ranges $A, A' \in A_\triangle$ the symmetric difference $|A \triangle A'| \geq k$. Setting $k = m\varepsilon$ then $(\frac{m}{k+\nu})^\nu = O(\frac{1}{\varepsilon^\nu})$, so $A_\triangle$ is an ε-covering.

Symmetric difference nets. We can construct an ε-net over the symmetric difference range space of $\mathcal{A}$ and then use these points to define $\mathcal{A}_\triangle$.

For a family of ranges $\mathcal{A}$, let $\mathcal{S}_\mathcal{A}$ be the family of ranges made up of the symmetric difference of ranges of $\mathcal{A}$. Specifically $\mathcal{S}_\mathcal{A} = \{A_1 \triangle A_2 \mid A_1, A_2 \in \mathcal{A}\}$. If range space $(X, \mathcal{A})$ has VC-dimension ν, then $(X, \mathcal{S}_\mathcal{A})$ has VC-dimension at most $O(\nu \log \nu)$ [21]. Thus for constant ν we can use asymptotically the same size random sample as before. Matheny *et al.* [19] pointed out two important properties connecting nets over symmetric difference range spaces and ε-coverings and then finding $\hat{A}_\varepsilon$.

(P1) An ε-net N for $(X, \mathcal{S}_\mathcal{A})$ induces $(N, \mathcal{A}_{|N})$ which is an ε-covering of $(X, \mathcal{A})$ [19].

(P2) Given an $\frac{\varepsilon}{2}$-covering $(N, \mathcal{A}_\triangle)$ and an $\frac{\varepsilon}{2}$-sample S over $(X, \mathcal{A})$ then for any range $A \in (X, \mathcal{A})$, there exists a range $\psi_\mathcal{A}(A') \cap X$ for $A' \in \mathcal{A}_{|N}$ so $\left| \frac{|A \cap X|}{|X|} - \frac{|\psi_\mathcal{A}(A') \cap S|}{|S|} \right| \leq \varepsilon$ [19].

For an appropriate constant C, by constructing (ε/C)-nets N_R and N_B, of size n, on the red $(R, \mathcal{S}_\mathcal{A})$ and blue $(B, \mathcal{S}_\mathcal{A})$ points, also constructing (ε/C)-samples of size s on $(R, \mathcal{A})$ and $(B, \mathcal{A})$, and invoking (P2) on the results, Matheny *et al.* [19] observed we can maximize $\Phi(\psi_\mathcal{A}(A') \cap S)$ over $A' \in \mathcal{A}_{|N_R} \cup \mathcal{A}_{|N_B}$ to find an ε-approximate $\hat{A}_\varepsilon$. They construct the ε-nets and ε-samples using random sampling, and apply the results to scan disk $\mathcal{D}$ and rectangle $\mathcal{R}_2$ range spaces towards finding $\hat{A}_\varepsilon$. Enumerating all ranges in $A' \in \mathcal{A}_{|N_R} \cup \mathcal{A}_{|N_B}$ and counting the intersections with the (ε/C)-samples, when C is a constant, is sufficient to find an $\hat{A}_\varepsilon$ in time $O(m + |N|^2|S| \log n) = O(m + \frac{1}{\varepsilon^4} \log^3 \frac{1}{\varepsilon})$ for disks $(X, \mathcal{D})$ and time $O(m + |N|^4 + |S| \log n) = O(m + \frac{1}{\varepsilon^4} \log^4 \frac{1}{\varepsilon})$ for rectangles $(X, \mathcal{R}_2)$.

We can ignore the distinct red and blue points, and focus on three aspects of this problem which can be further optimized: (1) More efficiently constructing a sparse set of ε-covering ranges $(X, \mathcal{A}_\triangle)$. (2) More efficiently constructing a smaller ε-sample S of $(X, \mathcal{A})$. (3) More efficiently scanning the resulting $(S, \mathcal{A}_\triangle)$.

3 General Results via ε-Coverings

For general range spaces of contant VC-dimension ν we can directly apply the work of Matheny *et al.* [19] to get a bound. A random sample N of size $O(\frac{\nu \log \nu}{\varepsilon} \log \frac{\nu}{\varepsilon})$ induces an ε-covering $(X, \mathcal{A}_{|N})$ with constant probability by (P1). A random sample S of size $O(\frac{\nu}{\varepsilon^2})$ induces an ε-sample with constant probability. By (P2), scanning the ranges in $(X, \mathcal{A}_{|N})$, evaluating $\Phi(A)$ on each ranges A using S, and returning the maximum $\hat{A}_\varepsilon$ induces the ε-approximation of $\Phi(A^*)$ as we desire. Including the time to calculate N and S we obtain the following result.

▶ **Theorem 1.** *Consider a range space $(X, \mathcal{A})$ with constant VC-dimension ν, with $|X| = m$, and conforming map $\psi_\mathcal{A}$. For $A^* = \arg\max_{A \in \mathcal{A}} \Phi(A)$, with probability at least $1 - \delta$, in time $O(m + \frac{1}{\varepsilon^{\nu+2}} \log^\nu \frac{1}{\varepsilon} \log \frac{1}{\delta})$, we can find a range $\hat{A}_\varepsilon$ so that $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$.*

Proof. First compute random samples N and S of size $O(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$ and $O(\frac{1}{\varepsilon^2})$ respectively. The algorithm naively considers all $O((\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})^\nu)$ subsets $B \subset N$ of size ν, and calculates the quantity $\Phi(S \cap \psi_\mathcal{A}(B))$. By (P2), this can be used to ε-approximate $\Phi(A)$ for any range $A \in \mathcal{A}$ which has less than ε-symmetric difference with $\psi_\mathcal{A}(B)$. Moreover, since $(X, \mathcal{A}_{|N})$ is an ε-cover, with constant probability any range A is within symmetric difference of at most εm of one induced by some subset B. Thus, with constant probability we observe some range $\hat{A}_\varepsilon = X \cap \psi_\mathcal{A}(B)$ for which $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$ (after adjusting constants in the size of N and S). To amplify the probability of success to $1 - \delta$, we repeat this process $O(\log \frac{1}{\delta})$ times, and return the $\hat{A}_\varepsilon$ with median score. ◀

3.1 Halfspaces

The above general result applied to halfspaces $(X, \mathcal{H}_d)$, would require $O(m + \frac{1}{\varepsilon^{d+2}} \log^d \frac{1}{\varepsilon} \log \frac{1}{\delta})$ time. We improve this runtime to $O(m + \frac{1}{\varepsilon^{d+1}} \log \frac{1}{\delta})$. First, a recent paper [18] shows that with constant probability an ε-sample S for $(X, \mathcal{H}_2)$ of size $s = O(\frac{1}{\varepsilon^{4/3}} \log^{2/3} \frac{1}{\varepsilon})$ can be constructed in $O(m + \frac{1}{\varepsilon^2} \log(\frac{1}{\varepsilon}))$ time. Second we create a weak ε-covering of $(X, \mathcal{H}_d)$ using $(X, \mathcal{H}_{d|N})$ for a random sample N. We show this only requires a random sample of size $O(\frac{d^2}{\varepsilon} \log d) = O(1/\varepsilon)$. Then, we show how to enumerate these ranges $(X, \mathcal{H}_{d|N})$ while maintaining the counts from S (an ε-sample of only $(X, \mathcal{H}_2)$) with less overhead than the previous brute force approaches. Ultimately this requires time $O(m + \frac{1}{\varepsilon^{d+1/3}} \log^{2/3} \frac{1}{\varepsilon})$, with constant probability. For space, the details are in the full version [17].

Moreover, this can be applied to disks $(X, \mathcal{D})$ in $O(m + \frac{1}{\varepsilon^{3+1/3}} \log^{2/3} \frac{1}{\varepsilon})$ time.

4 Rectangles

For the case of rectangles $(X, \mathcal{R}_d)$, we will describe two classes of algorithms. One simply creates an ε-cover $(X, \mathcal{R}_{d|N})$ and evaluates each rectangle A in this cover on an ε-sample S as before. The other takes specific advantage of the orthogonal structure of the rectangles and of "linearity" of Φ; this algorithm can find the maximum in Φ among ranges in $(X, \mathcal{R}_{d|N})$ without considering every possible range. Our techniques are inspired by several algorithms [4, 24, 8] for the exact maximization problem, but requires new ideas to efficiently take advantage of using both N and S. Common to all techniques will be an efficient way to compute an ε-cover based on a grid.

Grid ε-covers for rectangles. We create a grid G defined as the cross-product of $r = O(1/\varepsilon)$ cells along each axis. Straightforward details of its construction and use are in the full version [17]. We label the rectangular ranges of X restricted to this grid boundary as $(X, \mathcal{R}_{d|G})$; it is an ε-cover of $(X, \mathcal{R}_d)$. The main results of this ε-cover are in the next lemma and theorem.

▶ **Lemma 2.** *For range space $(X, \mathcal{R}_d)$ where $|X| = m$, the construction of grid G takes $O(m \log m + \frac{1}{\varepsilon^d})$ time, has $O(1/\varepsilon)$ cells on each side, and induces an ε-cover $(X, \mathcal{R}_{d|G})$ of $(X, \mathcal{R}_d)$ for constant $d > 1$.*

▶ **Theorem 3.** *Consider a range space $(X, \mathcal{R}_d)$ with $|X| = m$ and an Lipschitz-continuous function Φ with maximum range $A^* = \arg\max_{A \in \mathcal{R}_d} \Phi(A)$. With probability at least $1 - 1/e^{1/\varepsilon}$, in time $O(m + \frac{1}{\varepsilon^{2d}})$ we can find a range $\hat{A}_\varepsilon$ so that $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$.*

4.1 Algorithms for Decomposable Functions

Here we exploit a critical "linear" property of Φ that a rectangle A can be decomposed into any two parts A_1 and A_2 and $\Phi(A) = \Phi(A_1) + \Phi(A_2)$. Technically, we solve both $\Phi^+(A) = \mu_R(A) - \mu_B(A)$ and $\Phi^-(A) = \mu_B(A) - \mu_R(A)$ separately, and take their max. In particular, this allows us (following exact algorithms [4]) to decompose the problem along a separating line. The solution then either lies completely on one half, or spans the line. In the exact case on s points, this ultimately leads to a run time recurrence of $\mathcal{T}_1(s) = 2\mathcal{T}_1(s/2) + \mathcal{T}_2(s)$ where $\mathcal{T}_2(s)$ is the time to compute the problem spanning the line. The line spanning problem can then be handled using a different recurrence that leads to $\mathcal{T}_2(s) = O(s^2)$ and a total runtime for the problem of $\mathcal{T}_1(s) = 2\mathcal{T}_1(s/2) + O(s^2) = O(s^2)$ [4].

First we show we can efficiently construct a special sample S of size $s = O(1/(\varepsilon^2 \log \frac{1}{\varepsilon}))$, but this still would requires runtime of roughly $1/\varepsilon^4$.

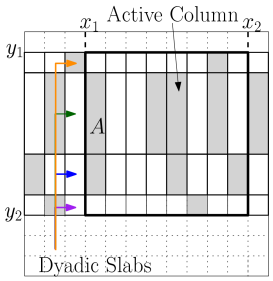
Our approximate algorithm will significantly improve upon this be compressing the representation at various points, but requiring some extra bookkeeping and a bit more complicated recurrence to analyze. In short, we can map S to an $r \times r$ grid (using Lemma 2), and then the recurrence only depends on the dyadic y-intervals of the grid. We can compress each such interval to have only $\varepsilon s/\log r$ error, since each query only touches about $\log r$ of these intervals. The challenge then falls to maintaining this compressed structure more efficiently during the recurrence.

The dense exact case on an $r \times r$ grid is also well studied. There exists a practically efficient $O(r^3)$ time method [5] based on Kadane's algorithm (which performs best as gridScan_linear; see Section 6), and a more complicated method taking $O(r^3(\frac{\log\log r}{\log r})^{\frac{1}{2}})$ time [24]. By allowing an approximation, we ultimately reduce this runtime to $O(r^2 \log r) = O(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})$.

We will focus on the 2d case. This is where the advantage over the Theorem 3 bound of $O(m+1/\varepsilon^4)$ is most notable. Generalization to high dimensions is straightforward: enumerate over pairs of grid cells to define the first $d-2$ dimensions, then apply the 2-dimensional result on the remaining dimensions.

Tree and slab approximation. The algorithm builds a binary tree over the rows (the y values) of G. We will assume that the number of cells in each axis $r = O(1/\varepsilon)$ is a power of 2 (otherwise we can round up), so it is a perfectly balanced binary tree.

At the ith level of the tree, each node contains $r/2^i$ rows and there are 2^i nodes. We refer to the family of rows represented by a subtree as a *slab*. Any grid-aligned rectangle $A = [x_1, x_2] \times [y_1, y_2]$ can be defined as the intersection of $[x_1, x_2]$ with at most $2\log_2 r$ slabs in the y-coordinate – the classic dyadic decomposition. This implies we can tolerate $\eta s = O(\varepsilon s/\log r)$ additive error in each slab to have at most $O(\varepsilon s)$ additive error overall (which implies the percentage of red and of blue points in each range has additive $O(\varepsilon)$ error).



Since the rectangle will span the entire vertical extent (y direction) of each slab in this decomposition, the additive error of a slab can be obtained along just the horizontal (x) direction. Thus, we can scan cells from left to right within a slab, and only retain the cumulative weight in a cell when it exceeds ηs. We refer to this operation as *η-compression*. We denote each column (and x value) within a slab where it has retained a non-zero value as *active*, all other columns are *inactive*. We store the active cells in a linked list.

Since there are $\Theta(s/r)$ points per row, it implies we can approximate each slab consisting of 1 row (a leaf of the tree, level $\log_2 r$) with weights in only $O(1/(r\eta)) = O(\log r)$ cells (since $r = O(\frac{1}{\varepsilon})$). And a slab at level i (originally with $\Theta(s/2^i)$ points) can be approximated by accumulating weight in $O(\min\{r, 1/(\eta 2^i)\})$ cells. For level $i > \log 1/\eta r$, this compresses the points in that slab.

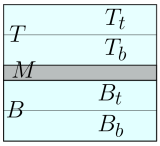
▶ **Lemma 4.** *In $O(r^2)$ time, we can compress all slabs in the tree, so a slab at level i contains $\ell_i = O(\min\{r, 1/(\eta 2^i)\})$ active columns where $\eta = O(\varepsilon/\log r)$.*

Interval Preprocessing and Merging. Now consider a subproblem, where we seek to find a rectangle $A = [x_1, x_2] \times [y_1, y_2]$ to maximize the total weight, restricted to a given horizontal extent $[y_1, y_2]$ (e.g., within a slab). We reduce this to a 1d problem by summing the weights for each x-coordinate to $w_x = \sum_{y \in [y_1, y_2]} w_{x,y}$. Then there is an often-used [4, 7, 2] way to preprocess intervals $[x'_1, x'_2]$ so they can be merged and updated. It maintains 3 maximal weight subintervals: (1) the maximal weight subinterval in $[x'_1, x'_2]$, (2) the maximal weight interval including the left boundary x'_1, and (3) the maximal weight interval including the right boundary x'_2. Given two preprocessed adjacent intervals $[x'_1, x'_2]$ and $[x'_2 + 1, x'_3]$, we can update these subintervals to $[x'_1, x'_3]$ in $O(1)$ time [4]. Thus given a horizontal extent with a active intervals, we can find the maximum weight subinterval in $O(a)$ time.

Recursive construction. Now we can describe our recursive algorithm for finding the maximal weight rectangle on the grid G. We find the maximum weight rectangle through 3 options: (1) completely in the top child's subtree, (2) completely in the bottom child's subtree, (3) overlapping both the top and bottom child's subtree. The total time can be written as a recurrence as $\mathcal{T}_1(r) = 2\mathcal{T}_1(r/2) + \mathcal{T}_2(r)$, where $\mathcal{T}_2$ is the time to solve case (3).

Case (3) requires another recurrence to understand, and it closely follows the "strip-constrained" algorithm of Barbay *et al.* [4]; our version will account for the dense grid.

We consider the Strip-constrained grid search problem: *First fix a* strip M *which is a consecutive set of rows. Then consider two slabs* T *and* B *where* T *is directly above (on* ***t****op of)* M *and* B *is directly* ***b****elow* M*. A column of* M *is active if it is active in* T *or* B*. Counts in active columns of* M *are maintained, and intervals of* M *described by consecutive inactive columns have been merged. The goal is to find the maximum weight rectangle with vertical span* $[y_1, y_2]$ *where* y_2 *is in* T *and* y_1 *is in* B *(it must cross* M*).*



We specifically want to solve this problem when M is empty, T is the top child and B the bottom child of the root, and all columns are initially active. We call this the case of size r since there are still r rows.

▶ **Lemma 5.** *The Strip-constrained grid search problem of size* r *over an* η*-compressed binary tree takes* $O(r/\eta)$ *time.*

Proof. Following Barbay *et al.* [4] we split the problem into 4 subcases, following the subtrees of the slabs. Slab T has a top T_t and bottom T_b sub-slab, and similarly B_t and B_b for B. Then we consider 4 recursive cases with new strip M': (1) slabs T_t and B_b with $M' = T_b \cup M \cup B_t$, (2) slabs T_b and B_b with $M' = M \cup B_t$, (3) slabs T_t and B_t with $M' = T_b \cup M$, and (4) slabs T_b and B_t with $M' = M$. The cost in a recursive step is the preprocessing of the new slab M'. We will describe the largest case (1); the others are similar.

Strip M already maintains preprocessed intervals of inactive columns. When T_b or B_t has an active column which is inactive in T_t and B_b, we treat this as a new inactive interval that needs to be maintained within M'. The weights from T_b and B_t are added to that in the column for M. If inactive intervals of M' are then adjacent to each other, they are merged, in $O(1)$ time each. This completes the recursive step for case (1).

In the base case when slabs T and B are single rows (at depth $O(\log r)$), the range maximum is restricted to use their active columns. We sum weights on active columns

in T, B, and M. Then also considering the inactive intervals on M, invoke the interval merging procedure [4] to find the maximal range, in time proportional to the number of active intervals, in $O(1/(2^{\log r}\eta) = O(1/(r\eta))$ time.

The cost of recursing in any case is also proportional to the number of active columns since this bounds the number of potential merges, and the time it takes to scan the linked lists of active columns to detect where the merging is needed. At level i this is bounded by $\ell_i = \min\{r, 1/(\eta 2^i)\} \leq O(1/(\eta 2^i))$.

At each level i there are 4^i recursive sub instances and at most $O(1/(2^i\eta))$ active columns, and therefore merging takes $Z_i = 4^i O(1/(2^i\eta)) = 2^i O(1/\eta)$ time. The cost is asymptotically dominated by the last level, which takes time $2^{\log_2 r} O(1/\eta) = O(r/\eta)$. ◀

Letting $\eta = \varepsilon/(\log r) = O(1/(r \log r))$ (since $r = O(1/\varepsilon)$) as it is in Lemma 4 we have a bound of $\mathcal{T}_2(r) = O(r^2 \log r)$. We can solve the first recurrence of $\mathcal{T}_1(r) = 2\mathcal{T}_1(r/2) + \mathcal{T}_2(r) = 2\mathcal{T}_1(r/2) + O(r^2 \log r) = O(r^2 \log r)$. Using $r = O(1/\varepsilon)$ this bounds the overall runtime of finding $\max_{R \in (S, \mathcal{R}_{d|G})} \Phi(R)$ as $O(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon})$.

▶ **Theorem 6.** *Consider $(X, \mathcal{R}_2)$ with $|X| = m$ and $A^* = \arg\max_{A \in \mathcal{R}_2} \Phi(A)$. With probability at least $1-\delta$, in time $O(m + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \log \frac{1}{\delta})$, we can find a range $\hat{A}_\varepsilon$ so $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$.*

In the full version [17], we reduce this time to $O(m + \frac{1}{\varepsilon^2} \log\log \frac{1}{\varepsilon} \log \frac{1}{\delta})$.
For $(X, \mathcal{R}_d)$ and d constant, the runtime increases to $O(m + \frac{1}{\varepsilon^{2d-2}} + \frac{1}{\varepsilon^2} \log\log \frac{1}{\varepsilon} \log \frac{1}{\delta})$.

Conditional lower bound. Backurs *et al.* [3] recently showed $\Omega(m^2)$ time is required to solve for $A^* = \arg\max_{A \in (X, \mathcal{R}_2)} \Phi(A)$, assuming that all pairs shortest path (APSP) requires cubic time. We can show this implies that our algorithm is nearly tight. If we set $\varepsilon = 1/4m$ then if any algorithm could find an $\hat{A}_\varepsilon$ such that $\Phi(\hat{A}_\varepsilon) \geq \Phi(A^*) - \varepsilon$, then it would imply that $|\mu_R(A^*) - \mu_B(A^*)| - |\mu_R(\hat{A}) - \mu_B(\hat{A})| \leq \varepsilon$. And hence the difference in counts of points in each pair μ_R and μ_B is off by at most $2\varepsilon m = 2(1/4m)m = 1/2$. Thus it must be the optimal solution. If this can run in $o(m + 1/\varepsilon^2)$ time, it implies an $o(m^2)$ algorithm, which implies a subcubic algorithm for APSP, which is believed impossible.

▶ **Theorem 7.** *For $(X, \mathcal{R}_2)$ with $|X| = m$, and $A^* = \arg\max_{A \in \mathcal{R}_2} \Phi(A)$. It takes $\Omega(m + \frac{1}{\varepsilon^2})$ time to find a range $\hat{A}_\varepsilon$ so that $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$, assuming APSP takes $\Omega(n^3)$ time.*

5 Statistical Discrepancy Function Approximation

In this section we address approximating $\max_{A \in (X, \mathcal{A})} \Phi(A)$ when it is a more general function of $\mu_R(A)$, and $\mu_B(A)$. Rewrite $\Phi(A) = \phi(\mu_R(A), \mu_B(A))$, and in this section it will be more convenient to discuss $\phi(r, b)$ where $r = \mu_R(A)$ and $b = \mu_B(A)$.

We say ϕ is (τ, γ)-*linear* if it can be represented with up to ε-error as the upper envelope of γ functions of slope at most τ. We can then simply maximize each function individually, and return the maximum overall score. When γ and τ are constant (as with $\phi(r, b) = |r - b|$), we simply say the function is *linear*.

First observe that Theorem 1, algorithms in Section 3.1 (see full version [17]), and Theorem 3 simply evaluate $\Phi(A)$, so if this can be done in constant time, and the slope τ is constant, then these results automatically hold. However, Theorem 6 requires the linearity property.

For the spatial scan statistic application, the most common function [12] is defined $\phi_K(r, b) = r \ln \frac{r}{b} + (1 - r) \ln \frac{1-r}{1-b}$, and is non-linear. We define a more general class of *statistical discrepancy functions* (SDF), which includes ϕ_K. Such ϕ have domain $r, b \in [0, 1]$,

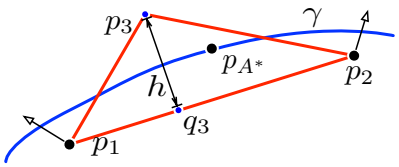


Figure 3 For Lemma 8.

$\phi(r,b) = 0$ when $r = b$ and this is its minimum, and $\phi(r,b)$ is convex on $(0,1)^2$. Moreover, for these functions, it suffices too consider a range $[\xi, 1-\xi]^2$ for small constant ξ (c.f. [2, 1, 19]), and that in this range ϕ is τ-Lipschitz where τ is a constant depending only ξ.

Agarwal *et al.* [2] approximated such functions by considering $O(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ linear functions, each tangent to ϕ, so their upper envelope $\tilde{\phi}$ satisfied $\max_{(r,b)\in[\xi,1-\xi]^2} |\phi(r,b) - \tilde{\phi}(r,b)| \leq \varepsilon$.

We will construct an approximation of ϕ with linear functions with a very different approach. Unlike the previous approach which only considers the function ϕ, our approach adapts the set of linear functions to the function ϕ *and data* $(X, \mathcal{A})$. It uses $O(1/\sqrt{\varepsilon})$ linear functions.

Function approximation. Consider the distinct ranges in $(X, \mathcal{A})$; each range A corresponds to a point $p_A = (\mu_R(A), \mu_B(A))$. Let $P = \{p_A \mid A \in (X, \mathcal{A})\}$ be this set of points. Then p_{A^*}, must lie on $\mathsf{CH}(P)$, the convex hull of P, where $A^* = \arg\max_{A\in(X,\mathcal{A})} \Phi(A)$.

Moreover, each point p on $\mathsf{CH}(P)$ maximizes some linear function, $f(r,b) = \alpha r + \beta b$. If $p = \arg\max_{p'\in P} f(r_p, b_p)$, then it also maximizes $f_c(r,b) = (\alpha/c)r + (\beta/c)b$ for any $c > 0$. We can therefore restrict our attention (by implicit choice of c) to only functions with $\alpha^2 + \beta^2 = 1$. These functions correspond to a dot product $\langle(\alpha,\beta),(r,b)\rangle$ and are maximized by points on $\mathsf{CH}(P)$ where (α,β) is between two adjacent normals on the boundary of $\mathsf{CH}(P)$.

To further simplify, we now parameterize these functions by an angle $\theta = \arccos(-\alpha)$ (where still $\alpha^2 + \beta^2 = 1$). We focus on $\theta \in [0, \pi/2]$ as we can always repeat the procedure on the other 3 quadrants.

Now let f^*_θ be any linear function such that $p_{A^*} = \arg\max_{p\in P} f^*_\theta(p)$ is maximized by the point p_{A^*} corresponding to the optimal range A^*.

▶ **Lemma 8.** *Consider* $p_1 = \arg\max_{p\in P} f_{\theta_1}(p)$ *and* $p_2 = \arg\max_{p\in P} f_{\theta_2}(p)$ *so that* $p_{A^*} = \arg\max_{p\in P} f^*_\theta(p)$ *and* $\theta_1 \leq \theta \leq \theta_2$. *Then* $\phi(p_{A^*}) \leq \max\{\phi(p_i), \phi(p_j)\} + \tau \cdot \frac{\|p_1 - p_2\|}{2} \tan(\frac{\theta_2 - \theta_1}{2})$.

Proof. Define a triangle through points p_1, p_2, and a point p_3. The point p_3 is defined at the intersections of the normals to f_{θ_1} at p_1 and to f_{θ_2} at p_2. We refer to "above" in the normal direction of the edge between p_1 and p_2, and in the direction of p_3.

First we show that p_{A^*} must be inside the triangle. If it is above the edge connecting p_1 and p_3, then it would be $\arg\max_{p\in P} f_{\theta_1}(p)$. Similarly it cannot be above the edge connecting p_2 and p_3. Also, it must be above the edge connecting p_1 and p_2, since otherwise by convexity $\max(\phi(p_1), \phi(p_2)) > \phi(p_{A^*})$ and one of p_1 or p_2 would maximize f^*_θ.

We say the height of the triangle h is defined as the distance from p_3 to q_3, where q_3 is the closest point on the edge through p_1 and p_2.

Let $\angle_1$ be the internal triangle angle at p_1, and $\angle_2$ at p_2. Then $(\theta_2 - \theta_1) = \angle_1 + \angle_2$. Now $h = ||p_1 - q_3|| \tan(\angle_1) = ||p_2 - q_3|| \tan(\angle_2)$ which, fixing $\|p_1 - p_2\|$, is maximized when $\angle_1 = \angle_2 = \frac{(\theta_2-\theta_1)}{2}$. Summing $h \leq ||p_1 - q_3|| \tan((\theta_2-\theta_1)/2)$ and $h \leq ||p_2 - q_3|| \tan((\theta_2-\theta_1)/2)$ it can be seen that $h \leq \frac{1}{2}(||p_1-q_3|| + ||p_2-q_3||) \tan((\theta_2-\theta_1)/2) = \frac{1}{2}(||p_1-p_2||) \tan((\theta_2-\theta_1)/2)$. Finally, we argue that $\min\{\phi(p_{A^*}) - \phi(p_1), \phi(p_{A^*}) - \phi(p_2)\} \leq \tau \cdot h$. Let γ be the iso-curve

of ϕ at value $\phi(p_{A^*})$. It must pass above p_1 and p_2, otherwise they would be the maximum. It also must pass within a distance of h from either p_1 or p_2 since γ is convex, it contains p_{A^*}, and p_{A^*} is within h of the edge between p_1 and p_2. Then the lemma follows since ϕ is τ-Lipschitz. ◀

To choose a set of linear functions we start with two linear functions f_0 and $f_{\pi/2}$, whose maximum in P are points p_1 and p'_1. These induce a triangle as in the proof of Lemma 8, and p_{A^*} must be in this triangle. If its height $h = \frac{\|p_1 - p'_1\|}{2} \tan(\frac{\pi}{4}) > \varepsilon/\tau$, then we choose a new function $f_{\pi/4}$ (at the midpoint of the two angles) whose maximum is point p_2. Now recurse on triangles defined by p_1 and p_2, and by p_2 and p'_1.

▶ **Lemma 9.** *The recursive algorithm considers at most $\sqrt{\tau/\varepsilon}$ functions to maximize.*

Proof. Index the points found by the algorithm $\{p_1, p_2, \ldots, p_{k+1}\}$ in the order they appear on the convex hull. Each consecutive pair p_i and p_{i+1} defines a triangle with height at most ε/τ. Let $\ell_i = \|p_i - p_{i+1}\|$ and $\gamma_i = \theta_{i+1} - \theta_i$ where the p_i and p_{i+1} where chosen by maximizing functions f_{θ_i} and $f_{\theta_{i+1}}$, respectively. It follows that $\sum_{i=1}^k \ell_i \leq 2$ and $\sum_{i=1}^k \gamma_i = \pi/2$. We also have for each triangle that $\frac{\varepsilon}{\tau} \leq \frac{\ell_i}{2} \tan(\frac{\gamma_i}{2}) \leq \frac{\ell_i}{2} \cdot \frac{2\gamma_i}{\pi}$. Thus for each term we have $\ell_i \geq \frac{\varepsilon\pi}{\tau} \frac{1}{\gamma_i}$, and summing over k terms $\sum_{i=1}^k \frac{\varepsilon\pi}{\tau} \frac{1}{\gamma_i} \leq \sum_{i=1}^k \ell_i \leq 2$. Now in the inequality $\frac{2\tau}{\varepsilon\pi} \geq \sum_{i=1}^k \frac{1}{\gamma_i}$ such that $\sum_{i=1}^k \gamma_i = \pi/2$, then k is the largest when all of the γ_i have the same value $\gamma_i = \frac{\pi}{2k}$. In this case, then $\frac{2\tau}{\varepsilon\pi} \geq \sum_{i=1}^k \frac{1}{\gamma_i} = \sum_{i=1}^k \frac{2k}{\pi} = k^2 \frac{2}{\pi}$. Solving for k reveals $k \leq \sqrt{\varepsilon/\tau}$. ◀

Now we analyze the full algorithm for maximizing a statistical discrepancy function over $(X, \mathcal{R}_d)$ with τ and d as constants. We first invoke Lemma 2 to construct the grid in $O(m + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \log \frac{1}{\delta} + \frac{1}{\varepsilon^d})$ time. We then use Theorem 6 in $F = O(\frac{1}{\varepsilon^{2d-2}} \log \frac{1}{\varepsilon})$ time to find the approximate maximum range for any linear function Φ'.

Then we run the above recursive triangle algorithm repeatedly on the constructed grid, and each function maximization takes F time. By Lemma 9 we need to make $O(\sqrt{1/\varepsilon})$ calls. And by Lemma 8 one of the function calls must find an approximately correct answer.

▶ **Theorem 10.** *Consider a range space $(X, \mathcal{R}_d)$ with $|X| = m$ and d constant. For a statistical discrepancy function Φ with τ constant and with maximum range $A^* = \arg\max_{A \in \mathcal{R}_d} \Phi(A)$, then with probability at least $1 - \delta$, in time $O(m + \frac{1}{\varepsilon^{2d-1.5}} \log \frac{1}{\varepsilon} + \frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon} \log \frac{1}{\delta})$, we can find a range $\hat{A}_\varepsilon$ so that $|\Phi(A^*) - \Phi(\hat{A}_\varepsilon)| \leq \varepsilon$.*

6 Experiments on Rectangles

We implemented 5 rectangle scanning algorithms. For baselines, we consider (1) Scanning all rectangles without sampling (based on common software for disks [13]) (SatScan (no sampling)), (2) Scanning all rectangles on one random sample [1] (SatScan), and (3) Scanning all rectangles on two random samples N and S [19] (netScan). Then we compare our algorithms which first round to a grid then (4) Efficiently enumerate the grid rectangles (gridScan, Theorem 3), or (5) Evaluate the maximum grid rectangle in $O(r^3)$ time [5] for a linear ϕ (gridScan_linear, Section 4.1) and using the linearization for non-linear ϕ (Section 5). This is the core operation within spatial scan statistics; it is typically run 1000 times to detect a region *and* determine significance [12], therefore scalability of this operation is paramount. Solutions with approximate ϕ within ε-error retain high statistical power [19], so it will be useful to directly compare the runtime performance of these algorithms which allow approximation.

Table 2 Runtimes on 1000 points with 1% error, over 20 trials; roughly $n = 19$ and $s = 350$.

	SatScan (no sampling)	SatScan	netScan	gridScan	gridScan_linear
Time (sec)	5287	7.44	.0279	.0194	.0082

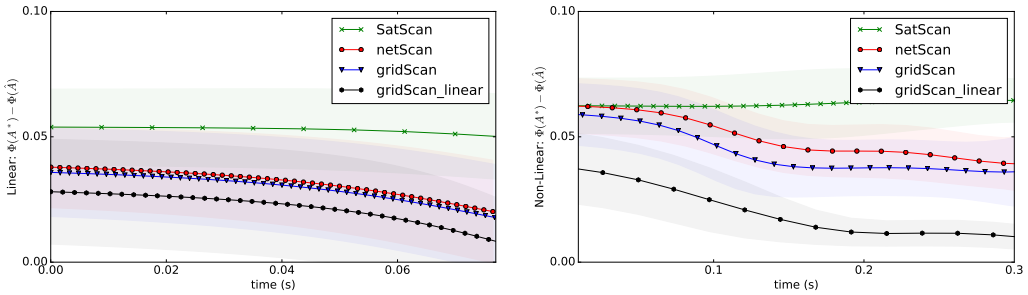


Figure 4 Trend of time versus error for on linear (left) and non-linear (right) functions.

First, fixing a tolerable error at 1% of $\phi(A^*)$, we run each algorithm on $m = 1000$ points, for a planted range with 5% of the data, and use ϕ as the Kuldorff scan statistic [12]. The results are in Table 2. All sampling methods drastically improve over the brute force approach, and using two-level sampling significantly improves over one random sample. Our method (gridScan_linear) improves over the previous best (netScan) by a factor of about 3.5.

We also compare the time-accuracy trade-off for sampling-based algorithms on $m = 1$ *million* points. SatScan without sampling is not tractable at this scale, so is not compared. We again plant a random rectangle A overlapping 1% of the data. Within A, points are made red (measured value 1) at rate 0.08, and outside at rate 0.01. The runtime includes the time to construct the grid, but not time to generate the initial sample – common to all algorithms. We calculate $\Phi(A^*) - \Phi(\hat{A})$ for the planted A^* and found $\hat{A}$ regions, using a linear $\phi(m, b) = \frac{1}{\sqrt{2}}(m - b)$ function and the non-linear Kuldorff [12] ϕ function. Figure 4 shows a kernel regression trend line (with 1 std-dev error bars) for 300 trials with various n, s values, always maintaining $n \approx \sqrt{s}$ as suggested the samping theorems. Again gridScan_linear is much faster than gridScan, which is slightly faster than netScan, which is significantly faster than SatScan. The improvement is more pronounced in the non-linear setting where ϕ is steeper; this is perhaps surprisingly even true for gridScan_linear which has an extra $\sqrt{1/\varepsilon}$-factor in runtime in that case due to the multiple linear functions considered.

Ultimately, these plots show that *discrete geometric approaches providing asymptotically efficient algorithms also give significant empirical improvements*, even compared to the ubiquitous and simple random sampling approaches.

References

1 Deepak Agarwal, Andrew McGregor, Jeff M. Phillips, Suresh Venkatasubramanian, and Zhengyuan Zhu. Spatial Scan Statistics: Approximations and Performance Study. In *KDD*, 2006.

2 Deepak Agarwal, Jeff M. Phillips, and Suresh Venkatasubramanian. The Hunting of the Bump: On Maximizing Statistical Discrepancy. *SODA*, 2006.

3 Arturs Backurs, Nishanth Dikkala, and Christos Tzamos. Tight Hardness Results for Maximum Weight Rectangles. In *ICALP*, 2016. `arXiv:1602.05837`.

4 Jérémy Barbay, Timothy M. Chan, Gonzalo Navarro, and Pablo Pérez-Lantero. Maximum-weight planar boxes in time (and better). *Information Processing Letters*, 114(8):437–445, 2014.

5 Jon Bentley. Programming Pearls – Perspective on Performance. *Communications of ACM*, 27:1087–1092, 1984.

6 Bernard Chazelle. *The Discrepancy Method*. Cambridge, 2000.

7 David Dobkin and David Eppstein. Computing the Discrepancy. In *Proceedings 9th Annual Symposium on Computational Geometry*, 1993.

8 David P. Dobkin, David Eppstein, and Don P. Mitchell. Computing the Discrepancy with Applications to Supersampling Patterns. *ACM Trans. Graph.*, 15(4):354–376, October 1996.

9 Takeshi Fukuda, Yasukiko Morimoto, Shinichi Morishita, and Takeshi Tokuyama. Data Mining Using Two-dimensional Optimized Association Rules: Scheme, Algorithms, and Visualization. *SIGMOD Rec.*, 25(2):13–23, June 1996.

10 David Haussler. Sphere Packing Numbers for Subsets of the Boolean n-Cube with Bounded Vapnik-Chervonenkis Dimension. *J. Combinatorial Theory, A*, 69:217–232, 1995.

11 Lan Huang, Martin Kulldorff, and David Gregorio. A Spatial Scan Statistic for Survival Data. *BioMetrics*, 63:109–118, 2007.

12 Martin Kulldorff. A Spatial Scan Statistic. *Communications in Statistics: Theory and Methods*, 26:1481–1496, 1997.

13 Martin Kulldorff. *SatScan User Guide*, 7.0 edition, 2006. URL: `http://www.satscan.org/`.

14 Martin Kulldorff, Lan Huang, Linda Pickle, and Luiz Duczmal. An elliptic spatial scan statistic. *Statistics in medicine*, 25 22:3929–43, 2006.

15 Yi Li, Philip M. Long, and Aravind Srinivasan. Improved Bounds on the Samples Complexity of Learning. *J. Comp. and Sys. Sci.*, 62:516–527, 2001.

16 Ming C Lin and Dinesh Manocha. *Applied Computational Geometry. Towards Geometric Engineering: Selected Papers*, volume 114. Springer Science & Business Media, 1996.

17 Michael Matheny and Jeff M. Phillips. Computing Approximate Statistical Discrepancy. Technical report, arXiv, 2018. `arXiv:1804.11287`.

18 Michael Matheny and Jeff M. Phillips. Practical Low-Dimensional Halfspace Range Space Sampling. In *European Symposium on Algorithms*, 2018. `arXiv:1804.11307`.

19 Michael Matheny, Raghvendra Singh, Liang Zhang, Kaiqiang Wang, and Jeff M. Phillips. Scalable Spatial Scan Statistics Through Sampling. In *SIGSPATIAL*, 2016.

20 Jiri Matousek. *Geometric Discrepancy*. Springer, 1999.

21 Jiri Matousek. *Lectures in Discrete Geometry*. Springer, 2002.

22 Daniel B. Neill and Andrew W. Moore. Rapid Detection of Significant Spatial Clusters. In *KDD*, 2004.

23 Norbert Sauer. On the Density of Families of Sets. *Journal of Combinatorial Theory, Series A*, 13:145–147, 1972.

24 Tadao Takaoka. Efficient Algorithms for the Maximum Subarray Problem by Distance Matrix Multiplication. *CATS*, 2002.

25 Toshiro Tango and Kunihiko Takahashi. A flexibly shaped spatial scan statistic for detecting clusters. *International Journal of Health Geographics*, 4(1):11, May 2005.

26 Vladimir Vapnik and Alexey Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to their Probabilities. *Theo. of Prob and App*, 16:264–280, 1971.

27 Guenther Walther. Optimal and fast detection of spatial clusters with scan statistics. *Ann. Statist.*, 38(2):1010–1033, April 2010.

28 Mingxi Wu, Xiuyao Song, Chris Jermaine, Sanjay Ranka, and John Gums. A LRT Framework for Fast Spatial Anomaly Detection. In *KDD*, 2009.