Nearest Neighbor Searching Under Uncertainty II

Pankaj K. Agarwal
Duke University

Boris Aronov
Polytechnic Institute of NYU

Sariel Har-Peled
University of Illinois

Jeff M. Phillips
University of Utah

Ke Yi
HKUST

Wuzhou Zhang
Duke University

ABSTRACT

Nearest-neighbor (NN) search, which returns the nearest neighbor of a query point in a set of points, is an important and widely studied problem in many fields, and it has wide range of applications. In many of them, such as sensor databases, location-based services, face recognition, and mobile data, the location of data is imprecise. We therefore study nearest neighbor queries in a probabilistic framework in which the location of each input point is specified as a probability distribution function. We present efficient algorithms for (i) computing all points that are nearest neighbors of a query point with nonzero probability; (ii) estimating, within a specified additive error, the probability of a point being the nearest neighbor of a query point; (iii) using it to return the point that maximizes the probability being the nearest neighbor, or all the points with probabilities greater than some threshold to be the NN. We also present some experimental results to demonstrate the effectiveness of our approach.

Categories and Subject Descriptors

F.2 [**Analysis of algorithms and problem complexity**]: Nonnumerical algorithms and problems; H.3.1 [**Information storage and retrieval**]: Content analysis and indexing—*indexing methods*

General Terms

Algorithms, Theory

Keywords

Indexing uncertain data, probabilistic nearest neighbor, approximate nearest neighbor, threshold queries

1. INTRODUCTION

Nearest-neighbor search is a fundamental problem in data management. It has applications in such diverse areas as spatial databases, information retrieval, data mining, pattern recognition, etc. In its simplest form, it asks for preprocessing a set S of n points in $\mathbb{R}^d$ into an index so that given any query point q, the nearest neighbor (NN) of q in S can be reported efficiently. This problem has been studied extensively in both the database and the computational geometry community, and is now relatively well understood. However, in some of the applications mentioned above, data is imprecise and is often modeled as probabilistic distributions. This has led to a flurry of research activities on query processing over probabilistic data, including the NN problem; see [7, 16] for surveys on uncertain data, and see, e.g., [15, 25] for application scenarios of NN search under uncertainty.

However, despite many efforts devoted to the probabilistic NN problem, it still lacks a theoretical foundation. Specifically, not only are we yet to understand its complexity (is the problem inherently more difficult than on precise data?), but we also lack efficient algorithms to solve it. Furthermore, existing solutions all use heuristics without nontrivial performance guarantees. This paper addresses some of these issues.

1.1 Problem definition

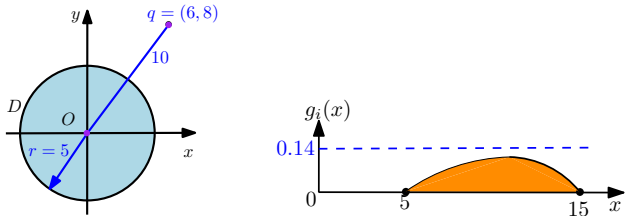
An *uncertain* point[1] P in $\mathbb{R}^2$ is represented as a continuous probability distribution defined by a probability density function (pdf) $f_P : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$; f_P may be a parametric pdf such as a uniform distribution or a Gaussian distribution, or may be a non-parametric pdf such as a histogram. The *uncertainty region* of P (or the *support* of f_P) is the set of points for which f_P is positive, i.e., $\text{Sup}\, f_P = \{x \in \mathbb{R}^2 \mid f_P(x) > 0\}$. We assume P has a bounded uncertainty region: if f_P is Gaussian, we work on truncated Gaussian, as in [10, 12]. We also consider the case where P is represented as a discrete distribution defined by a finite set $P = \{p_1, \ldots, p_k\} \subset \mathbb{R}^2$ along with a set of probabilities $\{w_1, \ldots, w_k\} \subset [0, 1]$, where $w_i = \Pr[P \text{ is } p_i]$ and $\sum_{i=1}^{k} w_i = 1$.

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$, and let $d(\cdot, \cdot)$ be the Euclidean distance. For a point $q \in \mathbb{R}^2$, let $\pi_i(q) = \pi(P_i, q)$ be the probability of $P_i \in \mathcal{P}$ being the nearest neighbor of q, referred to as its *qualification probability*, defined as follows:

For a point q, and $i = 1, \ldots, n$, let $g_{q,i}$ be the pdf of the distance between q and P_i. That is, $g_{q,i}(x) = \Pr[x \leq d(q, P_i) \leq x + \mathrm{d}x]$. See Fig. 1 for an example of $g_{q,i}$. Let $G_{q,i}(x) = \int_0^x g_{q,i}(y)\mathrm{d}y$ denote the cumulative distribution function (cdf) of the distance between q and P_i. Then $\pi_i(q)$,

[1] If the location of data is precise, we call it *certain*.

PODS'13, June 22–27, 2013, New York, New York, USA.



(a) (b)

Figure 1. (a) P_i is represented by a uniform distribution defined on a disk D of radius $r = 5$ and centered at origin O, $q = (6, 8)$, and $d(\cdot, \cdot)$ is L_2 metric; (b) $g_{q,i}(x)$.

the probability that P_i is the NN of q, is

$$\pi_i(q) = \int_0^\infty g_{q,i}(r) \prod_{j \neq i} (1 - G_{q,j}(r)) \mathrm{d}r. \tag{1}$$

Given a set $\mathcal{P}$ of n uncertain points, the *probabilistic nearest neighbor* (PNN) problem is to preprocess $\mathcal{P}$ into an index so that, for any given query point q, we can efficiently return all pairs $(P_i, \pi_i(q))$ such that $\pi_i(q) > 0$.

In addition, one can consider the *most likely NN* of q, denoted NN_L, which is the P_i with the maximum $\pi_i(q)$; or the *threshold NN*, i.e., all the P_i's with $\pi_i(q)$ exceeding a given threshold τ.

Usually, the PNN problem is divided into the following two subproblems, which are often considered separately.

Nonzero NNs. The first subproblem is to find all the P_i's with $\pi_i(q) > 0$ without computing the actual qualification probabilities, i.e., to find

$$\mathsf{NN}_{\neq 0}(q, \mathcal{P}) = \{P_i \mid \pi_i(q) > 0\}.$$

If the point set $\mathcal{P}$ is obvious from the context, we drop the argument $\mathcal{P}$ from $\mathsf{NN}_{\neq 0}(q, \mathcal{P})$, and write it as $\mathsf{NN}_{\neq 0}(q)$. Note that $\mathsf{NN}_{\neq 0}(q)$ depends (besides q) only on the uncertainty regions of the uncertain points, but not on the actual pdf's.

A possible approach to compute nearest neighbors is to use Voronoi diagrams. For example, the standard Voronoi diagram of a set of points in $\mathbb{R}^2$ (without uncertainty) is the planar subdivision so that all points in the same face have the same nearest neighbor. In our case, we define the *nonzero Voronoi diagram*, denoted by $\mathcal{V}_{\neq 0}(\mathcal{P})$, to be the subdivision of $\mathbb{R}^2$ into maximal connected regions such that $\mathsf{NN}_{\neq 0}(q)$ is the same for all points q within each region. That is, for a subset $\mathcal{T} \subseteq \mathcal{P}$, let

$$\mathrm{cell}_{\neq 0}(\mathcal{T}) = \{q \in \mathbb{R}^2 \mid \mathsf{NN}_{\neq 0}(q) = \mathcal{T}\}. \tag{2}$$

Although there are 2^n subsets of $\mathcal{P}$, we will see below that only a small number of them have nonempty Voronoi cells. The planar subdivision $\mathcal{V}_{\neq 0}(\mathcal{P})$ is induced by all the nonempty $\mathrm{cell}_{\neq 0}(\mathcal{T})$'s for $\mathcal{T} \subseteq \mathcal{P}$. The (combinatorial) complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ is the total number of vertices, edges, and faces in $\mathcal{V}_{\neq 0}(\mathcal{P})$. In this paper, we study the worst-case complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ and how it can be efficiently constructed. The complexity of the Voronoi diagram is often regarded as a measure of the complexity of the corresponding nearest-neighbor problem. In addition, once we have $\mathcal{V}_{\neq 0}(\mathcal{P})$, a point-location structure can be built on top of it to support $\mathsf{NN}_{\neq 0}$ queries in logarithmic time.

Similarly, one can consider the *most likely Voronoi diagram* (partitioning the plane into regions having the same most likely NN) and the *threshold Voronoi diagram* (partitioning the plane into regions having the same set of points with qualification probabilities exceeding τ). However, these Voronoi diagrams tend to be more complex as they depend on the actual distributions of the uncertain points.

Computing qualification probabilities. The second subproblem is to compute the qualification probability $\pi_i(q)$ for a given q and P_i. Since exact values of these probabilities are often unstable — a far away point can affect these probabilities — and computing them requires a complex n-dimensional integration, which is often expensive, we resort to computing $\pi_i(q)$ approximately within a given error tolerance $0 < \varepsilon < 1$. More precisely, we aim at returning a value $\hat{\pi}_i(q)$ such that $|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon$.

Note that, having solved these two subproblems, we obtain immediate approximate solutions to the most likely NN and the threshold NN problems.

1.2 Previous work

Nonzero NNs. Sember and Evans [28] showed that the worst-case complexity of the nonzero Voronoi diagram (though they did not use this term explicitly) when the uncertainty regions of the uncertain points are disks is $O(n^4)$; they did not offer any lower bound. If one only considers those cells of $\mathcal{V}_{\neq 0}(P)$ in which $\mathsf{NN}_{\neq 0}(q)$ contains only one uncertain point P_i, they showed that the complexity of these cells is $O(n)$. Note that for such a cell, we always have $\pi_i(q) = 1$ for any q in the cell, so they are called the *guaranteed Voronoi diagram*. Probably unaware of the work by Sember and Evans [28], Cheng *et al.* [15] proved an exponential upper bound for the complexity of the nonzero Voronoi diagram, which they referred to as UV-diagram.

The nonzero Voronoi diagram is not the only way to find the nonzero NNs. Cheng *et al.* [14] designed a branch-and-prune solution based on the R-tree. Recently, Zhang *et al.* [32] proposed to combine the nonzero Voronoi diagram with R-tree-like bounding rectangles. These methods do not have any performance guarantees.

Computing qualification probabilities. Computing the qualification probabilities has attracted a lot of attention in the database community. Cheng *et al.* [14] used numerical integration, which is quite expensive. Cheng *et al.* [12] and Bernecker *et al.* [9] proposed some filter-refinement methods to give upper and lower bounds on the qualification probabilities. Kriegel *et al.* [23] took a random sample from the continuous distribution of each uncertain point to convert it to a discrete one, so that the integration becomes a sum, and they clustered each sample to further reduce the complexity of the query computation. These methods are best-effort based: they do not always give the ε-error that we aim at — how tight the bounds are depends on the data.

Other variants of the problem. The PNN problem we focus on in this paper is the most commonly studied version of the problem, but many variants and extensions have been considered.

The probabilistic model we use is often called the *locational model*, where the location of an uncertain point follows the given distribution. This is to be compared with the *existential model*, where each point has a precise location but it appears with a given probability.

Besides using the qualification probability, one can also consider the expected distance from a query point q to an uncertain point, and return the one minimizing the expected distance as the nearest neighbor; this was studied by Agarwal *et al.* [3]. This NN definition is much easier since the expected distance to each uncertain point can be computed separately, whereas the qualification probability involves the interaction among all uncertain points. However, the expected nearest neighbor is not a good indicator under large uncertainty (see [31] for details).

Finally, instead of returning only the nearest neighbor, one can ask to return the k nearest neighbors in a ranked order (the kNN problem). If we use expected distance, the ranking is straightforward. However, when qualification probabilities are considered, many different criteria for ranking the results are possible, leading to different problem variants.

Various combinations of these extensions have been studied in the literature; see, e.g., [10, 13, 22, 25, 31].

1.3 Our results

In this paper, we present efficient algorithms with proven guarantees on their performances for the nonzero NN problem as well as for computing the qualification probabilities.

Nonzero NNs. We first study (in Section 2.1) the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$. Suppose the uncertainty region of each $P_i \in \mathcal{P}$ is a disk and $d(\cdot,\cdot)$ is the L_2 metric. We show that $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $O(n^3)$ complexity, and that this bound is tight in the worst case. This significantly improves the bound in [28] and closes the problem. If the disks are pairwise disjoint and the ratio of their radii is at most λ, then the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ is $O(\lambda n^2)$. In either case, $\mathcal{V}_{\neq 0}(\mathcal{P})$ can be computed in $O(n^2 \log n + \mu)$ expected time, where μ is the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$. We can build a point-location structure on top of $\mathcal{V}_{\neq 0}(\mathcal{P})$ whose size is proportional to the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ and answer an $\mathsf{NN}_{\neq 0}$ query in $O(\log n + t)$ time, where t is the output size.

If each point in $\mathcal{P}$ has a discrete distribution of size at most k, then we show that $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $O(kn^3)$ complexity. Hence, we can answer an $\mathsf{NN}_{\neq 0}$ query in $O(\log(nk) + t)$ time using $O(kn^3)$ space.

Next, we consider (in Section 2.2) how quickly $\mathsf{NN}_{\neq 0}$ queries can be answered using less space. If the uncertainty region of each uncertain point is a disk, then an $\mathsf{NN}_{\neq 0}$ query can be answered in $O(\log n + t)$ time using $O(n^{1+\varepsilon})$ space, for any constant $\varepsilon > 0$, where t is the output size. If each uncertain point has at most k possible locations, then an $\mathsf{NN}_{\neq 0}$ query can be answered in $O(\log(nk) + t)$ time using $O((nk)^{2+\varepsilon})$ (for any $\varepsilon > 0$) space, or in $O((nk)^{1/2+\varepsilon} + t)$ time using $O(nk)$ space, where t is the output size.

Computing qualification probabilities. We present two algorithms for computing the qualification probabilities efficiently. The first (see Section 3.1) is a Monte-Carlo algorithm for estimating $\pi_i(q)$ for any P_i and q within error ε with high probability. First we argue that if each uncertain point has a discrete distribution of size poly(n), then we can estimate $\pi_i(q)$ within error ε by using $s_\varepsilon = O(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon})$ random instantiations of $\mathcal{P}$. (Note that there are at most $1/\varepsilon$ P_i's for which $\pi_i(q) > \varepsilon$.) Consequently, we can preprocess $\mathcal{P}$ into an index of size $O(\frac{n}{\varepsilon^2} \log \frac{n}{\varepsilon})$ so that for any query point $q \in \mathbb{R}^2$, $\pi_i(q)$ for all P_i's can be estimated within error ε in $O(\frac{1}{\varepsilon^2} \log \frac{n}{\varepsilon} \log n)$ time, with probability at least $1 - 1/n$. The algorithms explicitly computes the estimates of $\pi_i(q)$'s for at most s_ε points and sets the estimate to 0 for the rest of the points. This index can also be used to find the (approximate) most likely NN and the threshold NN within the same time bound. We also show that this approach works even if the distribution of each P_i is continuous.

Next, we describe (in Section 3.2) a deterministic algorithm for computing $\pi_i(q)$ approximately provided that the distribution of each P_i is discrete. Let $P_i = \{p_{i1}, \ldots, p_{ik}\}$ and $w_{ij} = \Pr[P_i \text{ is } p_{ij}]$. We set $\rho = \frac{\max w_{ij}}{\min w_{ij}}$, where maximum and minimum are taken over all the location probabilities of points in $S = \bigcup_{i=1}^n P_i$. We show that $\mathcal{P}$ can be preprocessed into an index of $O(n)$ size so that for any $q \in \mathbb{R}^2$ and for any $\varepsilon > 0$, $\pi_i(q)$, for all $i \le n$, can be computed with error at most ε in $O(\rho k \log(\rho/\varepsilon) + \log n)$ time. Our result shows that there are at most $m(\rho, \varepsilon) = \rho k \ln(\rho/\varepsilon) + k - 1$ points of $\mathcal{P}$ for which $\pi_i(q) > \varepsilon$. The algorithm explicitly estimates $\pi_i(q)$ for at most $m(\rho, \varepsilon)$ points and sets the estimate to 0 for the rest of the points. As earlier, this index can be used to solve the most likely NN and the threshold NN problem approximately within the same time bound.

Finally, we present experimental results, in Section 4, to demonstrate the efficacy of our approach for estimating quantification probabilities.

2. NONZERO PROBABILISTIC NN

In this section, we describe algorithms for answering $\mathsf{NN}_{\neq 0}$ queries. We first describe algorithms for computing $\mathcal{V}_{\neq 0}$ so that an $\mathsf{NN}_{\neq 0}$ query can be answered in logarithmic time by preprocessing $\mathcal{V}_{\neq 0}$ for point-location queries, and then describe indexing methods for answering $\mathsf{NN}_{\neq 0}$ queries using less space.

2.1 Nonzero Probabilistic Voronoi Diagram

Let $\mathcal{P}$ be a set of n uncertain points as described earlier. We analyze the combinatorial structure of $\mathcal{V}_{\neq 0}(\mathcal{P})$ and describe algorithms for constructing it. We first consider the case when the distribution of each point is continuous and then consider the discrete case.

Continuous case. For simplicity, we assume that the uncertainty region of each P_i is a circular disk D_i of radius r_i centered at c_i.

We first observe that the actual pdf of P_i is not important for computing $\mathcal{V}_{\neq 0}(\mathcal{P})$. What really matters is the uncertainty region D_i. More precisely, for each $1 \le i \le n$ and for $q \in \mathbb{R}^2$, let

$$\Delta_i(q) = \max_{p \in D_i} d(q,p) = d(q,c_i) + r_i,$$
$$\delta_i(q) = \min_{p \in D_i} d(x,q) = \max\{d(q,c_i) - r_i, 0\}$$

be the maximum and minimum possible distance, respectively, from q to a P_i.

The proof of the following lemma is straightforward.

LEMMA 2.1. *For a point $x \in \mathbb{R}^2$, a point $P_i \in \mathcal{P}$ belongs to $\mathsf{NN}_{\neq 0}(x, \mathcal{P})$ if and only if*

$$\delta_i(x) < \Delta_j(x) \text{ for all } 1 \le j \ne i \le n.$$

Let $\Delta \colon \mathbb{R}^2 \to \mathbb{R}$ denote the *lower envelope*[2] of $\Delta_1, \ldots, \Delta_n$;

[2]The lower envelope, L_F, of a set F of functions is their pointwise minimum, i.e., $L_F(x) = \min_{f \in F} f(x)$. The upper envelope, U_F, of F is the pointwise maximum, i.e., $U_F(x) = \max_{f \in F} f(x)$.

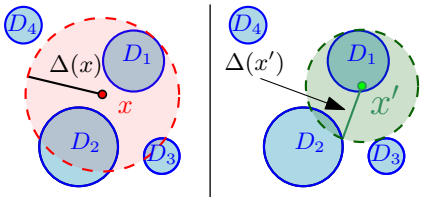


Figure 2. $\mathcal{P} = \{P_1, \ldots, P_5\}$. $\Delta(x) = \Delta_1(x)$, $\mathsf{NN}_{\neq 0}(x, \mathcal{P}) = \{P_1, P_2, P_3\}$, $\Delta(x') = \Delta_1(x')$, $\mathsf{NN}_{\neq 0}(x', \mathcal{P}) = \{P_1, P_2\}$, and x' lies on an edge of $\mathcal{V}_{\neq 0}(\mathcal{P})$.

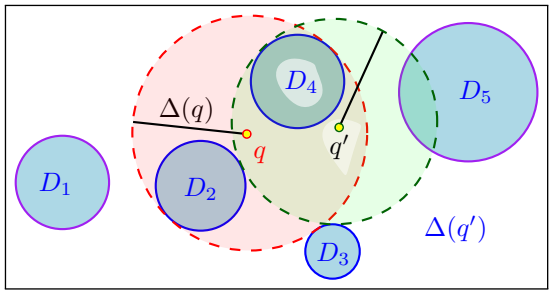


Figure 3. The point q is a breakpoint of γ_3 and q' is an intersection point of γ_2 and γ_3.

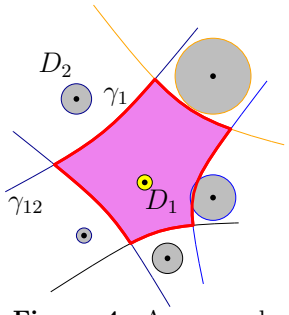


Figure 4. An example of γ_1.

that is, for any $q \in \mathbb{R}^2$,

$$\Delta(q) = \min_{1 \le i \le n} \Delta_i(q).$$

The projection of the graph of $\Delta(x)$ onto the xy-plane is the additive-weighted Voronoi diagram of the points $c_1, \ldots, c_n$, where the weight of c_i is r_i, and the weighted distance from q to c_i is $d(q, c_i) + r_i$, for $i = 1, \ldots, n$. Let $\mathbb{M}$ denote this planar subdivision. It has linear complexity and each of its edges is a hyperbolic arc; see [8]. Lemma 2.1 implies that, for any $q \in \mathbb{R}^2$,

$$\mathsf{NN}_{\neq 0}(q, \mathcal{P}) = \{P_i \mid \delta_i(q) < \Delta(q)\}. \tag{3}$$

See Fig. 2. It also implies that, as we move x continuously in $\mathbb{R}^2$, $\mathsf{NN}_{\neq 0}(x, \mathcal{P})$ remains the same until $\delta_i(x)$, for some $1 \le i \le n$, becomes equal to $\Delta(x)$ (e.g., x' in Fig. 2). The above was also observed in previous work. See, e.g. [12, 14]. Using this observation we can now characterize $\mathcal{V}_{\neq 0}(\mathcal{P})$.

For $i = 1, \ldots, n$, let $\gamma_i = \{x \in \mathbb{R}^2 \mid \delta_i(x) = \Delta(x)\}$ be the zero set of the function $\Delta(x) - \delta_i(x)$. Set $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$.

The curve γ_i partitions the plane into two open regions: $\Delta(x) < \delta_i(x)$ and $\Delta(x) > \delta_i(x)$. By Eq. (3), $P_i \in \mathsf{NN}_{\neq 0}(x, \mathcal{P})$ for all points x inside the latter region and for none of the points x inside the former region. It is well known that for any fixed $j \neq i$, $\gamma_{ij} = \{x \in \mathbb{R}^2 \mid \delta_i(x) = \Delta_j(x)\}$ is a hyperbolic curve [8]. The curve γ_i is composed of pieces of γ_{ij}, for $j \neq i$. We refer to the endpoints of these pieces as *breakpoints* of γ_i. They are the intersection points of γ_i with an edge of $\mathbb{M}$ and correspond to points q such that the disk of radius $\Delta(q)$ centered at q touches (at least) two disks of $\mathcal{D}$ from inside, touches D_i from outside, and does not contain any disk of $\mathcal{D}$ in its interior. See Fig. 3. Formally, we say that a disk D_1 touches a disk D_2 from the *outside* (resp. *inside*) if $\partial D_1 \cap \partial D_2 \neq \emptyset$ and $\operatorname{int} D_1 \cap \operatorname{int} D_2 = \emptyset$ (resp. $\operatorname{int} D_2 \subseteq \operatorname{int} D_1$).

LEMMA 2.2. *The curve γ_i, $1 \le i \le n$, has at most $2n$ breakpoints, and it can be computed in $O(n \log n)$ time.*

PROOF. Let $\Gamma_i = \{\gamma_{ij} \mid j \neq i, 1 \le j \le n\}$. It can be verified that a ray emanating from c_i intersects γ_{ij}, for any $j \neq i$, in at most one point, so γ_{ij} can be viewed as the graph of a function in polar coordinates with c_i as the origin. That is, let $\gamma_{ij} \colon [0, 2\pi) \to \mathbb{R}_{\ge 0}$, where $\gamma_{ij}(\theta)$ is the distance from c_i to γ_{ij} in direction θ. Then γ_i is the lower envelope of Γ_i. Since each pair of arcs in Γ_i intersects at most twice, a well-known result on lower envelopes implies that γ_i has at most $2n$ breakpoints, and that it can be computed in $O(n \log n)$ time [29]. See Fig. 4 for an example. □

Let $\mathcal{A}(\Gamma)$ denote the planar subdivision induced by Γ: its vertices are the breakpoints of γ_i's and the intersection points of two curves in Γ, its edges are the portions of γ_i's between two consecutive vertices, and its cells are the maximal connected regions of Γ that do not intersect any curve of Γ. We refer to vertices, edges, and cells of $\mathcal{A}(\Gamma)$ as its 0-, 1-, and 2-dimensional *faces*.

For a face ϕ (of any dimension), and for any two points $x, y \in \phi$, the sets $\{P_i \mid \delta_i(x) < \Delta(x)\}$ and $\{P_j \mid \delta_j(y) < \Delta(y)\}$ are the same; we denote this set by $\mathcal{P}_\phi$.

LEMMA 2.3. *Let $x \in \mathbb{R}^2$ be a point lying in a face ϕ of $\mathcal{A}(\Gamma)$. Then $\mathsf{NN}_{\neq 0}(x, \mathcal{P}) = \mathcal{P}_\phi$.*

For a subset $\mathcal{T} \subseteq \mathcal{P}$, let $\operatorname{cell}_{\neq 0}(\mathcal{T})$ be as defined in Eq. (2). An immediate corollary of the above lemma is:

COROLLARY 2.4. *(i) For any $\mathcal{T} \subseteq \mathcal{P}$, $\operatorname{cell}_{\neq 0}(\mathcal{T}) \neq \emptyset$ if and only if there is a face ϕ of $\mathcal{A}(\Gamma)$ with $\mathcal{T} = \mathcal{P}_\phi$.*
(ii) The planar subdivision $\mathcal{A}(\Gamma)$ coincides with $\mathcal{V}_{\neq 0}(\mathcal{P})$.

We now bound the complexity of $\mathcal{A}(\Gamma)$ and thus of $\mathcal{V}_{\neq 0}(\mathcal{P})$.

THEOREM 2.5. *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$ such that the uncertainty region of each point is a disk. Then $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $O(n^3)$ complexity. Moreover, it can be computed in $O(n^2 \log n + \mu)$ expected time, where μ is the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$.*

PROOF. Since $\mathcal{V}_{\neq 0}(\mathcal{P})$ is a planar subdivision, the number of edges and cells in it is proportional to the number of its vertices, so it suffices to bound the number of vertices. By Lemma 2.2, each γ_i has $O(n)$ breakpoints, so there are a total of $O(n^2)$ breakpoints. We claim that each pair of curves γ_i and γ_j intersect $O(n)$ times — each such intersection point corresponds to a point $v \in \mathbb{R}^2$ such that the disk of radius $\Delta(v)$ centered at v touches D_i and D_j from the outside and another disk D_k of $\mathcal{D}$, the one realizing the value of $\Delta(v)$, from the inside (e.g., q' in Fig. 3). For a fixed k, it can be shown that there are at most two points v such that $\delta_i(v) = \delta_j(v) = \Delta_k(v)$. Hence, the number of vertices in $\mathcal{V}_{\neq 0}(\mathcal{P})$ is $O(n^3)$, as claimed.

By Lemma 2.10, one can first compute all these curves in Γ in $O(n^2 \log n)$ time, and then compute the planar subdivision $\mathcal{A}(\Gamma)$ of Γ in $O(\mu)$ time using randomized incremental method [6], where μ is the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$. Hence $\mathcal{V}_{\neq 0}(\mathcal{P})$ can be computed in $O(n^2 \log n + \mu)$ expected time. □

Remarks. This bound holds even if the uncertainty region of each point is a *semialgebraic* set of *constant description complexity*, i.e., each region is defined by Boolean operations (union, intersection, and complementation) of a constant number of bivariate polynomial inequalities of constant maximum degree each.

Next we show that the above upper bound is tight in the worst case.

THEOREM 2.6. *There exists a set $\mathcal{P}$ of n uncertain points for which $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $\Omega(n^3)$ vertices.*

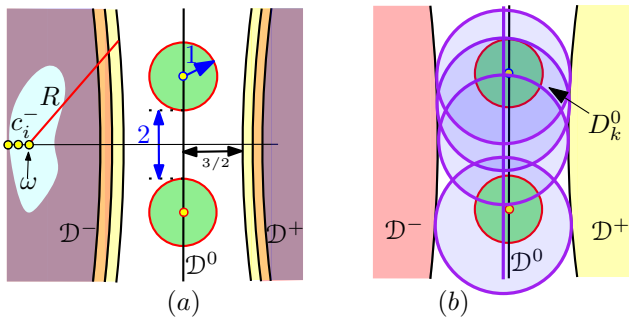


Figure 5. (a) $\Omega(n^3)$ lower bound construction with $m = 3$; only some disks are drawn. (b) Illustration of the proof.

PROOF. Assume that $n = 4m$ for some $m \in \mathbb{N}^+$. We choose two parameters $R = 8n^2$ and $\omega = 1/n^2$. We construct three families of disks: $\mathcal{D}^- = \{D_1^-, \ldots, D_m^-\}$, $\mathcal{D}^+ = \{D_1^+, \ldots, D_m^+\}$, and $\mathcal{D}^0 = \{D_1^0, \ldots, D_{2m}^0\}$. The radius of all disks in $\mathcal{D}^- \cup \mathcal{D}^+$ is R and their centers lie on the x-axis; the radius of all disks in $\mathcal{D}^0$ is 1 and their centers lie on the y-axis. More precisely, for $1 \le i, j \le m$, the center of D_i^- is $c_i^- = (-R - 3/2 - (i-1)\omega, 0)$ and the center of D_j^+ is $c_j^+ = (R + 3/2 + (j-1)\omega, 0)$, and for $1 \le k \le 2m$, the center of D_k^0 is $(0, 4(k-m) - 2)$. See Fig. 5 (a).

We claim that for every triple i, j, k with $1 \le i, j \le m$ and $1 \le k \le 2m$, there are two disks touching D_i^- and D_j^+ from the outside and D_k^0 from the inside and not containing any disk of $\mathcal{D}^- \cup \mathcal{D}^+ \cup \mathcal{D}^0$ in its interior. See Fig. 5(b).

Fix such a triple i, j, k. Since the radius of D_i^- and D_j^+ is the same, the locus b_{ij} of the centers of disks that simultaneously touch D_i^- and D_j^+ from the outside is the bisector of their centers, i.e., b_{ij} is the vertical line $x = (x(c_i^-) + x(c_j^+))/2 = (j-i)\omega/2$. Let σ_{ij} denote the intersection point of b_{ij} and the x-axis; $\sigma_{ij} = (\frac{1}{2}(j-i)\omega, 0)$. A point on b_{ij} can be represented by its y-coordinate; we will not distinguish between the two. For y-value a, let ξ_a be the disk centered at a and simultaneously touching D_i^- and D_j^+. The radius of ξ_a is

$$\begin{aligned}\|a - c_i^-\| - R &= \sqrt{a^2 + \|c_i^- - \sigma_{ij}\|^2} - R \\ &= \sqrt{a^2 + \left(R + 3/2 + \left(\frac{i+j}{2} - 1\right)\omega\right)^2} - R.\end{aligned}$$

The radius of ξ_a is thus at least $3/2$, and for $a \in [-4m, 4m]$, it is at most 2 (using the fact that $R \ge 8n^2$ and $\omega = 1/n^2$). Hence for $a \in [-4m, 4m]$, ξ_a contains at most one disk of $\mathcal{D}^0$ in its interior, and obviously ξ_a does not contain any disk of $\mathcal{D}^- \cup \mathcal{D}^+$ in its interior.

Let $a_k = 4(k-m) - 2$. Then the disk ξ_{a_k} contains D_k^0 in its interior because the distance between the centers of D_k^0 and ξ_{a_k} is at most $m\omega \le 1/(4n)$, the radius of D_k^0 is 1, and the radius of ξ_{a_k} is at least $3/2$. On the other hand, the disk ξ_a for $a = a_k \pm 2$ does not contain D_k^0 in its interior because the radius of ξ_a is at most 2 and the distance between the center of D_k^0 and ξ_a is at least 2. Therefore, by a continuity argument, there is a value $a^+ \in [a_k, a_k + 2]$ at which ξ_{a^+} touches D_k^0 from the inside. Similarly, there is a value $a^- \in [a_k - 2, a_k]$ at which ξ_{a^-} touches D_k^0 from the inside.

This proves the claim that there are two disks touching D_i^- and D_j^+ from the outside and D_k^0 from the inside and not containing any disk of $\mathcal{D}^- \cup \mathcal{D}^+ \cup \mathcal{D}^0$ in its interior. In other words, each triple i, j, k contributes two vertices to $\mathcal{V}_{\neq 0}(\mathcal{P})$. Hence $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $\Omega(n^3)$ vertices. □

Remarks. A more careful construction gives an $\Omega(n^3)$ lower bound on the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ even if all disks in $\mathcal{D}$ have the same radius.

Next, if the disks in $\mathcal{D}$ are pairwise disjoint and the ratio of the radii of the largest to the smallest disk is bounded by λ, then we prove a refined bound on the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ that depends on λ.

LEMMA 2.7. *If $\mathcal{P} = \{P_1, \ldots, P_n\}$ is a set of n uncertain points in $\mathbb{R}^2$ such that their uncertainty regions are pairwise-disjoint disks with radii in the range $[1, \lambda]$, a pair of curves in Γ intersects in $O(\lambda)$ points.*

PROOF. Fix a pair of curves γ_1 and γ_2, let D_1 and D_2 be the corresponding disks, and let c_1 and c_2 be their centers, respectively. By applying rotation and translation to the plane, we can assume D_1 and D_2 are centered on the x-axis, with D_1 to the left of D_2.

For a parameter t, $1 \le t \le \lambda$, let $\mathcal{D}$ denote the set of all the disks associated with $\mathcal{P}$, excluding D_1 and D_2, with radii between t and $2t$. An intersection point $q \in \gamma_1 \cap \gamma_2$ corresponds to a *witness* disk W centered at q that touches both D_1 and D_2 from the outside, touches exactly one other disk $E \in \mathcal{D}$ from the inside, and properly contains no disks of $\mathcal{D}$. The family of disks that touch both D_1 and D_2 from the outside is a *pencil*, which sweeps over portion of the plane as the tangency points with D_1 and D_2 move continuously (see Fig. 5(b)). A disk of $\mathcal{D}$ can contribute at most two intersection points to $\gamma_1 \cap \gamma_2$, as its boundary gets swept over at most twice by the circles of the pencil.

For a disk $E \in \mathcal{D}$, if its tangency point with its witness disk W is on the top portion of W (i.e., we break ∂W into two curves, *top* and *bottom*, at W's tangency points with D_1 and D_2) then it is a *top tangency event*, otherwise it is a *bottom tangency event*. Let $\mathcal{D}_1$ (resp. $\mathcal{D}_2$) be the set of disks in $\mathcal{D}$ that are closer to D_1 (resp. D_2). See Fig. 6(a).

Below we bound the number of top tangency events involving disks in $\mathcal{D}_2$. Other tangency events are handled by a symmetric argument.

We remove from $\mathcal{D}_2$ all the disks at distance at most $T = \xi t$ from D_2, where ξ is a sufficiently large constant. The ring with outer radius $\mathsf{r}(D_2) + 4T$ and inner radius $\mathsf{r}(D_2)$ has area $\alpha = \pi\big((\mathsf{r}(D_2) + 4T)^2 - (\mathsf{r}(D_2))^2\big) = O(T\mathsf{r}(D_2) + T^2)$. Disks removed from $\mathcal{D}_2$ have the following properties:

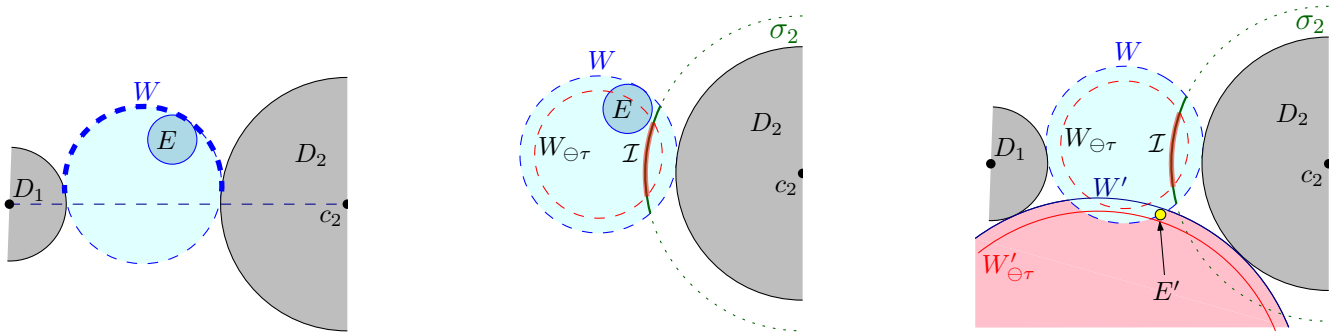
(i) they are interior-disjoint,
(ii) their radius is $\in [t, 2t]$,
(iii) they are contained in the aforementioned ring, and
(iv) the area of each such disk is at least πt^2.

Hence the number of removed disks is

$$O((T\mathsf{r}(D_2) + T^2)/t^2) = O(\lambda/t),$$

as $\mathsf{r}(D_2) \le \lambda$.

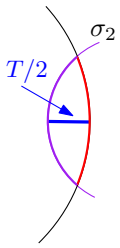
Consider the circle σ_2 of radius $\mathsf{r}(D_2) + T/2$ centered at c_2. Consider any disk $E \in \mathcal{D}_2$ and its witness disk W touching both D_1 and D_2 from the outside. Let $W_{\ominus\tau}$ be



(a) (b) (c)

Figure 6. An illustration for the proof of Lemma 2.7.

the disk concentric with W with radius $\mathsf{r}(W) - \tau$, where $\tau = 4t$. (Note that since E has not been removed from $\mathcal{D}_2$, $\mathsf{r}(W) \geq (T + 2t)/2$; in particular it is larger than $T/2$.) The disk $W_{\ominus\tau}$ is interior-disjoint from all disks in $\mathcal{D}_2$, as E touches W from inside and W cannot fully contain any other disks from $\mathcal{D}_2$.



The witness disk W covers an arc of length at least $T/2$ on σ_2. Indeed, neither of these two disks covers the center of the other, and the inner distance between the two intersection arcs is $T/2$, see figure on the right. Similarly, let $\mathcal{I}(E)$ be the arc $W_{\ominus\tau} \cap \sigma_2$. By the same argument, we have that $\mathcal{I}(E)$ is of length at least $T/2 - \tau = \Omega(t)$.

The perimeter of σ_2 is $2\pi(\mathsf{r}(D_2) + T/2) = O(\lambda)$, so if the arcs $\mathcal{I}(E)$, for $E \in \mathcal{D}_2$, are pairwise disjoint, we are done, as this implies that there could be at most $\lambda/(T/2 - \tau) = O(\lambda/t)$ such arcs and thus the size of the original $\mathcal{D}_2$ is $O(\lambda/t)$. See Fig. 6(b). We will therefore proceed to prove that any two such arcs are disjoint.

So, consider two disks $E, E' \in \mathcal{D}_2$, both realizing a top tangency event. Let W (resp. W') be the witness disk that is tangent to D_1, D_2 and E (resp. E'). Assume that the tangency of W with D_2 is clockwise to the tangency of W' with D_2 (i.e., E is "above" E'). If $W_{\ominus\tau}$ and $W'_{\ominus\tau}$ are disjoint then their corresponding arcs are disjoint. Otherwise, as we already observed, E' and $W_{\ominus\tau}$ are disjoint. Furthermore, it is easy to verify that E' must lie in the region "between" σ_2, $W_{\ominus\tau}$, and $W'_{\ominus\tau}$, and therefore the arcs $\mathcal{I}(E)$ and $\mathcal{I}(E')$ are disjoint, as claimed; refer to Fig. 6(c).

We now repeat the above counting argument, for $t = 1, 2, 4, \ldots, 2^m$, where $m = \lceil \log_2 \lambda \rceil$. We get that the number of intersection points between γ_1 and γ_2 is bounded by $\sum_{i=1}^{m} O(\lambda/2^i) = O(\lambda)$, as claimed. □

THEOREM 2.8. *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$ such that their uncertainty regions are pairwise-disjoint disks and that the ratio of the largest and the smallest radii of the disks is at most λ. Then the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ is $O(\lambda n^2)$, and it can be computed in $O(n^2 \log n + \mu)$ expected time, where μ is the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$. Furthermore, there exists such a set $\mathcal{P}$ of uncertain points for which $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $\Omega(n^2)$ complexity.*

PROOF. The upper bound on the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ follows from Lemma 2.7. By the same argument as in the proof of Theorem 2.5, $\mathcal{V}_{\neq 0}(\mathcal{P})$ can be computed in $O(n^2 \log n + \mu)$ time, where μ is the number of vertices in $\mathcal{V}_{\neq 0}(\mathcal{P})$. The lower-bound construction is omitted from this abstract due to lack of space. □

We store the index i of each uncertain point P_i instead of P_i itself. If we store $\mathcal{P}_\phi$ for each cell ϕ of $\mathcal{V}_{\neq 0}(\mathcal{P})$ explicitly, the size increases by a factor of n. However, we observe that for two adjacent cells ϕ, ϕ' of $\mathcal{V}_{\neq 0}(\mathcal{P})$, i.e., two cells that share a common edge, $|\mathcal{P}_\phi \oplus \mathcal{P}_{\phi'}| = 1$, where $\oplus$ denotes the symmetric difference of two sets. Therefore, using a persistent data structure [18], we can store $\mathcal{P}_\phi$ for all cells of $\mathcal{V}_{\neq 0}(\mathcal{P})$ in $O(\mu)$ space, where μ is the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$, so that for any cell ϕ, $\mathcal{P}_\phi$ can be retrieved in $O(\log n + |\mathcal{P}_\phi|)$ time. By combining this with the planar point-location indexing schemes [17], we obtain the following:

THEOREM 2.9. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, and let μ be the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$. Then $\mathcal{V}_{\neq 0}(\mathcal{P})$ can be preprocessed in $O(\mu \log \mu)$ time into an index of size $O(\mu)$ so that for a query point $q \in \mathbb{R}^2$, $\mathsf{NN}_{\neq 0}(q, \mathcal{P})$ can be computed in $O(\log n + t)$ time, where t is the output size.*

Remarks. This bound can be extended to the case when each uncertainty region is an α-fat semialgebraic set of constant description complexity. A set C is called α-fat, if there exist two concentric disks, $D \subseteq C \subseteq D'$, such that the ratio between the radii of D' and D is at most α. The constant of proportionality also depends on α and the number and maximum degree of polynomials defining the uncertainty regions.

Discrete case. We now analyze the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ when the distribution of each point P_i in $\mathcal{P}$ is discrete. Let $P_i = \{p_{i1}, \ldots, p_{ik}\}$. For $1 \leq j \leq k$, let $w_{ij} = \Pr[P_i \text{ is } p_{ij}]$. As in the previous section, for a point x, let $\Delta_i(x) = \max_{1 \leq j \leq k} d(x, p_{ij})$ and $\delta_i(x) = \min_{1 \leq j \leq k} d(x, p_{ij})$. Note that the projection of the graph of Δ_i (resp. δ_i) onto the xy-plane is the farthest-point (resp. nearest-point) Voronoi diagram of P_i. Let $\Delta(x) = \min_{1 \leq i \leq n} \Delta_i(x)$. For each i, let $\gamma_i = \{x \in \mathbb{R}^2 \mid \delta_i(x) = \Delta(x)\}$, and set $\Gamma = \{\gamma_1, \ldots, \gamma_n\}$. Then $\mathcal{V}_{\neq 0}(\mathcal{P})$ is the planar subdivision induced by Γ, as defined above. We need the following lemma to bound the complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$.

LEMMA 2.10. *For any pair i, j, $1 \leq i \neq j \leq n$, let $\gamma_{ij} = \{x \in \mathbb{R}^2 \mid \delta_i(x) = \Delta_j(x)\}$, then γ_{ij} is a convex polygonal curve with $O(k)$ vertices.*

PROOF. Let $p \in \mathbb{R}^2$ be a fixed point. Then we define a linear function $g\colon \mathbb{R}^2 \to \mathbb{R}$ as

$$g(x) = d^2(x,p) - \|x\|^2 = \|p\|^2 - 2\langle x, p\rangle.$$

For $1 \le i \le n$, define $\varphi_i(x) = \min_{1\le j\le k} g(x, p_{ij})$ and $\Phi_i(x) = \max_{1\le j\le k} g(x, p_{ij})$. Then for any pair i, j, $\delta_i(x) = \Delta_j(x)$ if and only if $\varphi_i(x) = \Phi_j(x)$. Hence, γ_{ij} is also the zero set of the function $\Phi_j(x) - \varphi_i(x)$.

Note that Φ_j is the upper envelope of k linear functions, and that it is a piecewise-linear concave function, and that φ_i, the lower envelope of k linear functions, is a piecewise-linear convex function. Hence $\Phi_j(x) - \varphi_i(x)$ is a piecewise-linear concave function, which implies that $\gamma_{ij} = \{x \in \mathbb{R}^2 \mid \Phi_j(x) = \varphi_i(x)\}$ is a convex polygonal curve. Since γ_{ij} is the projection of the intersection curve of the graphs of Φ_j and φ_i, each of which is a convex polyhedron with at most k faces, γ_{ij} has $O(k)$ vertices. □

THEOREM 2.11. *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$, where each P_i has a discrete distribution of size at most k. The complexity of $\mathcal{V}_{\neq 0}(\mathcal{P})$ is $\mu = O(kn^3)$ in the worst case, and it can be computed in expected time $O(n^2 \log n + \mu)$. Furthermore, it can be preprocessed into an index of size $O(\mu)$ so that an $\mathsf{NN}_{\neq 0}(q)$ query can be answered in $O(\log \mu + t)$, where t is the output size.*

PROOF. We follow the same argument as in the proof of Theorem 2.5. We need to bound the number of intersection points between a pair of curves γ_i and γ_j. Fix an index u. Let $\gamma_{iu} = \{x \in \mathbb{R}^2 \mid \delta_i(x) = \Delta_u(x)\}$ and $\gamma_{ju} = \{x \in \mathbb{R}^2 \mid \delta_j(x) = \Delta_u(x)\}$. By Lemma 2.10, each of γ_{iu} and γ_{ju} is a convex polygonal curve in $\mathbb{R}^2$ with $O(k)$ vertices. Since two convex polygonal curves in general position with n_1 and n_2 vertices intersect in at most $2(n_1 + n_2)$ points, γ_{iu} and γ_{ju} intersect at $O(k)$ points. Hence γ_i and γ_j intersect at $O(nk)$ points, implying that $\mathcal{V}_{\neq 0}(\mathcal{P})$ has $O(kn^3)$ vertices. The running time follows from the proof of Theorem 2.5. □

2.2 Indexing schemes for $\mathsf{NN}_{\neq 0}$ queries

Despite the maximum size of $\mathcal{V}_{\neq 0}$ being $\Theta(n^3)$ or $\Theta(n^2)$, we can obtain indexing schemes with less space such that $\mathsf{NN}_{\neq 0}$ queries can be answered in poly-logarithmic or sublinear time. We consider both continuous and discrete cases.

An $\mathsf{NN}_{\neq 0}(q)$ query is answered in two stages. The first stage computes $\Delta(q)$, and the second stage computes all points $P_i \in \mathcal{P}$ for which $\delta_i(q) < \Delta(q)$. We build a separate index for each stage.

Continuous case. We assume that the uncertainty region of each point P_i is a disk D_i of radius r_i centered at c_i. Recall from Section 2.1 that the projection of the graph of the function Δ onto the xy-plane, a planar subdivision $\mathbb{M}$, is the weighted Voronoi diagram of the point set $c_1, \ldots, c_n$, and it has linear complexity. Hence $\mathbb{M}$ can be preprocessed in $O(n \log n)$ time into an index of size $O(n)$ so that for a query point $q \in \mathbb{R}^2$, $\Delta(q)$ can be computed in $O(\log n)$ time.

Next, the problem of reporting $\mathsf{NN}_{\neq 0}(q)$ reduces to reporting all disks of $D_1, \ldots, D_n$ that intersect the disk of radius $\Delta(q)$ centered at q. Using the approach described in [4], $D_1, \ldots, D_n$ can be preprocessed into an index of size $O(n^{1+\varepsilon})$, for any constant $\varepsilon > 0$, so that all t disks intersecting a query disk can be reported in $O(\log n + t)$ time. We thus obtain the following:

THEOREM 2.12. *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$ so that the uncertainty region of each P_i is a disk. Then $\mathcal{P}$ can be preprocessed into an index of size $O(n^{1+\varepsilon})$, for any $\varepsilon > 0$, so that an $\mathsf{NN}_{\neq 0}(q)$ query can be answered in $O(\log n + t)$ time, where t is the output size.*

Remarks. (i) Note that Theorem 2.12 gives a better result than Theorem 2.9 if the uncertainty regions of $\mathcal{P}$ are allowed to intersect, but the Voronoi–diagram-based index is much simpler and practical.

(ii) If we use L_1 or L_∞ metric to compute the distance between points and use disks in L_1 or L_∞ metric (i.e., a diamond or a square), then an $\mathsf{NN}_{\neq 0}(q)$ query can be answered in $O(\log^2 n + t)$ time using $O(n \log^2 n)$ space.

Discrete case. If the distribution of each P_i is discrete, then the functions Δ_i and δ_i are complex and thus the index for $\mathsf{NN}_{\neq 0}(q)$ queries is more involved. First we observe that the problem of reporting all points $\mathcal{P}_i \in \mathcal{P}$ such that $\delta_i(q) \le R$ for a query point $q \in \mathbb{R}^2$ and $R > 0$, can be formulated as a colored disk range reporting. Namely, we color all k points of P_i with color i. Let $S = \bigcup_{i=1}^n P_i$. Then given a disk D of radius R centered at q, we wish to report the colors of all points in S that lie inside D — each color should be reported only once. Following the same approach as in [19], this can be done with $O(\log^2(nk) + t)$ query time using $O((nk)^{2+\varepsilon})$ space (for any $\varepsilon > 0$), or with $O((nk)^{1/2+\varepsilon} + t)$ query time using $O((nk)\log^2(nk))$ space.

Alternatively, using standard reduction from reporting to emptiness, this can be solved using, space $O(nk \log n)$, preprocessing $O(nk \log^2 n)$, and $O((1+t)\log^2 n)$ query time. Indeed, build a balanced tree over the colors, and for each internal node, build a standard emptiness range searching data-structure for all the disks having the colors stored in this subtree. Here, the emptiness data-structure is a point-location data-structure in a weighted additive Voronoi diagram. Now, given a query disk, traverse this color tree, recursing into a subtree if the emptiness data-structure reports that a disk in this subtree intersects the query. An emptiness query takes $O(\log n)$ time, and $O(t \log n)$ nodes in the tree are visited by the query process.

It thus suffices to describe how we compute $\Delta(q)$ for a query point $q \in \mathbb{R}^2$. Recall that the projection of Δ_i onto the xy-plane, for $1 \le i \le n$, is the farthest-point Voronoi diagram of P_i, and that Δ is the lower envelope of $\Delta_1, \ldots, \Delta_n$. Following the same argument as by Huttenlocher *et al.* [21], we can prove the following.

LEMMA 2.13. *The xy-projection of the graph of the function Δ is a planar subdivision with $O(n^2 k \alpha(n^2 k))$ vertices, and it can be computed in $O(n^2 k \log(nk))$ time, where $\alpha(\cdot)$ is the inverse Ackerman function.*

If the convex hulls of the point clouds are disjoint, the problem is significantly easier, see [11].

Hence by preprocessing the projection of Δ for point-location queries, $\Delta(q)$ can be computed in $O(\log(nk))$ time, for any query point q.

If we wish to construct a linear-size index, we rely on multi-level partition–tree-based [5] indexing schemes. We sketch the main idea and omit the details. Let $S = \bigcup_{i=1}^n P_i$, which is a set of nk (certain) points in $\mathbb{R}^2$. For a point $q \in \mathbb{R}^2$, let $S(q) = \{p_1, \ldots, p_n\}$ where p_i is the farthest neighbor of q in P_i. We build a partition tree $\mathcal{T}$ on S and the farthest-point Voronoi diagrams of $P_1, \ldots, P_n$ of size $O(nk)$, which basically

constructs a family $\mathcal{F} = \{e_1, \ldots, e_m\}$ of "canonical" subsets of S such that:

(i) $\sum_i |e_i| = O(nk)$;

(ii) for any query point $q \in \mathbb{R}^2$, $S(q)$ can be represented as the union of $O((nk)^{1/2+\varepsilon})$ (for any $\varepsilon > 0$) canonical subsets of $\mathcal{F}$, denoted by $\mathcal{F}(q)$.

$\mathcal{T}$ can be constructed in $O(nk \log(nk))$ time, and using the hierarchical structure of $\mathcal{T}$, $\mathcal{F}(q)$ for a query point q can be computed in $O((nk)^{1/2+\varepsilon})$ time. Next, we build a linear-size index on each e_i for answering NN queries in $O(\log n)$ time. Putting everything together, the overall size of the index is $O(nk)$ and it can be constructed in $O(nk \log(nk))$ time. See [5, 26] for details.

Given a query point $q \in \mathbb{R}^2$, we first compute $\mathcal{F}(q)$, then for each $e \in \mathcal{F}(q)$, we compute the nearest neighbor of q in e, and finally choose the nearest one among them. The total query time is $O((nk)^{1/2+\varepsilon} \log n) = O((nk)^{1/2+\varepsilon'})$ for any $\varepsilon' > \varepsilon$.

Hence, we obtain the following:

THEOREM 2.14. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each of size at most k. $\mathcal{P}$ can be preprocessed into an index of size $O((nk)^{2+\varepsilon})$, for any $\varepsilon > 0$, so that an $\mathsf{NN}_{\neq 0}(q)$ query can be answered in $O(\log(nk) + t)$ time, or into an index of size $O(nk)$ with $O((nk)^{1/2+\varepsilon} + t)$ query time, where t is the output size. The preprocessing times are $O((nk)^{2+\varepsilon})$ and $O(nk \log(nk))$ time, respectively.*

3. QUANTIFICATION PROBABILITIES

We begin with exact algorithms for uncertain point sets, in which each uncertain point has k possible locations. We can build a structure called the *probabilistic Voronoi diagram* $\mathcal{V}_{\Pr}(\mathcal{P})$ that decomposes $\mathbb{R}^2$ into a set of cells, so that any point q in a cell has the same $\pi_i(q)$ value for all $P_i \in \mathcal{P}$; that is, for any point q in this cell, we know exactly the probability of each point $P \in \mathcal{P}$ being the NN of q.

LEMMA 3.1. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with at most k possible locations, then the complexity of $\mathcal{V}_{\Pr}(\mathcal{P})$ is $O(n^4k^4)$.*

PROOF. There are nk possible locations. Each pair of possible locations determines a bisector, resulting in $O(n^2k^2)$ bisectors. These bisectors partition the plane into $O(n^4k^4)$ convex cells so the order of all distances to each of the nk possible locations, and thus also all the qualification probabilities, are preserved within each cell. Therefore the resulting planar subdivision is a refinement of $\mathcal{V}_{\Pr}(\mathcal{P})$, and thus $O(n^4k^4)$ is an upper bound on the complexity of $\mathcal{V}_{\Pr}(\mathcal{P})$. □

Note, that the related notion of most likely NN is not stable in the sense that a single possible location of point that is possibly far from a query can affect which point is the most likely NN. Since the $\mathcal{V}_{\Pr}(\mathcal{P})$ is too large to be efficient in practice, we explore how to approximate $\pi_i(q)$.

3.1 Monte Carlo Algorithm

In this section we describe a simple Monte Carlo approach to build an index for quickly computing $\hat{\pi}_i(q)$ for all P_i for any query point q, which approximates the quantification probability $\pi_i(q)$. For a fixed value s, to be specified later, the preprocessing step has s rounds. In the jth round the algorithm creates a sample $R_j = \{r_{j1}, r_{j2}, \ldots, r_{jn}\} \subseteq \mathbb{R}^2$ by choosing each r_{ji} using the distribution of P_i. For each $j \leq s$, we construct the Voronoi diagram $\mathsf{Vor}(R_j)$ in $O(n \log n)$ time and preprocess it for point-location queries in additional $O(n \log n)$ time.

To estimate quantification probabilities of a query q, we initialize a counter $c_i = 0$ for each point P_i. For each R_j, we find the point r_{ji} whose cell in $\mathsf{Vor}(R_j)$ contains the query point q, and increment c_i by 1. Finally we estimate $\hat{\pi}_i(q) = c_i/s$. Note that at most s distinct c_i's have nonzero values, so we can implicitly set the others to 0.

Discrete case. If each $P_i \in \mathcal{P}$ has a discrete distribution of size k, then this algorithm can be implemented very efficiently. Each r_i can be selected in $O(\log k)$ time after preprocessing each P_i, in $O(k)$ time, into a balanced binary tree with total weight calculated for each subtree [27]. Thus total preprocessing takes $O(s(n(\log n + \log k)) + nk) = O(nk + sn \log(nk))$ time and $O(sn)$ space, and each query takes $O(s \log n)$ time.

It remains to determine the value of s so that $|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon$ for all P_i and all queries q, with probability at least $1 - \delta$. For fixed q, P_i, and instantiation R_j, let X_i be the random indicator variable, which is 1 if r_i is the NN of q and 0 otherwise. Since $\mathsf{E}[\mathrm{X}_i] = \pi_i(q)$ and $\mathrm{X}_i \in \{0, 1\}$, applying a Chernoff-Hoeffding bound to

$$\hat{\pi}_i(q) = \frac{c_i}{s} = \frac{1}{s} \sum_i \mathrm{X}_i,$$

we observe that

$$\Pr\left[|\hat{\pi}_i(q) - \pi_i(q)| \geq \varepsilon\right] \leq 2 \exp(-2\varepsilon^2 s). \tag{4}$$

For each cell of $\mathcal{V}_{\Pr}(\mathcal{P})$, we choose one point, and let Q be the resulting set of points. If $|\hat{\pi}_i(q) - \pi_i(q)| \leq \varepsilon$ for every point $q \in Q$, then $|\hat{\pi}_i(q) - \pi_i(q)| \leq \varepsilon$ for every point $q \in \mathbb{R}^2$. Since there are n different values of i, by applying the union bound to (4), the probability that there exist a point $q \in \mathbb{R}^2$ and an index $i \leq n$ with $|\hat{\pi}_i(q) - \pi_i(q)| \geq \varepsilon$ is at most $2n|Q| \exp(-2\varepsilon^2 s)$. Hence, by setting

$$s = \frac{1}{2\varepsilon^2} \ln \frac{2n|Q|}{\delta},$$

$|\hat{\pi}_i(q) - \pi_i(q)| \leq \varepsilon$ for all $q \in \mathbb{R}^2$ and for all $i \leq n$, with probability at least $1 - \delta$. By Lemma 3.1, $|Q| = O(n^4k^4)$, so we obtain the following result.

THEOREM 3.2. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a discrete distribution of size k, and let $\varepsilon, \delta \in (0, 1)$ be two parameters. $\mathcal{P}$ can be preprocessed, in*

$$O(nk + (n/\varepsilon^2) \log(nk) \log(nk/\delta))$$

time, into an index of size $O((n/\varepsilon^2) \log(nk/\delta))$, which computes, for any query point $q \in \mathbb{R}^2$, in $O((1/\varepsilon^2) \log(nk/\delta) \log n)$ time, a value $\hat{\pi}_i(q)$ for every P_i such that $|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon$ for all i with probability at least $1 - \delta$.

Continuous case. There are two technical issues in extending this technique and analysis to continuous distributions. First, we instantiate a certain point r_i from each P_i. Herein we assume the representation of the pdf is such that this can be done in constant time for each P_i.

Second, we need to bound the number of distinct queries that need to be considered to apply the union bound as we did above. Since $\pi_i(q)$ may vary continuously with the query location, unlike the discrete case, we cannot hope for a bounded number of distinct results. However, we just need

to define a finite set $\bar{Q}$ of query points so that any query $q \in \mathbb{R}^2$ has $\max_i |\pi_i(q) - \pi_i(q')| \leq \varepsilon/2$ for some $q' \in \bar{Q}$. Then we can choose s large enough so that it permits at most $\varepsilon/2$ error on each query in $\bar{Q}$. Specifically, choosing $s = O((1/\varepsilon^2)\log(n|\bar{Q}|/\delta))$ is sufficient, so all that remains is to bound $|\bar{Q}|$.

To choose $\bar{Q}$, we show that each pdf of P_i can be approximated with a discrete distribution of size $O((n^2/\varepsilon^2)\log(n/\delta))$, and then reduce the problem to the discrete case.

For parameters $\alpha > 0$ and $\delta' \in (0,1)$, set

$$k(\alpha) = \frac{c}{\alpha^2}\log\frac{1}{\delta'},$$

where c is a constant. For each $i \leq n$, we choose a random sample $\bar{P}_i \subset P_i$ of size $k(\alpha)$, according to the distribution defined by the location pdf f_i of P_i. We regard $\bar{P}_i$ as an uncertain point with uniform location probability. Set $\bar{\mathcal{P}} = \{\bar{P}_1, \ldots, \bar{P}_n\}$.

For a point $q \in \mathbb{R}^2$, let $\bar{G}_{q,i}$ denote the cdf of the distance between q and $\bar{P}_i$, i.e., $\bar{G}_{q,i}(r) = \Pr[d(q,\bar{P}_i) \leq r]$, or equivalently, it is the probability of $\bar{P}_i$ lying in the disk of radius r centered at q. A well-known result in the theory of random sampling [24, 30] implies that for all $r \geq 0$,

$$\left|G_{q,i}(r) - \bar{G}_{q,i}(r)\right| \leq \alpha, \tag{5}$$

with probability at least $1-\delta'$, provided that the constant c in $k(\alpha)$ is chosen sufficiently large.

Let $\bar{\pi}_i(q)$ denote the probability of $\bar{P}_i$ being the NN of q in $\bar{\mathcal{P}}$. We prove the following:

Lemma 3.3. *For any $q \in \mathbb{R}^2$, and for any fixed $i \leq n$,*

$$|\pi_i(q) - \bar{\pi}_i(q)| \leq \alpha n,$$

with probability at least $1-\delta'$.

Proof. Recall that by (1),

$$\pi_i(q) = \int_0^\infty g_{q,i}(r)\prod_{j\neq i}(1 - G_{q,j}(r))\mathrm{d}r.$$

Using (5), and the fact that $G_{q,j}(r), \bar{G}_{q,j}(r) \in [0,1]$ for all j, we obtain

$$\pi_i(q) \leq \int_0^\infty g_{q,i}(r)\prod_{j\neq i}(1 - \bar{G}_{q,j}(r))\mathrm{d}r + (n-1)\alpha.$$

Note that $\prod_{j\neq i}(1-\bar{G}_{q,j}(r))$ is the probability that the closest point of q in $\bar{\mathcal{P}}\setminus\{\bar{P}_i\}$ is at least distance r away from q. Let $h_{q,i}$ be the pdf of the distance between q and its closest point in $\bar{\mathcal{P}} \setminus \{\bar{P}_i\}$. Then

$$\prod_{j\neq i}(1 - \bar{G}_{q,j}(r)) = \int_r^\infty h_{q,i}(\theta)\mathrm{d}\theta.$$

Therefore

$$\pi_i(q) \leq \int_0^\infty \int_r^\infty g_{q,i}(r)h_{q,i}(\theta)\mathrm{d}\theta\mathrm{d}r + (n-1)\alpha.$$

By reversing the order of integration, we obtain

$$\begin{aligned}\pi_i(q) &\leq \int_0^\infty \int_0^\theta h_{q,i}(\theta)g_{q,i}(r)\mathrm{d}r\mathrm{d}\theta + (n-1)\alpha\\ &= \int_0^\infty h_{q,i}(\theta)G_{q,i}(\theta)\mathrm{d}\theta + (n-1)\alpha\\ &\leq \int_0^\infty h_{q,i}(\theta)(\bar{G}_{q,i}(\theta)+\alpha)\mathrm{d}\theta + (n-1)\alpha\\ &\quad \text{(using (5))}\\ &= \int_0^\infty h_{q,i}(\theta)\bar{G}_{q,i}(\theta)\mathrm{d}\theta + n\alpha\\ &= \bar{\pi}_i(q) + n\alpha.\end{aligned}$$

A similar argument shows that $\pi_i(q) \geq \bar{\pi}_i(q) - n\alpha$. This completes the proof of the lemma. □

Thus by setting $\alpha = \varepsilon/2n$, a random sample $\bar{P}_i$ of size $O((n^2/\varepsilon^2)\log(n/\delta))$ from each P_i ensures that

$$|\pi_i(q) - \bar{\pi}_i(q)| \leq \varepsilon/2 \tag{6}$$

for all queries. By choosing $\delta' = \delta/2n$, (6) holds for all $i \leq n$ with probability at least $1-\delta/2$.

We consider $\mathcal{V}_{\mathrm{Pr}}(\bar{\mathcal{P}})$, choose one point from each of its cells, and set $\bar{Q}$ to be the resulting set of points. For a point $q \in \mathbb{R}^2$, let $\bar{q} \in \bar{Q}$ be the representative point of the cell of $\mathcal{V}_{\mathrm{Pr}}(\bar{\mathcal{P}})$ that contains q. Then $|\pi_i(q) - \bar{\pi}_i(\bar{q})| < \varepsilon/2$ for all points $q \in \mathbb{R}^2$ and $i \leq n$, with probability at least $1-\delta/2$.

Now applying the analysis for the discrete case on the point set $\bar{\mathcal{P}}$, if we choose

$$s = O\Big(\frac{1}{\varepsilon^2}\log\frac{n|\bar{Q}|}{\delta}\Big),$$

then $|\bar{\pi}_i(q) - \hat{\pi}_i(q)| < \varepsilon$ for all points $q \in \mathbb{R}^2$ and for all $i \leq n$ with probability at least $1-\delta/2$. Since

$$|\bar{P}_i| = k\Big(\frac{\varepsilon}{2n}\Big) = O\Big(\frac{n^2}{\varepsilon^2}\log\frac{n}{\delta}\Big),$$

by Lemma 3.1,

$$|\bar{Q}| = O\Big(n^4\Big(k\Big(\frac{\varepsilon}{2n}\Big)\Big)^4\Big) = O\Big(\frac{n^{12}}{\varepsilon^8}\log^4\frac{n}{\delta}\Big).$$

Putting everything together, we obtain the following.

Theorem 3.4. *Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$ so a random instantiation of P_i can be performed in $O(1)$ time, let $0 < \varepsilon, \delta < 1$. $\mathcal{P}$ can be preprocessed in $O((n/\varepsilon^2)\log(n/\varepsilon\delta)\log n)$ time into an index of size $O((n/\varepsilon^2)\log(n/\varepsilon\delta))$, which computes for any query point $q \in \mathbb{R}^2$, in $O((1/\varepsilon^2)\log(n/\varepsilon\delta)\log n)$ time, a value $\hat{\pi}_i(q)$ for every P_i such that $|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon$ for all i with probability at least $1-\delta$.*

3.2 Spiral Search Algorithm

If the distribution of each point in $\mathcal{P}$ is discrete, then there is an alternative approach to approximate the quantification probabilities for a given query q: set a parameter $m > 1$, choose m points of $S = \bigcup_{i=1}^n P_i$ that are closest to q, and use only these m points to estimate $\pi_i(q)$ for each P_i. We show this works for a small value of m when, for each P_i, each location is approximately equally likely, but is not efficient if we have no bounds on the weights of these locations.

Let $P_i = \{p_{i1}, \ldots, p_{ik}\}$ and $w_{ij} = \Pr[P_i = p_{ij}]$. Set $S = \bigcup_{i=1}^n P_i$. We refer to the quantity

$$\rho = \frac{\max w_{ij}}{\min w_{ij}} \tag{7}$$

as the *spread* of location probabilities. Set

$$m(\rho,\varepsilon) = \rho k \ln(\rho/\varepsilon) + k - 1.$$

Fix a query point $q \in \mathbb{R}^2$, and let $\bar{S} \subseteq S$ be the $m(\rho, \varepsilon)$ nearest neighbors of q in S. Let $\bar{P}_i = \bar{S} \cap P_i$, and $\bar{\mathcal{P}} = \{\bar{P}_1, \ldots, \bar{P}_n\}$. Note that $\sum_{p_{i,a} \in \bar{P}_i} w_{i,a}$ is not necessarily equal to 1, so we cannot regard $\bar{P}_i$ as an uncertain point, but still it will be useful to think of $\bar{P}_i$ as an uncertain point.

For a set Y of points and another point $\xi \in \mathbb{R}^2$, let

$$Y[\xi] = \{p \in Y \mid d(q,p) \leq d(q,\xi)\}.$$

Then for a point $p := p_{i,a} \in P_i$, $\pi_p(q)$, the probability that p is the nearest neighbor of q in $\mathcal{P}$ is

$$\pi_p(q) = w_{i,a} \prod_{j \neq i} \Big(1 - \sum_{p_{j,\ell} \in P_j[p]} w_{j,\ell}\Big). \tag{8}$$

Moreover,

$$\pi_i(q) = \sum_{p_{i,a} \in P_i} \pi_{p_{i,a}}(q). \tag{9}$$

For each $i \leq n$, we analogously define a quantity $\hat{\pi}_i(q)$ using (8) and (9) but replacing P_j with $\bar{P}_j$ for every $j \leq n$. Intuitively, if $\bar{\mathcal{P}}$ were a family of uncertain points, then $\hat{\pi}_i(q)$ would be the probability of $\bar{P}_i$ being the NN of q in $\bar{\mathcal{P}}$.

LEMMA 3.5. *For all $i \leq n$,*

$$|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon.$$

PROOF. Fix a point $p \in P_i$. Set $x_j = |P_j[p]|$ and $m = \sum_{j \neq i} x_j$. Note that each $w_{j,a}$ satisfies

$$1/\rho k \leq w_{j,a} \leq \rho/k.$$

Then for a point $p := p_{i,a} \in P_i$, we obtain using (8)

$$\begin{aligned}
\pi_p(q) &= w_{i,a} \prod_{j \neq i} \Big(1 - \sum_{p_\ell \in P_j[p]} w_{j,\ell}\Big) \\
&\leq \frac{\rho}{k} \prod_{j \neq i} \Big(1 - \frac{x_j}{\rho k}\Big) \\
&\leq \frac{\rho}{k} \prod_{j \neq i} \exp\left(-x_j/\rho k\right) = \frac{\rho}{k} \exp\left(-m/\rho k\right).
\end{aligned}$$

Thus any point $p \in P_i$ that has $m \geq \rho k \ln(\rho/\varepsilon)$ points in $\mathcal{P} \setminus \{P_i\}$ closer to q than itself, has probability at most ε/k of being the closest point to q. Since each $P_i \in \mathcal{P}$ consists of at most k points, the combined effect of all of these far points cannot contribute more than ε to the total probability that P_i is the nearest neighbor. Also $k-1$ points from P_i may also be closer to q than p. Thus if p is not an $m(\rho,\varepsilon)$-nearest neighbor of q in S, i.e., $p \notin \bar{P}_i$, then $\pi_p(q) < \varepsilon/k$. Hence,

$$\pi_i(q) \leq \sum_{p \in \bar{P}_i} \pi_p(q) + \varepsilon = \sum_{p \in \bar{P}_i} \hat{\pi}_p(q) + \varepsilon = \hat{\pi}_i(q) + \varepsilon.$$

This completes the proof of the lemma. □

For any i, if $P_i \cap \bar{S}(q) = \emptyset$, then we can implicitly set $\hat{\pi}_i(q)$ to 0. Finally, the following result shows that the m nearest neighbors of q in S can be chosen efficiently in $\mathbb{R}^2$.

LEMMA 3.6 (AFSHANI AND CHAN [1]). *Given a set of N points in $\mathbb{R}^2$, with $O(N \log N)$ expected preprocessing time and $O(N)$ space, we can return the closest m points to q in $O(m + \log N)$ time, for any query point $q \in \mathbb{R}^2$.*

We thus obtain the following result:

THEOREM 3.7. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, let ρ be the spread of location probabilities, and let $\varepsilon > 0$ be a parameter. $\mathcal{P}$ can be preprocessed in $O(nk \log(nk))$ expected time into an index of $O(nk)$ size, so that for a query point $q \in \mathbb{R}^2$ and a parameter $\varepsilon > 0$, it can compute, in time $O(\rho k \log(\rho/\varepsilon) + \log(nk))$, values $\hat{\pi}_i(q)$ for all $P_i \in \mathcal{P}$ such that $|\pi_i(q) - \hat{\pi}_i(q)| \leq \varepsilon$ for all $i \leq n$.*

Remarks. (i) Unfortunately, this approach is not efficient when the spread of location probabilities is unbounded. In this case, one may have to retrieve $\Omega(n)$ points. Another approach may be to ignore points with weight smaller than ε/k, since even k such weights from a single uncertain point P_i cannot contribute more than ε to $\pi_i(q)$. However, the union of all such points may distort other probabilities.

Consider the following example. Let $p_1 \in P_1 \in \mathcal{P}$ be the closest point to the query point q. Let $w(p_1) = 3\varepsilon$. Let the next $n/2$ closest points $p_3, \ldots, p_{n/2+2}$ be from different uncertain points $P_3, \ldots, P_{n/2+2}$ and each have weights $w(p) = 2/(n+2) \ll \varepsilon/k$. Let the next closest point $p_2 \in P_2 \in \mathcal{P}$ have weight $w(p_2) = 5\varepsilon$. With probability $\pi_{p_1}(q) = 3\varepsilon$ the nearest neighbor is p_1. The probability that p_2 is the nearest neighbor is $\pi_{p_2}(q) = (5\varepsilon)(1-3\varepsilon)(1-2/n)^{n/2} < (5\varepsilon)(1-3\varepsilon)(1/e) < 2\varepsilon$. Thus p_1 is more likely to be the nearest neighbor than p_2. However, if we ignore points $p_3, \ldots, p_{n/2+2}$ because they have small weights, then we calculate p_2 has probability $\hat{\pi}_{p_2}(q) = (1-3\varepsilon)(5\varepsilon) > 4\varepsilon$ for being the nearest neighbor. So $\pi_2(q)$ will be off by more than 2ε and it would incorrectly appear that p_2 is more likely to be the nearest neighbor than p_1.

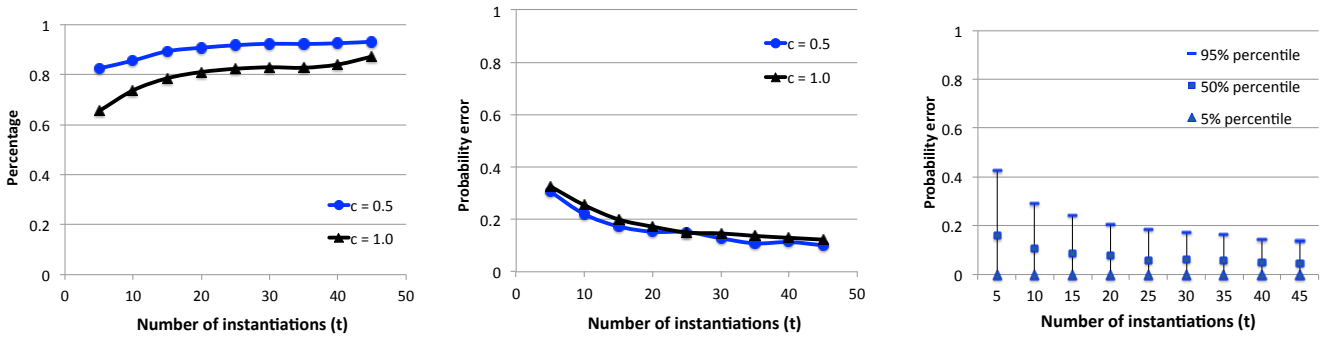
(ii) Though Lemma 3.6 is optimal theoretically, it is too complex to be implemented. Instead, one may use order-m Voronoi diagram to retrieve the m closest points (in unsorted order) to q. This would yield an index with $O(m(nk-m))$ space and $O(m(nk-m)\log(nk) + nk\log^3(nk))$ expected preprocessing time [2], while preserving the query time $(\log(nk) + m)$, where $m = O(\rho k \log(\rho/\varepsilon))$. Alternatively, one may use quad-trees and a branch-and-bound algorithm to retrieve m points of S closest to q [20].

4. EXPERIMENTAL RESULTS

We have conducted experiments on synthetic datasets to demonstrate the efficacy of our methods for estimating qualification probabilities.

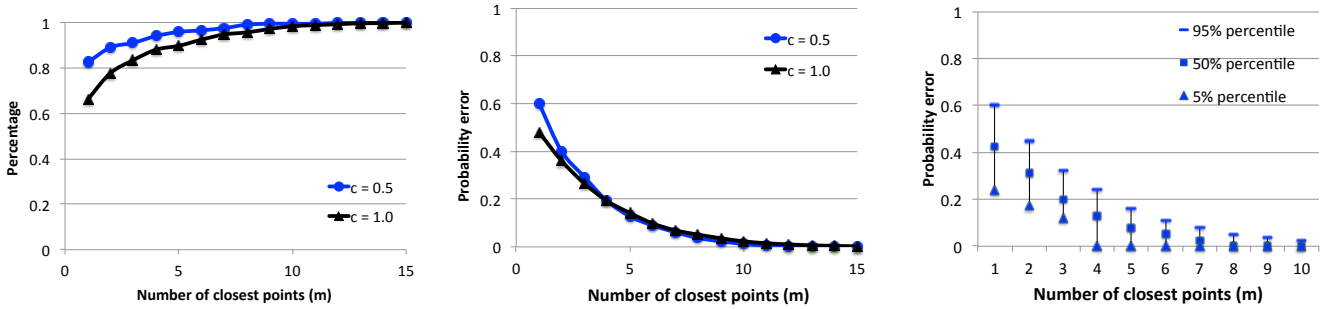
Experimental setup. We assume each uncertain point has a discrete distribution of size k. We set $r = \frac{c}{\sqrt{n}}$, where $c > 0$ is a parameter. The value of c indicates the level of uncertainty: the bigger value c is, the larger uncertainty each uncertain point has. We synthetically generated n uncertain points in two steps as follows: (1) For each uncertain point, we first generate a disk of radius r whose center is randomly chosen inside the unit square $[0,1]^2$. (2) We then choose k possible locations within the disk of each uncertain point. We chose k possible locations uniformly inside the disk (we also tried Gaussian distribution and got similar results). In our experiments, we set $n = 1000, k = 5$, and $c \in \{0.5, 1.0\}$.

Measuring the effectiveness. We test how effective the Monte Carlo method and the spiral-search methods are in computing the most likely nearest neighbor, NN_L, and the estimates of qualification probabilities. In the experiments, 1000 queries were issued for each input, and we measured the following three quantities:



(a) (b) (c)

Figure 7. Monte Carlo method: (a) percentage of NN_L; (b) probability error of NN_L; (c) probability error of all points.



(a) (b) (c)

Figure 8. Spiral search method: (a) percentage of NN_L; (b) probability error of NN_L; (c) probability error of all points.

(i) The percentage of trials in which the algorithm returns the correct NN_L.

(ii) The error in estimated qualification probability of NN_L. Specifically for each query, suppose P_i is the true NN_L, then the *NN_L probability error* is $|\pi_i - \hat{\pi}_i|$; we report the 90% percentile of these errors.

(iii) For each query q, we compute $\max_i |\pi_i(q) - \hat{\pi}_i(q)|$, the maximum error in probability. Among all 1000 trials, we report the 5%, 50%, and 95% percentiles of these errors. We only used $c = 0.5$ for quantity (iii).

Monte Carlo method. We tested how quantities (i)-(iii) changed as we varied t, the number of instantiations t. Fig. 7(a)–(c) show these quantities as t varies from 5 to 45. Not surprisingly, as t increases, the percentage of correct NN_L increases, and the probability errors decrease. The smaller uncertainty (as denoted by c) we have, the better performance. For both $c = 0.5$ and $c = 1.0$, the NN_L is returned correctly at least 80% of the times if $t \geq 20$, and this also generally provides probability error less than 0.15.

Spiral search method. We also tested how (i) – (iii) changed as we varied m, the number of closest points retrieved to estimate the qualification probabilities. Fig. 8(a)–(c) show these quantities as m varies from 1 to 15 (or 10 in Fig. 8(c)). Compared to the Monte Carlo approach, the spiral search method accuracy seems to converge much faster (although t versus m is not directly comparable). After only $m = 9$ closest points are retrieved, the NN_L is found more than 95% of the time, and the probability error goes to practically 0. Recall $k = 5$ so from these experiments it appears retrieving only $2k$ points to be effective. This method also seems less affected by the scale of uncertainty (parameter c). Since many practical k-nearest-neighbor algorithms exist, we believe this has the potential for practical use.

5. CONCLUSION

In this paper, we investigated NN queries in a probabilistic framework in which the location of each input point is specified as a probability distribution function. We presented efficient methods for returning all the non-zero probability points, estimating the quantification probabilities and using it for threshold NN queries. We also conducted some preliminary experiments to demonstrate the effectiveness of our methods. We conclude by mentioning two open problems:

(i) What is the complexity of the probabilistic Voronoi diagram? The bound proved in Lemma 3.1 is not tight, and it does not work for continuous distributions.

(ii) Extend the spiral search method to continuous distributions (at least for some simple, well-behaved distributions such as Gaussian), so that the query time is always sublinear.

Acknowledgments. P. Agarwal and W. Zhang are supported by NSF under grants CCF-09-40671, CCF-10-12254, and CCF-11-61359, by ARO grants W911NF-07-1-0376 and W911NF-08-1-0452, and by an ERDC contract W9132V-11-C-0003. B. Aronov is supported by NSF grants CCF-08-30691, CCF-11-17336, and CCF-12-18791, and by NSA MSP Grant H98230-10-1-0210. S. Har-Peled is supported by NSF grants CCF-09-15984 and CCF-12-17462.

6. REFERENCES

[1] P. Afshani and T. M. Chan. Optimal halfspace range reporting in three dimensions. In *Proc. 20th ACM-SIAM Sympos. Discrete Algs.*, pages 180–186, 2009.

[2] P. K. Agarwal, M. de Berg, J. Matoušek, and O. Schwarzkopf. Constructing levels in arrangements and higher order Voronoi diagrams. *SIAM J. Comput.*, 27:654–667, 1998.

[3] P. K. Agarwal, A. Efrat, S. Sankararaman, and W. Zhang. Nearest-neighbor searching under uncertainty. In *Proc. 31st ACM Sympos. Principles Database Syst.*, pages 225–236, 2012.

[4] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.

[5] P. K. Agarwal and J. Erickson. Geometric range searching and its relatives. In B. Chazelle, J. E. Goodman, and R. Pollack, editors, *Advances in Discrete and Computational Geometry*, pages 1–56. Amer. Math. Soc., 1999.

[6] P. K. Agarwal and M. Sharir. Arrangements and their applications. In J.-R. Sack and J. Urrutia, editors, *Handbook of Computational Geometry*, pages 49–119. Elsevier, 2000.

[7] C. Aggarwal. *Managing and Mining Uncertain Data*. Springer-Verlag, 2009.

[8] P. F. Ash and E. D. Bolker. Generalized Dirichlet tessellations. *Geometriae Dedicata*, 20:209–243, 1986.

[9] T. Bernecker, T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz, and A. Zuefle. A novel probabilistic pruning approach to speed up similarity queries in uncertain databases. In *Proc. 27th IEEE Int. Conf. Data Eng.*, pages 339–350, 2011.

[10] G. Beskales, M. A. Soliman, and I. F. IIyas. Efficient search for the top-k probable nearest neighbors in uncertain databases. *Proc. Int. Conf. Very Large Data.*, pages 326–339, 2008.

[11] P. Cheilaris, E. Khramtcova, and E. Papadopoulou. Randomized incremental construction of the Hausdorff Voronoi diagram of non-crossing clusters. Technical Report 2012/03, University of Lugano, 2012.

[12] R. Cheng, J. Chen, M. Mokbel, and C. Chow. Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data. In *Proc. 24th IEEE Int. Conf. Data Eng.*, pages 973–982, 2008.

[13] R. Cheng, L. Chen, J. Chen, and X. Xie. Evaluating probability threshold k-nearest-neighbor queries over uncertain data. In *Proc. 12th Int. Conf. Ext. Database Tech.*, pages 672–683, 2009.

[14] R. Cheng, D. V. Kalashnikov, and S. Prabhakar. Querying imprecise data in moving object environments. *IEEE Trans. Know. Data Eng.*, 16(9):1112–1127, 2004.

[15] R. Cheng, X. Xie, M. L. Yiu, J. Chen, and L. Sun. UV-diagram: A Voronoi diagram for uncertain data. In *Proc. 26th IEEE Int. Conf. Data Eng.*, pages 796–807, 2010.

[16] N. N. Dalvi, C. Ré, and D. Suciu. Probabilistic databases: Diamonds in the dirt. *Commun. ACM*, 52(7):86–94, 2009.

[17] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[18] J. R. Driscoll, N. Sarnak, D. D. Sleator, and R. E. Tarjan. Making data structures persistent. *J. Comput. Syst. Sci.*, 38:86–124, 1989.

[19] P. Gupta, R. Janardan, and M. Smid. Algorithms for generalized halfspace range searching and other intersection searching problems. *Comput. Geom. Theory Appl.*, 5:321–340, 1996.

[20] S. Har-Peled. *Geometric Approximation Algorithms*. Mathematical Surveys and Monographs. American Mathematical Society, 2011.

[21] D. P. Huttenlocher, K. Kedem, and M. Sharir. The upper envelope of Voronoi surfaces and its applications. In *Proc. 7th Annu. ACM Sympos. Comput. Geom.*, pages 194–203, 1991.

[22] P. Kamousi, T. M. Chan, and S. Suri. Closest pair and the post office problem for stochastic points. In *Proc. 12th Workshop Algorithms Data Struct.*, pages 548–559, 2011.

[23] H.-P. Kriegel, P. Kunath, and M. Renz. Probabilistic nearest-neighbor query on uncertain objects. In *Proc. 12th Int. Conf. Database Sys. Adv. App.*, pages 337–348, 2007.

[24] Y. Li, P. M. Long, and A. Srinivasan. Improved bounds on the sample complexity of learning. *J. Comput. Syst. Sci.*, 62(3):516–527, 2001.

[25] V. Ljosa and A. K. Singh. APLA: Indexing arbitrary probability distributions. In *Proc. 23rd IEEE Int. Conf. Data Eng.*, pages 946–955, 2007.

[26] J. Matoušek. Range searching with efficient hierarchical cuttings. *Discrete Comput. Geom.*, 10(2):157–182, 1993.

[27] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[28] J. Sember and W. Evans. Guaranteed Voronoi diagrams of uncertain sites. In *Proc. 20th Canad. Conf. Comput. Geom.*, 2008.

[29] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, 1995.

[30] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.

[31] S. M. Yuen, Y. Tao, X. Xiao, J. Pei, and D. Zhang. Superseding nearest neighbor search on uncertain spatial databases. *IEEE Trans. Know. Data Eng.*, 22(7):1041–1055, 2010.

[32] P. Zhang, R. Cheng, N. Mamoulis, M. Renz, A. Zufile, Y. Tang, and T. Emrich. Voronoi-based nearest neighbor search for multi-dimensional uncertain databases. In *Proc. 29th IEEE Int. Conf. Data Eng.*, 2013. to appear.