Nearest-Neighbor Searching Under Uncertainty[*]

Pankaj K. Agarwal
Department of Computer Science
Duke University
pankaj@cs.duke.edu

Alon Efrat
Department of Computer Science
The University of Arizona
alon@cs.arizona.edu

Swaminathan Sankararaman[*]
Department of Computer Science
Duke University
swami@cs.duke.edu

Wuzhou Zhang
Department of Computer Science
Duke University
wuzhou@cs.duke.edu

ABSTRACT

Nearest-neighbor queries, which ask for returning the nearest neighbor of a query point in a set of points, are important and widely studied in many fields because of a wide range of applications. In many of these applications, 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 and/or query point is specified as a probability density function and the goal is to return the point that minimizes the expected distance, which we refer to as the expected nearest neighbor (ENN). We present methods for computing an exact ENN or an ε-approximate ENN, for a given error parameter $0 < \varepsilon < 1$, under different distance functions. These methods build an index of near-linear size and answer ENN queries in polylogarithmic or sublinear time, depending on the underlying function. As far as we know, these are the first nontrivial methods for answering exact or ε-approximate ENN queries with provable performance guarantees.

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

Theory

Keywords

Indexing uncertain data, nearest-neighbor queries, expected nearest neighbor (ENN), approximate nearest neighbor

1. INTRODUCTION

Motivated by a wide range of applications, nearest neighbor searching has been studied in many different fields including computational geometry, database systems and information retrieval; see [7, 14] for surveys on this topic. In its simplest form, it asks for preprocessing a set S of n points in $\mathbb{R}^d$ into an index so that the nearest neighbor (NN) in S of a query point can be reported quickly. The earlier methods for answering NN queries assumed that the input points and the query points were precise. In many applications, such as sensor databases, location based services, face recognition, and mobile data, the location of data is imprecise. This has led to a flurry of activity on query processing over uncertain data, and algorithms for answering range query, top-k queries, skyline queries and NN queries have been proposed, among numerous results. See e.g. [4, 15] for recent work.

In this paper we are interested in answering NN queries over uncertain data —the location of input points or the query point is not precisely known, and we assume that it is given as a probability density function. The existing methods for answering NN queries on precise data cannot be applied directly to this setting and new methods are needed.

Our model. An *uncertain* point P in $\mathbb{R}^d$ is represented as a probability density function (pdf) $f_P : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$[1]. We assume f_P to be a simple pdf such as Gaussian distribution, inverse distance function, or a histogram; we make this notion more precise later. We also consider *discrete pdfs*, in which P is represented as a finite set $P = \{p_1, \cdots, p_k\} \subset \mathbb{R}^d$ along with a set $\{w_1, \cdots, w_k\} \subset [0, 1]$ where $w_i = \Pr\{P \text{ is } p_i\}$ and

[*] P.A. and W.Z. are supported by NSF under grants CNS-05-40347, CCF-06 -35000, IIS-07-13498, and CCF-09-40671, by ARO grants W911NF-07-1-0376 and W911NF-08-1-0452, by an NIH grant 1P50-GM-08183-01, and by a grant from the U.S.–Israel Binational Science Foundation. A.E. and S.S. are supported by NSF CAREER Grant 0348000.

[*] This work was partly undertaken while S.S. was at the University of Arizona.

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

PODS'12, May 21–23, 2012, Scottsdale, Arizona, USA.
Copyright 2012 ACM 978-1-4503-1248-6/12/05 ...$10.00.

$\sum_{i=1}^{k} w_i = 1$. Discrete pdfs arise in a wide range of applications [19, 23], e.g., because of multiple observations, and continuous pdfs can be approximated as discrete pdfs.

Let $\mathrm{d}(\cdot,\cdot)$ denote a distance function in $\mathbb{R}^d$; we consider L_1, L_2, L_∞-metrics or squared Euclidean distance[2]. For a given $\mathrm{d}(\cdot,\cdot)$, the *expected distance* between two independent uncertain points P and Q is defined as

$$\mathsf{Ed}(P,Q) = \iint f_P(x) f_Q(y) \mathrm{d}(x,y) dx dy.$$

If f_P and f_Q are discrete pdfs of size k each, then

$$\mathsf{Ed}(P,Q) = \sum_{i=1}^{k}\sum_{j=1}^{k} w_i w'_j \mathrm{d}(p_i, q_j),$$

where w_i, w'_j are the probabilities of P and Q being at p_i and q_j respectively. If Q is a (certain) point in $\mathbb{R}^d$, i.e., $Q = \{q\}$, then

$$\mathsf{Ed}(P,q) = \sum_{i=1}^{k} w_i \mathrm{d}(p_i, q).$$

We say that the *description complexity* of f_P is k if it can be represented using $O(k)$ parameters and certain basic primitive operations on f_P can be performed in $O(k)$ time. In particular, $\mathsf{Ed}(P,x)$, for any $x \in \mathbb{R}^2$, can be computed in $O(k)$ time, and the expected location $\int x f_P(x) dx$ of P, also called the *centroid* of P, can be computed in $O(k)$ time. Examples include a discrete pdf consisting of at most k points, and piecewise-constant or piecewise-linear pdf consisting of at most k pieces. Gaussian (under certain distance functions) and inverse-distance distributions have constant description complexity.

Let $\mathcal{P} = \{P_1, \cdots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^d$, each of which is independently chosen. For simplicity, let f_i denote f_{P_i}, the pdf of P_i. For an uncertain point Q, its *expected nearest neighbor* (ENN), denoted by $\varphi(\mathcal{P}, Q)$, is

$$\varphi(\mathcal{P}, Q) = \operatorname*{argmin}_{P \in \mathcal{P}} \mathsf{Ed}(P,Q).$$

For a parameter $0 < \varepsilon < 1$, we call a point $P \in \mathcal{P}$ an ε-*approximate* ENN (or ε-ENN, for brevity) of Q if

$$\mathsf{Ed}(P,Q) \le (1+\varepsilon)\mathsf{Ed}(\varphi(\mathcal{P},Q),Q).$$

Next, we introduce the notion of the *expected Voronoi diagram* of $\mathcal{P}$. For $1 \le i \le n$, we define the *expected Voronoi cell* $\mathrm{EVor}(P_i)$ as

$$\mathrm{EVor}(P_i) = \{x \in \mathbb{R}^d \mid \mathsf{Ed}(P_i, x) \le \mathsf{Ed}(P_j, x), \forall j\}.$$

The decomposition of $\mathbb{R}^d$ into maximal connected regions induced by $\mathrm{EVor}(P_i)$, $1 \le i \le n$, is called the expected Voronoi diagram, $\mathrm{EVD}(\mathcal{P})$ of $\mathcal{P}$. See Figure 2 for an example of an EVD in $\mathbb{R}^2$ where $\mathrm{d}(\cdot,\cdot)$ is the L_1 metric. A decomposition of $\mathbb{R}^d$ into connected cells, each cell τ labeled with $\lambda(\tau) \in \mathcal{P}$, is called an ε-*approximate* EVD of $\mathcal{P}$ (ε-$\mathrm{EVD}(\mathcal{P})$, for brevity), if for all $x \in \tau$, $\lambda(\tau)$ is an ε-ENN of x.

In this paper, we study the problem of answering exact or approximate ENN queries when input points are uncertain or the query is an uncertain point. We also study the problem of computing $\mathrm{EVD}(\mathcal{P})$ and ε-$\mathrm{EVD}(\mathcal{P})$.

Previous results. In the exact setting, Voronoi diagrams may be used to perform nearest neighbor searching among a set of n input data points in $\mathbb{R}^2$ with $O(n \log n)$ preprocessing time, $O(n)$ space and $O(\log n)$ query time. Unfortunately, the size of the Voronoi diagram is $\Theta(n^{\lceil d/2 \rceil})$ in $\mathbb{R}^d$. The best known method for answering an NN query, requires $O((n/m^{1/\lceil d/2 \rceil})$ polylog $n)$ query time for an $O(m)$-space structure, where $n < m < n^{\lceil d/2 \rceil}$ [3]. To obtain better performance, many researchers turned to approximate nearest neighbor searching: Given any $\varepsilon > 0$, a point p is an ε-approximate nearest neighbor of q if $\mathrm{d}(q,p) \le (1+\varepsilon)\mathrm{d}(q,p^*)$, where p^* is the actual nearest neighbor. Arya *et al.* [6] generalized space-time trade-offs for approximate nearest neighbor searching: Given a tradeoff parameter γ, where $2 \le \gamma \le 1/\varepsilon$, there exists an index of space $O(n\gamma^{d-1}\log(1/\varepsilon))$ that can answer queries in time $O(\log(n\gamma) + 1/(\varepsilon\gamma)^{(d-1)/2})$. There is also extensive work on answering approximate nearest neighbor queries using locality sensitive hashing, when d is not a constant and the goal is to have an algorithm whose query time is polynomial in d; see e.g. [5, 18, 20].

Different models have been proposed for geometric computing on *uncertain* data: mainly classified into *deterministic models* and *probabilistic models*. In deterministic models, each point is assumed to be inside a given region (see e.g. [9, 10, 29, 37]). Probabilistic models can be further classified into the *existential model* and the *locational model*. In the existential model, each point is assumed to appear with certain probability. Suri *et al.* [22] proposed a linear-space index with $O(\log n)$ query time to compute an ε-approximate value of the expected distance from a query point to its nearest neighbor when the dimension d is a constant.

In the locational model, the coordinates of each point are assumed to be chosen from a known probability distribution. In this paper, we focus on the locational model of uncertainty to study the problem of nearest neighbor searching. When the data is uncertain but the query is exact, researchers have studied top-k probable nearest neighbor, ranking queries, probabilistic nearest neighbor, and superseding nearest neighbor [8, 11, 12, 19, 23, 28, 36, 39]. Ljosa *et al.* [28] investigated the expected k-NN under L_1 metric using and obtained ε-approximation. Cheng *et al.* [11] studied the probabilistic nearest neighbor query that returns those uncertain points whose probabilities of being the nearest neighbor are higher than some threshold, allowing some given error in the answers. All of these methods were based on heuristics and did not provide any guarantee on the query time in the worst case. Moreover, recent results that rely on Voronoi diagram for supporting nearest neighbor queries under uncertainty cannot be adapted to answer ENN (see [13, 21, 33]). We are not aware of any index that uses near-linear space and returns in sublinear time the expected nearest neighbor or a point that is the most likely nearest neighbor.

The problem of computing the expected nearest neighbor when the queries are uncertain but the input is exact is closely related to the *aggregate nearest neighbors* (ANN) problem. Given a set of points $\mathcal{P}$ in a metric space X with a distance function d, the aggregate nearest neighbor to a set of query points Q is defined as $\mathrm{ANN}(Q, \mathcal{P}) = \arg\min_{p \in \mathcal{P}} g(p, Q)$, where $g(p,Q)$ is some aggregation function of the distances from points of Q to p. The aggregation functions commonly considered are SUM, corresponding to a summation of the individual distances and MAX, corresponding to the minimization of the maximum distance.

[2] the squared Euclidean distance between two points $p, q \in \mathbb{R}^d$ is $||p-q||^2$ where $||\cdot||$ is the L_2 metric.

If the pdf is a uniform distribution, the ENN problem is the same as the ANN problem under the SUM aggregation function. Several heuristics are known for answering ANN queries [17, 25, 27, 30, 31, 34, 38]. Li *et al.* [24] provided a polynomial-time approximation scheme for ANN queries under the MAX aggregation function. Li *et al.* [26] presented approximation schemes under MAX and SUM functions for any metric space as long as an efficient nearest-neighbor algorithm is provided. For the SUM function, they provide a 3-approximation which, to the best of our knowledge, is the best approximation-factor known previously. See also [27].

Our results. We present efficient algorithms for answering ENN queries under various distance functions. For simplicity, we state the results in $\mathbb{R}^2$, i.e., $\mathcal{P} = \{P_1, \cdots, P_n\}$ is a set of n uncertain points in $\mathbb{R}^2$. We assume that the description complexity of the pdf of each P_i is at most k.

Squared Euclidean distance. If $\mathrm{d}(\cdot,\cdot)$ is the squared Euclidean distance, then we show that a set $\mathcal{P}$ of n uncertain points can be replaced by a set $\overline{\mathcal{P}}$ of n weighted points such that the weighted Voronoi diagram of $\overline{\mathcal{P}}$ under $\mathrm{d}(\cdot,\cdot)$ (also called the *power diagram* of $\overline{\mathcal{P}}$ [7]) is the same as EVD($\mathcal{P}$). In particular, EVD($\mathcal{P}$) has linear size and can be computed in $O(n \log n + nk)$ time if the pdf of each P_i has description complexity at most k. Furthermore, EVD($\mathcal{P}$) can be preprocessed in $O(n \log n)$ time into a linear size index so that an ENN query for a (certain) point can be answered in $O(\log n)$ time. If the query is also an uncertain point Q, then we show that $\varphi(\mathcal{P}, Q)$ is the same as $\varphi(\mathcal{P}, \overline{q})$, where $\overline{q} = \int x f_Q(x)dx$ is the centroid of Q.

Rectilinear distance. We assume that each pdf f_i is a discrete pdf consisting of k points. We show that EVD($\mathcal{P}$) has $O(n^2k^2\alpha(n))$ complexity and it can be computed in the same time. We also show that there exists a set $\mathcal{P}$ of n uncertain points with $k = 2$ such that EVD($\mathcal{P}$) has $\Omega(n^2)$ vertices. We then describe an index of size $O(k^2 n \log^2 n)$ that can answer an ENN query in $O(\log^3(kn))$ time. The index can be built in $O(k^2 n \log^3 n)$ time. Next, we show that a set $\mathcal{P}$ of n (certain) points in $\mathbb{R}^2$ can be stored in an index of size $O(n \log^2 n)$ so that for an uncertain point with discrete pdf consisting of at most k points, an ENN query can be answered in $O(k^2 \log^3 n)$ time. The index can be built in $O(n \log^2 n)$ time. We note that L_1 and L_∞ metrics are closely related, so these results also hold for the L_∞ metric.

Euclidean distance. Since the expected distance function under Euclidean distance is algebraically quite complex even for discrete pdfs, we focus on answering ε-ENN queries. First, we show that the expected distance to an uncertain point P can be approximated by a piecewise-constant function, consisting of $O(1/\varepsilon^2 \log(1/\varepsilon))$ pieces, plus the distance to the centroid of P. Using this result, we construct, in $O((n/\varepsilon^2)\log^2 n \log(n/\varepsilon)\log(1/\varepsilon))$ time, an ε-EVD of $\mathcal{P}$ of size $O((n/\varepsilon^2)\log(1/\varepsilon))$; each face of the subdivision is the region lying between two nested squares —it can be partitioned into at most four rectangles, so that the ε-EVD is a rectangular subdivision. Moreover, for any query point, we can return its ϵ-ENN in $O(\log(n/\varepsilon))$ time.

Finally, we show that a set $\mathcal{P}$ of n (certain) points in $\mathbb{R}^2$ can be stored in an index of linear size so that, for an uncertain point with pdf of k description complexity, an ENN query can be answered in $O((k/\varepsilon^2)\log(1/\varepsilon)\log n)$ time. The index can be built in $O(n \log n)$ time. These results can be extended to any L_p metric.

We remark that most of our algorithms extend to higher dimensions, but the query time increases exponentially with d; we mention specific results in the appropriate sections.

Outline of the paper. We begin in Section 2 by describing a few geometric concepts that will be useful. Section 3 describes our algorithms for the squared Euclidean distance function, Section 4 for the L_1 and L_∞ metrics, and Section 5 for the Euclidean distance. We conclude by making a few final remarks in Section 6.

2. PRELIMINARIES

In this section, we describe a few geometric concepts and data structures that we need.

Lower envelopes and Voronoi diagrams. Let $F = \{f_1, \cdots, f_n\}$ be a set of n bivariate functions. The lower envelope of F is defined as

$$\mathbb{L}_F(x) = \min_{1 \le i \le n} f_i(x),$$

and the *minimization diagram* of F, denoted by $\mathbb{M}(F)$, is the projection of the graph of $\mathbb{L}_F$. $\mathbb{M}(F)$ is a planar subdivision in which the same function appears on the lower envelope for all points inside a cell. The (combinatorial) complexity of $\mathbb{L}_F$ and $\mathbb{M}(F)$ is the number of vertices, edges, and faces in $\mathbb{M}(F)$. If we define $f_i(x)$ to be $\mathsf{Ed}(P_i, x)$, then EVD($\mathcal{P}$) is the minimization diagram of the resulting functions. Figure 1 shows the Voronoi diagram of a set of exact points as the minimization diagram of its distance functions.

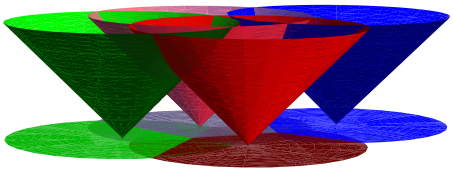


Figure 1. Euclidean Voronoi diagram of (certain) points as the minimization diagram of their distance functions.

The notion of lower envelope and minimization diagram can be extended to partially defined functions: f_i is defined over a region $V_i \subseteq \mathbb{R}^2$, then $\mathbb{L}_F(x)$ is the minimum over all functions f_i of F that are defined at x, i.e., $x \in V_i$. Let $\mathcal{R}$ be a set of polygons, each consisting of a constant number of vertices (e.g., triangles, rectangles) in $\mathbb{R}^3$. By viewing each of them as the graph of a partially-defined linear function, we can define the lower envelope $\mathbb{L}_\mathcal{R}$ and minimization diagram $\mathbb{M}(\mathcal{R})$ of $\mathcal{R}$. It is known that the complexity of $\mathbb{M}(\mathcal{R})$ is $\Theta(n^2\alpha(n))$ and that it can be computed in $O(n^2\alpha(n))$ time [35], where $\alpha(n)$ is the inverse Ackermann function.

Compressed quadtree. A square is called *canonical* if its side length is 2^l for an integer l and its bottom-left corner is $(2^l a, 2^l b)$ for some integers a, b. Note that two canonical squares are either disjoint or one of them is contained in the other.

A *quadtree* on a canonical square H is a 4-way tree T, each of whose nodes v is associated with a canonical square $\Box_v \subseteq H$. The root of T is associated with H itself. The squares associated with the children of a node v are obtained by dividing each side of $\Box_v$ into two halves, thereby dividing $\Box_v$ into four congruent canonical squares. If the side length of H is 2^L, then the nodes of T at depth δ induce a $2^\delta \times 2^\delta$ uniform grid inside H; each grid cell has side length $2^{L-\delta}$.

Let $\mathcal{B} = \{B_1, \cdots, B_m\}$ be a set of m canonical squares inside H. We construct a *compressed quadtree* $\mathbb{T}$ on $(\mathcal{B}, H)$ as follows: Let T be the quadtree on H as described above. A square $B \in \mathcal{B}$ is stored at a node v if $\square_v = B$. The leaves of $\mathbb{T}$ are the lowest nodes that store a square of $\mathcal{B}$. They induce a subdivision of H into canonical squares, none of them contains any square of $\mathcal{B}$ in its interior. If a node $v \in T$ does not store a square of $\mathcal{B}$ and both v and $p(v)$, the parent of v, have degree one, we delete v and the child of v becomes the child of $p(v)$. We repeat this step until no such node is left. The size of $\mathbb{T}$ is $O(m)$, and it can be constructed directly, without constructing T, in $O(m \log m)$ time [18].

We call a node v of $\mathbb{T}$ *exposed* if its degree is at most one. We associate a region R_v with each exposed node v. If v is a leaf, then $R_v = \square_v$. Otherwise, v has one child w and we set $R_v = \square_v \setminus \square_w$. For a point $x \in R_v$, v is the lowest node such that $x \in \square_v$. The regions R_v of the exposed nodes induce a partition $\mathbb{M}(\mathcal{B}, H)$ of H of size $O(m)$. Each face of $\mathbb{M}(\mathcal{B}, H)$ is a canonical square or the difference between two canonical squares, and none of the faces contains a square of $\mathcal{B}$ in its interior. The depth of $\mathbb{T}$ is $\Theta(m)$ in the worst case. Nevertheless, using standard tree-decomposition schemes, for a point $x \in H$, the lowest node of $\mathbb{T}$ such that $x \in \square_v$ can be computed in $O(\log m)$ time [18].

$\mathbb{T}$ can also be used to store a set $S = \{p_1, \cdots, p_m\}$ of points in H. We again build a quadtree T on H. A node $v \in T$ is a leaf if $|S \cap \square_v| \le 1$. We compress the nodes as above and define the partition $\mathbb{M}(S, H)$ as earlier. Again, using tree-decomposition schemes, we can now determine in $O(\log m)$ time whether $\sigma \cap S \neq \emptyset$ for a canonical square σ. If the answer is yes, we can also return a point of $S \cap \sigma$. We thus have the following:

Lemma 2.1. *Let H be a canonical square, let $\mathcal{B}$ be a set of m canonical squares in H, and let S be a set of n points in H.*

(i) A compressed quadtree $\mathbb{T}$ on $(\mathcal{B}, H)$ of size $O(m)$ can be constructed in $O(m \log m)$ time. Furthermore, it can be processed in $O(m \log m)$ time into a linear-size index, so that for a point $q \in H$, the lowest node v of $\mathbb{T}$ such that $q \in \square_v$ can be reported in $O(\log m)$ time.

(ii) A compressed quadtree $\mathbb{T}$ on (S, H) of size $O(n)$ can be constructed in $O(n \log n)$ time. Furthermore, it can be processed in $O(n \log n)$ time into a linear-size index, so that for a canonical square σ, a point of $S \cap \sigma$ can be returned in $O(\log n)$ time if $S \cap \sigma \neq \emptyset$.

3. SQUARED EUCLIDEAN DISTANCE

In this section, for two points $a, b \in \mathbb{R}^d$, $\mathrm{d}(a, b) = \|a - b\|^2$. We first show how to compute the EVD of a set of uncertain points, and then show how to answer an ENN query with an uncertain point.

3.1 Uncertain data

Let $\mathcal{P} = \{P_1, \ldots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$. The following lemma, well known in mathematics, suggests how to replace $\mathcal{P}$ with a set of weighted points. We state the lemma in $\mathbb{R}^d$ and provide a proof for the sake of completeness.

Lemma 3.1. *Let P be an uncertain point in $\mathbb{R}^d$, let f be its pdf, let $\overline{p}$ be its centroid, and let $\sigma^2 = \int_{\mathbb{R}^d} \|x - \overline{p}\|^2 f(x) dx$. Then for any point $q \in \mathbb{R}^d$,*

$$\mathsf{Ed}(P, q) = \|q - \overline{p}\|^2 + \sigma^2.$$

Proof. Let $\langle p, q \rangle$ denote the inner product of p and q, and $\| \cdot \|$ denote the Euclidean metric. Using the fact that $\int_{\mathbb{R}^d} f(x) dx = 1$, we obtain

$$\begin{aligned}
\mathsf{Ed}(P, q) = &\int_{\mathbb{R}^d} ||q - x||^2 f(x) dx \\
&= ||q||^2 - 2\langle q, \int_{\mathbb{R}^d} x f(x) dx \rangle + \int_{\mathbb{R}^d} ||x||^2 f(x) dx \\
&= ||q - \overline{p}||^2 - ||\overline{p}||^2 + \int_{\mathbb{R}^d} ||x||^2 f(x) dx \\
&= ||q - \overline{p}||^2 - 2||\overline{p}||^2 \\
&\quad + \int_{\mathbb{R}^d} \Big(||x - \overline{p}||^2 + 2\langle x, \overline{p} \rangle \Big) f(x) dx \\
&= ||q - \overline{p}||^2 - 2||\overline{p}||^2 + \sigma^2 + 2\langle \overline{p}, \overline{p} \rangle \\
&= ||q - \overline{p}||^2 + \sigma^2.
\end{aligned}$$

□

Let p be a weighted point in $\mathbb{R}^d$ with weight w_p. For a point $q \in \mathbb{R}^d$, we define the (weighted) distance from q to p as

$$\delta(q, p) = \|q - p\|^2 + w_p.$$

If we replace each point in $P_i \in \mathcal{P}$ by a weighted point $\overline{p}_i$ whose weight is $\sigma_i^2 = \int_{\mathbb{R}^d} \|x - \overline{p}_i\|^2 f_i(x) dx$, then by the above lemma $\delta(q, \overline{p}_i) = \mathsf{Ed}(P_i, q)$. Set $\overline{\mathcal{P}} = \{\overline{p}_1, \ldots, \overline{p}_n\}$. EVD($\mathcal{P}$) is the same as the Voronoi diagram of $\overline{\mathcal{P}}$ under the distance function $\delta(\cdot, \cdot)$. We now show how to compute the Voronoi diagram of $\overline{\mathcal{P}}$.

For each $1 \le i \le n$, we define a linear function $h_i : \mathbb{R}^2 \to \mathbb{R}$ as

$$h_i(x) = 2\langle \overline{p}_i, x \rangle - ||\overline{p}_i||^2 - \sigma_i^2.$$

The proof of the following lemma is straightforward.

Lemma 3.2. *For any $q \in \mathbb{R}^2$,*

$$\arg \min_{1 \le i \le n} \delta_i(q, \overline{p}_i) = \arg \max_{1 \le i \le n} h_i(q).$$

Let $h_i^+ = \{x \in \mathbb{R}^3 \mid h_i(x) \ge 0\}$ be the halfspace lying above the plane h_i. Set $H^+ = \{h_i^+ \mid 1 \le i \le n\}$. By Lemma 3.2, the minimization diagram of functions $\{\delta(x, \overline{p}_i)\}$ is the same as the xy-projection of $\bigcap_{h^+ \in H^+} h^+$. Since the intersection of n halfspaces in $\mathbb{R}^3$ has linear size and can be computed in $O(n \log n)$ time [16], we conclude that the Voronoi diagram of $\overline{\mathcal{P}}$, under $\delta(\cdot, \cdot)$ as the distance function, can be computed in $O(n \log n)$ time, and thus EVD($\mathcal{P}$) can be computed in $O(n \log n + nk)$ time, where the extra $O(nk)$ time is for computing $\overline{\mathcal{P}}$. Furthermore, by preprocessing EVD($\mathcal{P}$) into a linear-size index for point-location queries, an ENN query for a point $q \in \mathbb{R}^2$ can be answered in $O(\log n)$ time. We thus obtain the following.

Theorem 3.3. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$.* EVD($\mathcal{P}$) *under the squared Euclidean distance has $O(n)$ size. If the description complexity of the pdf of every point in $\mathcal{P}$ is k, then* EVD($\mathcal{P}$) *can be computed in $O(n \log n + nk)$ time. Furthermore,* EVD($\mathcal{P}$) *can be preprocessed in a linear-size index so that for a point $q \in \mathbb{R}^2$, an* ENN *query can be answered in $O(\log n)$ time.*

3.2 Uncertain query

Let Q be an uncertain query point in $\mathbb{R}^2$ represented as a pdf f_Q.

Lemma 3.4. *For an uncertain point Q with a pdf f_Q,*

$$\varphi(\mathcal{P}, Q) = \arg\min_{p \in \mathcal{P}} ||\bar{q} - p||^2,$$

where $\bar{q}$ is the centroid of Q.

Proof. For a point $p \in \mathcal{P}$, we have

$$\begin{aligned}\mathsf{Ed}(p, Q) &= \int_{\mathbb{R}^d} ||p - x||^2 f_Q(x) dx \\ &= ||p||^2 + \int_{\mathbb{R}^d} ||x||^2 f_Q(x) dx - 2\langle p, \bar{q}\rangle. \qquad (1)\end{aligned}$$

Observing that the second term in RHS of (1) is independent of p, we obtain

$$\begin{aligned}\arg\min_{p \in \mathcal{P}} \mathsf{Ed}(p, Q) &= \arg\min_{p \in \mathcal{P}} ||p||^2 - 2\langle p, \bar{q}\rangle \\ &= \arg\min_{p \in \mathcal{P}} ||p - \bar{q}||^2,\end{aligned}$$

as claimed. □

The preprocessing step is to compute the Voronoi diagram $\mathrm{VD}(\mathcal{P})$ of the points $\mathcal{P}$ in time $O(n \log n)$. Once a query Q with a pdf of description complexity k is given, we compute its centroid $\bar{q}$ in $O(k)$ time and find the nearest neighbor $\mathrm{NN}(\mathcal{P}, \bar{q}) = \arg\min_{p \in \mathcal{P}} ||\bar{q} - p||^2$ in $O(\log n)$ time by querying $\mathrm{VD}(\mathcal{P})$ with q.

Theorem 3.5. *Let $\mathcal{P}$ be a set of n exact points in $\mathbb{R}^2$. We can preprocess $\mathcal{P}$ into an index of size $O(n)$ in $O(n \log n)$ time so that, for a query point Q with a pdf of description complexity k, $\varphi(\mathcal{P}, Q)$, under the squared Euclidean distance, can be computed in $O(k + \log n)$ time.*

Remarks. The algorithm extends to higher dimensions but the query time becomes roughly $O(n^{1-1/\lceil d/2 \rceil})$.

4. RECTILINEAR METRIC

In this section we assume the distance function to be the L_1 metric. That is, for any two points $p = (p_x, p_y)$ and $q = (q_x, q_y)$,

$$\mathrm{d}(p, q) = |p_x - q_x| + |p_y - q_y|.$$

The results in this section also hold for the L_∞ metric, i.e., when $\mathrm{d}(p, q) = \max\{|p_x - q_x|, |p_y - q_y|\}$. We first consider the case when the input is a set of n uncertain points in $\mathbb{R}^2$, each with a discrete pdf, and the query is a certain point and then consider the case when the input is a set of certain points and the query is a uncertain point with a discrete pdf.

4.1 Uncertain data

Let $\mathcal{P} = \{P_1, \cdots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a discrete pdf of size k as described above. We first prove a lower bound on the complexity of $\mathrm{EVD}(\mathcal{P})$ and then present a near-linear-size index to answer ENN queries.

Expected Voronoi diagram. Fix a point $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\}$ of $\mathcal{P}$. Let H_i^{-} (resp. $H_i^{|}$) be the set of k horizontal (resp. vertical) lines in $\mathbb{R}^2$ passing through the points of P_i. Let $\mathcal{B}_i$ be the set of $O(k^2)$ rectangles in the grid induced by the lines in $H_i^{-} \cup H_i^{|}$. It can be checked that $\mathsf{Ed}(P_i, q)$ is a linear function $f_\square$ within each rectangle $\square$ of $\mathcal{B}_i$; see Figure 3. For each $\square \in \mathcal{B}_i$, let $\square^{\uparrow}$ be the rectangle in $\mathbb{R}^3$ formed by restricting the graph of $f_\square$ with $\square$, i.e.,

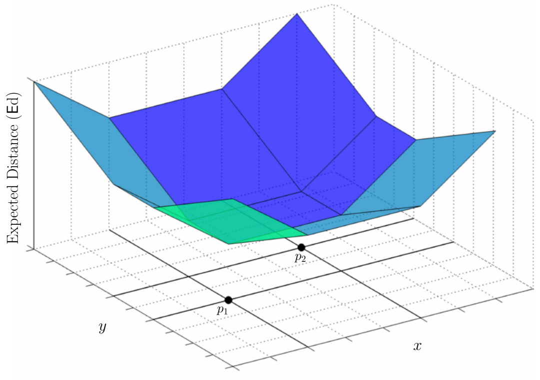


Figure 3. $\mathsf{Ed}(P_i, q)$ when the uncertain point P_i is composed of two points p_1 and p_2 with probabilities 0.5 each. The grid induced by H_i^{-} and $H_i^{|}$ is shown below and $\mathsf{Ed}(P_i, q)$ is linear within each rectangle of B_i.

$$\square^{\uparrow} = \{(x, y, f_\square(x, y)) \mid (x, y) \in \square\}.$$

Let $\mathcal{B}_i^{\uparrow} = \{\square^{\uparrow} \mid \square \in \mathcal{B}_i\}$. By definition, the rectangles in $\mathcal{B}_i^{\uparrow}$ form the graph of the function $\mathsf{Ed}(P_i, q)$. Set $\mathcal{B} = \bigcup_{i=1}^{n} \mathcal{B}_i$ and $\mathcal{B}^{\uparrow} = \bigcup_{i=1}^{n} \mathcal{B}_i^{\uparrow}$. By construction and the discussion in Section 2, $\mathrm{EVD}(\mathcal{P})$ is the minimization diagram $\mathbb{M}(\mathcal{B}^{\uparrow})$ of $\mathcal{B}^{\uparrow}$. We prove an almost tight bound on the complexity of $\mathrm{EVD}(\mathcal{P})$.

Theorem 4.1. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a discrete pdf consisting of k points, and let $\mathrm{d}(\cdot,\cdot)$ be the L_1 metric. Then the complexity of $\mathrm{EVD}(\mathcal{P})$ is $O(k^2 n^2 \alpha(n))$, where $\alpha(n)$ is the inverse Ackermann function. Moreover, there is a set $\mathcal{P}$ of n uncertain points with $k = 2$ such that $\mathrm{EVD}(\mathcal{P})$ has $\Omega(n^2)$ size.*

Proof. We first prove the upper bound. Set $H^{-} = \bigcup_{i=1}^{n} H_i^{-}$ and $H^{|} = \bigcup_{i=1}^{n} H_i^{|}$; $|H^{-}| = |H^{|}| = nk$. We sort the lines in H^{-} by their y values and choose a subset G^{-} of n lines by selecting every kth line. Let $G^{|}$ be a similar subset of $H^{|}$. Let $\mathcal{R}$ be the set of rectangles in the non-uniform grid formed by the lines in $G^{-} \cup G^{|}$. For each rectangle $R \in \mathcal{R}$, let $\mathcal{B}_R \subseteq \mathcal{B}$ be the set of rectangles whose boundaries intersect R, and let $\overline{\mathcal{B}}_R \subseteq \mathcal{B}$ be the set of rectangles that contain R. Since R lies between two adjacent lines of G^{-} and $G^{|}$, at most $2n$ lines of $H^{-} \cup H^{|}$ intersect R, implying that $|\mathcal{B}_R| \le 2n$. $|\overline{\mathcal{B}}_R| \le n$ because at most one rectangle of $\mathcal{B}_i$ can contain R for any $1 \le i \le n$. Set $\mathcal{B}_R^{\uparrow} = \{\square^{\uparrow} \mid \square \in \mathcal{B}_R\}$ and $\overline{\mathcal{B}}_R^{\uparrow} = \{\square^{\uparrow} \mid \square \in \overline{\mathcal{B}}_R\}$. We note that $\mathbb{M}(\mathcal{B}^{\uparrow}) \cap R = \mathbb{M}(\mathcal{B}_R^{\uparrow} \cup \overline{\mathcal{B}}_R^{\uparrow}) \cap R$. Since $|\mathcal{B}_R| + |\overline{\mathcal{B}}_R| \le 3n$, the complexity of $\mathbb{M}(\mathcal{B}_R^{\uparrow} \cup \overline{\mathcal{B}}_R^{\uparrow})$ is $O(n^2 \alpha(n))$. The $O(k^2)$ rectangles in $\mathcal{R}$ tile the entire plane, therefore the complexity of $\mathbb{M}(\mathcal{B}^{\uparrow})$, and thus of $\mathrm{EVD}(\mathcal{P})$, is $O(k^2 n^2 \alpha(n))$.

Next, we show that there exists a set $\mathcal{P}$ of n uncertain points in $\mathbb{R}^2$ with $k = 2$ such that $\mathrm{EVD}(\mathcal{P})$ has $\Omega(n^2)$ size. Assume that $n = 2m$ for some positive integer m. Each point $P_i \in \mathcal{P}$ has two possible locations p_{i1} and p_{i2}, each with

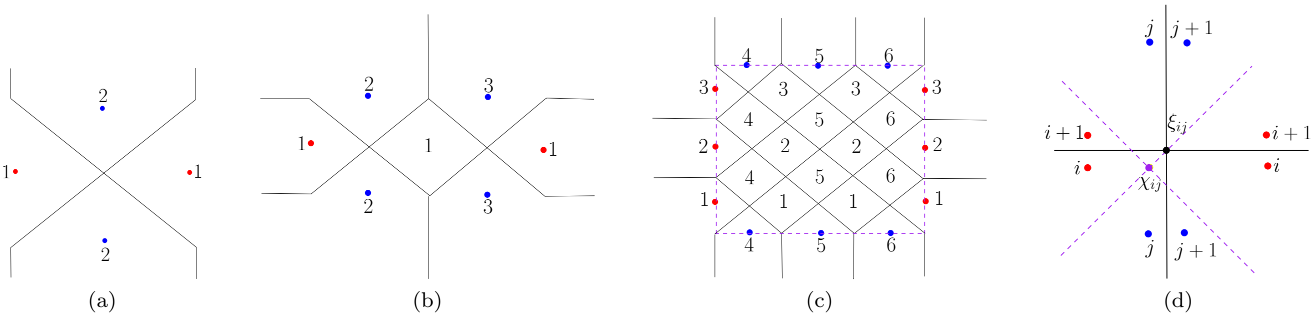


Figure 2. Lower bound construction for EVD. EVD of (a) 2 points, (b) 3 points, (c) 6 points on the boundary of a square σ. (d) Bisectors of (P_i, P_j), (P_i, P_{i+1}), and (P_j, P_{j+1}).

probability 0.5. All the points lie on the boundary of the square $\sigma = [0, 2m]^2$ (see Figure 2). More specifically, for $1 \le i \le m$, the two possible locations of P_i are $p_{i1} = (0, 2i-1)$ and $p_{i2} = (2m, 2i-1)$. For $1 \le j \le m$, the two possible locations of P_{m+j} are $p_{m+j,1} = (2j-1, 0)$ and $p_{m+j,2} = (2j-1, 2m)$. We claim that EVD($\mathcal{P}$) has $\Omega(n^2)$ size. Notice that for any pair $1 \le i \le m < j \le 2m$, the bisector of P_i and P_j inside the square σ consists of two lines: $y = x+2(m+i-j)$ and $y = -x+2(i+j-m-1)$ (see Figure 2(d), dashed lines). Let χ_{ij} be the intersection point of these two lines. We observe that $\mathsf{Ed}(P_i, \chi_{ij}) = \mathsf{Ed}(P_j, \chi_{ij}) = m$ and $\mathsf{Ed}(P_k, \chi_{ij}) > m$ for all $k \notin \{i, j\}$. Hence χ_{ij} is a vertex of EVD($\mathcal{P}$). Similarly, for all $1 \le i < m$, the bisector of P_i and P_{i+1} is the line $y = 2i$, and for all $m < j < 2m$, the bisector of P_j and P_{j+1} is the line $x = 2j - 2m$ (see Figure 2(d), solid lines). The intersection point of these two bisectors, $\xi_{ij} = (2j-2m, 2i)$, is also a vertex of EVD($\mathcal{P}$): $\mathsf{Ed}(P_i, \xi_{ij}) = \mathsf{Ed}(P_{i+1}, \xi_{ij}) = \mathsf{Ed}(P_j, \xi_{ij}) = \mathsf{Ed}(P_{j+1}, \xi_{ij}) = m + 0.5 < \mathsf{Ed}(P_k, \xi_{ij})$, for all $k \notin \{i, i+1, j, j+1\}$. Hence EVD($\mathcal{P}$) has $\Omega(n^2)$ vertices inside σ.

□

Remark. By preprocessing EVD($\mathcal{P}$) for point-location queries [32, 16], an ENN query can be answered in $O(\log n)$ time using $O(k^2 n^2 \alpha(n))$ space. For higher dimensions, the complexity of EVD($\mathcal{P}$) is $O(k^d n^d \alpha(n))$.

Near-linear size index. Next we show that despite the size of EVD($\mathcal{P}$) being $\Omega(n^2)$, we can build an index of size $O(k^2 n \log^2 n)$ so that an ENN query can be answered in $O(\log^3(kn))$ time.

For a query point $q \in \mathbb{R}^2$, let l_q be the line parallel to the z-axis passing through q, and oriented in the $(+z)$-direction. Then $\varphi(\mathcal{P}, q)$ is P_i if the first rectangle of $\mathcal{B}^\uparrow$ that l_q intersects belongs to $\mathcal{B}_i$. We label each rectangle $\square^\uparrow$ in $\mathcal{B}_i^\uparrow$ with i and build an index on $\mathcal{B}^\uparrow$ so that the first rectangle intersected by a line parallel to the z-axis can be reported quickly. The index works in two stages. In the first stage, it builds a family $\mathcal{F} = \{C_1, C_2, \cdots, C_u\}$ of *canonical subsets* of $\mathcal{B}$, i.e., each $C_i \subseteq \mathcal{B}$, so that for a query point $q \in \mathbb{R}^2$, the subset $\mathcal{B}_q \subseteq \mathcal{B}$ of rectangles containing q can be represented as the union of $O(\log^2(kn))$ canonical subsets of $\mathcal{F}$. That is, there exists a subset $\mathcal{F}_q \subseteq \mathcal{F}$ of size $O(\log^2 n)$ such that $\mathcal{B}_q = \bigcup \mathcal{F}_q$. Furthermore, $\sum_{i \ge 1} |C_i| = O(k^2 n \log^2 n)$. Next, for a rectangle $\square \in \mathcal{B}$, let $\gamma_\square$ be the plane containing the rectangle $\square^\uparrow$, i.e., the graph of the linear function $f_\square$. For each $1 \le i \le u$, set $T_i = \{\gamma_\square \mid \square \in C_i\}$. We build an index of linear size on T_i that can report the first plane of T_i intersected by a vertical line l_q in $O(\log n)$ time [16]. This index is similar to one described by Agarwal *et al.* [1], so we omit the details from here and conclude the following.

Theorem 4.2. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a discrete pdf consisting of k points, and let $\mathrm{d}(\cdot,\cdot)$ be the L_1 or L_∞ metric. $\mathcal{P}$ can be stored in an index of size $O(k^2 n \log^2 n)$, so that an* ENN *query can be answered in $O(\log^3(kn))$ time. The index can be built in $O(k^2 n \log^3 n)$ time.*

Remarks. The algorithm extends to higher dimension. For $d \ge 3$, the size of the index will be $O(k^d n \log^d n)$ and the query time will be $O(n^{1-1/\lceil d/2 \rceil} \log^{O(d)} n)$ [3].

4.2 Uncertain query

Let $\mathcal{P} = \{p_1, p_2 \ldots, p_n\}$ be a set of n certain input data points in $\mathbb{R}^2$. We first build an index such that the ENN of an uncertain query point Q, which is represented as a discrete pdf of k points, can be found quickly.

Given an uncertain query Q, which has a discrete pdf of k points $\{q_1, \ldots, q_k\}$ with associated probabilities $\{w_1, \ldots, w_k\}$, let H^- (resp. $H^|$) be the set of k horizontal (resp. vertical) lines in $\mathbb{R}^2$ passing through the points of Q. Let $\mathcal{B}$ be the set of $O(k^2)$ rectangles in the grid induced by the lines in $H^- \cup H^|$ (see Figure 4(a)). For every rectangle $\square \in \mathcal{B}$, let $\mathcal{P}_\square = \mathcal{P} \cap \square$. Let k_{nw} denote the number of points of Q which are above and to the right of $\square$. We similarly define $k_{\mathrm{ne}}, k_{\mathrm{sw}}, k_{\mathrm{se}}$ for points of Q which are at top-right, bottom-left and bottom-right of $\square$. We call these regions the quadrants of $\square$; see Figure 4(b).

Lemma 4.3. *For every point $p = (x_p, y_p) \in \square$,*

$$\mathsf{Ed}(p, Q) = k_x x_p + k_y y_p + c,$$

where $k_x = k_{\mathrm{nw}} + k_{\mathrm{sw}} - k_{\mathrm{ne}} - k_{\mathrm{se}}$, $k_y = k_{\mathrm{sw}} + k_{\mathrm{se}} - k_{\mathrm{nw}} - k_{\mathrm{ne}}$ and c is independent of p.

Proof. Note that no point of Q lies vertically above or below, or horizontally to the left or right of $\square$. Let $v_{\mathrm{nw}} = (x_{\mathrm{nw}}, y_{\mathrm{nw}})$ (resp. $v_{\mathrm{ne}}, v_{\mathrm{sw}}, v_{\mathrm{se}}$) denote the top-left (resp. top-right, bottom-left, bottom-right) corner of $\square$. Let Q_{nw}, Q_{ne}, Q_{sw} and Q_{se} denote the points of Q that lie in the top-left, top-right, bottom-left and bottom-right quadrants of $\square$

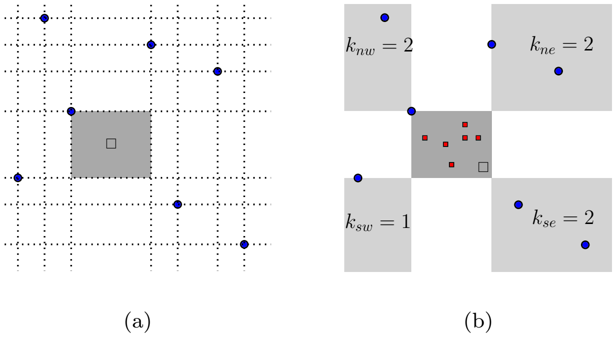


Figure 4. (a) The set of rectangles $\mathcal{B}$ induced by horizontal lines H^- and vertical lines $H^|$ through the points of Q. A single rectangle $\Box$ is also shown. (b) The four quadrants of $\Box$ are shown along with the number of points of Q in each. The points $\mathcal{P}_\Box$ in $\Box$ are shown as red squares.

respectively, and let $q \in Q_{\mathrm{nw}}$. Then $\mathrm{d}(p,q) = \mathrm{d}(p, v_{\mathrm{nw}}) + \mathrm{d}(v_{\mathrm{nw}}, q)$. Thus the total contributions of points of Q in this quadrant to $\mathsf{Ed}(p,Q)$ is

$$\begin{aligned} &\sum_{q \in Q_{\mathrm{nw}}} \mathrm{d}(q, v_{\mathrm{nw}}) + k_{\mathrm{nw}} \mathrm{d}(v_{\mathrm{nw}}, q) \\ = &\sum_{q \in Q_{\mathrm{nw}}} \mathrm{d}(q, v_{\mathrm{nw}}) + k_{\mathrm{nw}}(x_p - x_{\mathrm{nw}}) + k_{\mathrm{nw}}(y_{\mathrm{nw}} - y_p). \end{aligned}$$

Similar expressions holds for the remaining quadrants. Thus, summing over all quadrants, the lemma follows. □

LEMMA 4.4. *The point p^* such that*

$$p^* = \arg\min_{p \in P_\Box} \mathsf{Ed}(p, Q)$$

is a vertex of the convex hull $\mathrm{conv}(\mathcal{P}_\Box)$ *of* $\mathcal{P}_\Box$*.*

PROOF. By Lemma 4.3, p^* is an extreme point of $\mathcal{P}_\Box$ minimizing a linear function of $p \in \mathcal{P}_\Box$. Thus, without loss of generality, it realizes its minimum when p^* is a vertex of $\mathrm{conv}(\mathcal{P}_\Box)$. □

Preprocessing step. Our index is simply a two dimensional range-tree on the points in $\mathcal{P}$ [16] with a single modification to enable efficient ENN queries. The range-tree consists of two levels. We first construct a balanced binary tree $\mathcal{T}$ on the x-coordinates of the points of $\mathcal{P}$. We call this the *primary tree.* Its leaves store the points of $\mathcal{P}$ in sorted x-order from left to right, and internal nodes store splitting values to guide the search. Each node v in this tree corresponds to a subset of the points of $\mathcal{P}$ whose x-coordinates lie in the interval corresponding to the node. For each node v in the tree, a similar balanced binary tree $\mathcal{T}_v$ is constructed on the y-coordinates of the points of $\mathcal{P}$ in the subtree of $\mathcal{T}$ rooted at v. We call these *secondary trees.* For a node u in a secondary tree $\mathcal{T}_v$ corresponding to a node v in $\mathcal{T}$, we have an associated subset of the points of $\mathcal{P}$. All such subsets are termed as *canonical subsets.* Given a query rectangle, the points of $\mathcal{P}$ in the rectangle are reported as the disjoint union of $O(\log^2 n)$ canonical subsets. See [16] for more details on range-trees.

We make the following modification to the range-tree structure. For any canonical subset $\mathcal{P}_u$ corresponding to a node u in a secondary tree $\mathcal{T}_v$, we store the convex hull $\mathrm{conv}(\mathcal{P}_u)$ of the points of $\mathcal{P}_u$. For any secondary tree $\mathcal{T}_v$, the convex hull of the canonical subsets may be computed by performing a bottom-up traversal while merging the convex hulls of the children at any internal node. Thus, if there are m nodes in $\mathcal{T}_v$, the total time for constructing the convex hulls is $O(m \log m)$. This, in turn, implies that the total preprocessing time is $O(n \log^2 n)$ and the space required for the index is $O(n \log^2 n)$ as well.

Query step. When a query Q is given, we construct $\mathcal{B}$ as above, and compute, for each rectangle $\Box \in \mathcal{B}$, the values $k_{\mathrm{ne}}, k_{\mathrm{nw}}, k_{\mathrm{se}}$ and k_{sw}. Next, we perform a range query in $\mathcal{T}$, to find the points of $\mathcal{P}_\Box$ as the union of $O(\log^2 n)$ canonical subsets of $\mathcal{P}$. We find the point p^* by performing a binary search on the points on the convex hulls of each subset and picking the minimum over all subsets. By Lemma 4.4, the point p^* must be among these points. Hence, the total time is $O(k^2 \log^3 n)$.

THEOREM 4.5. *Let $\mathcal{P}$ be a set of n exact points in $\mathbb{R}^2$ and let $\mathrm{d}(\cdot,\cdot)$ be the L_1 or L_∞ metric. $\mathcal{P}$ can be stored in an index of size $O(n \log^2 n)$, which can be constructed in $O(n \log^2 n)$ time, such that for an uncertain query Q as a discrete pdf with k points, its* ENN *can be reported in $O(k^2 \log^3 n)$ time.*

Remark. If we know an upper bound on k in advance of the query, we may perform further preprocessing to obtain a query time of $O(k^2 \log^2 n \log k)$ since the linear function from Lemma 4.3 can have at most $O(k^2)$ orientations, corresponding to the possible coefficients of x_p and y_p. Thus, we may find the minimum of this function for all the $O(k^2)$ possible orientations in advance.

5. EUCLIDEAN DISTANCE

We now consider the case when $\mathrm{d}(\cdot,\cdot)$ is the Euclidean distance. For any two points $a, b \in \mathbb{R}^2$, we use $||a - b||$ to denote the Euclidean distance $\mathrm{d}(a,b)$. Since the expected distance under the Euclidean distance is algebraically quite complex, we focus on answering ε-ENN queries in this section. We first describe an algorithm for computing a function that approximates the expected distance from a fixed uncertain point to any (certain) point in $\mathbb{R}^2$. The construction is similar to the one given in [2]. We use this algorithm to answer ε-ENN queries, first when the input is a set of uncertain points but the query is a certain point and next, when the input data points are certain but the query point is uncertain. In the former case, we construct an approximate expected Voronoi diagram of $\mathcal{P}$.

5.1 Approximation of the expected Euclidean distance

Let P be an uncertain point in $\mathbb{R}^2$, and let $f_P : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ be its pdf. Let the description complexity of f_P be k. We construct a function $g_P : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ of description complexity $O((1/\varepsilon^2)\log(1/\varepsilon))$ such that for any $x \in \mathbb{R}^2$,

$$\mathsf{Ed}(P,x) \leq g_P(x) \leq (1+\varepsilon)\mathsf{Ed}(P,x).$$

Let $\overline{p}$ be the centroid of P. The following two lemmas follow from the triangle inequality.

LEMMA 5.1. *For any two points $a, b \in \mathbb{R}^2$,*

$$|\mathsf{Ed}(P,a) - \mathsf{Ed}(P,b)| \leq ||a - b||.$$

PROOF.

$$\begin{aligned}|\mathsf{Ed}(P,a)-\mathsf{Ed}(P,b)| =&\int_{\mathbb{R}^2} f_P(x)|\|x-a\|-\|x-b\||dx\\ \leq&\int_{\mathbb{R}^2} f_P(x)\|a-b\|dx\\ =&\|a-b\|.\end{aligned}$$

□

LEMMA 5.2. $\mathsf{Ed}(P,\overline{p}) \leq 2\min_{x\in\mathbb{R}^2}\mathsf{Ed}(P,x)$.

PROOF. Let $p_{\min} = \arg\min_{x\in\mathbb{R}^2}\mathsf{Ed}(P,x)$. By Lemma 5.1,

$$\begin{aligned}|\mathsf{Ed}(P,\overline{p})-\mathsf{Ed}(P,p_{\min})| &\leq \|\overline{p}-p_{\min}\|\\ &= ||p_{\min}-\int_{\mathbb{R}^2} xf_P(x)dx||\\ &= ||\int_{\mathbb{R}^2} f_P(x)(x-p_{\min})dx||\\ &\leq \int_{\mathbb{R}^2} f_P(x)||x-p_{\min}||dx\\ &= \mathsf{Ed}(P,p_{\min}).\end{aligned}$$

The lemma now follows. □

LEMMA 5.3. *Let $0<\varepsilon<1$ be a parameter, let $\bar{\rho} = \mathsf{Ed}(P,\bar{p})$, and for any $x\in\mathbb{R}^2$, let*

$$g(x) = \|x-\bar{p}\| + \bar{\rho}.$$

Then for any point $q\in\mathbb{R}^2$ such that $||q-\overline{p}|| > 8\bar{\rho}/\varepsilon$, we have

$$\mathsf{Ed}(P,q) \leq g(q) \leq (1+\varepsilon)\mathsf{Ed}(P,q).$$

PROOF. Let $q\in\mathbb{R}^2$ be a point with $||q-\overline{p}|| > 8\bar{\rho}/\varepsilon$. By Lemma 5.1,

$$\mathsf{Ed}(P,q) \leq \mathsf{Ed}(P,\bar{p}) + \|q-\overline{p}\| = \bar{\rho} + \|q-\overline{p}\| \leq g(q).$$

Similarly,

$$\mathsf{Ed}(P,q) \geq \|q-\overline{p}\| - \mathsf{Ed}(P,\overline{p}) = \|q-\overline{p}\| - \bar{\rho}.$$

Therefore

$$g(q) = \|q-\overline{p}\| + \bar{\rho} \leq \mathsf{Ed}(P,q) + 2\bar{\rho}. \tag{2}$$

Let D be the disk of radius $4\bar{\rho}/\varepsilon$ centered at $\bar{p}$. For any point $x\notin D$, $\|x-\overline{p}\| > 4\bar{\rho}/\varepsilon$. Hence,

$$\begin{aligned}\bar{\rho} = &\int_{\mathbb{R}^2}\|x-\bar{p}\|f_P(x)dx \geq \int_{\mathbb{R}^2\setminus D}\|x-\overline{p}\|f_P(x)dx\\ \geq &\frac{4\bar{\rho}}{\varepsilon}\int_{\mathbb{R}^2\setminus D} f_P(x)dx,\end{aligned}$$

implying that

$$\int_{\mathbb{R}^2\setminus D} f_P(x)dx \leq \varepsilon/4 \quad\text{and}\quad \int_D f_P(x)dx \geq 1-\varepsilon/4.$$

On the other hand, for any point $x\in D$, $\|x-q\| > 4\bar{\rho}/\varepsilon$. Therefore

$$\begin{aligned}\mathsf{Ed}(P,q) = &\int_{\mathbb{R}^2}\|x-q\|f_P(x)dx \geq \int_D \|x-q\|f_P(x)dx\\ \geq &\frac{4\bar{\rho}}{\varepsilon}\int_D f_P(x)dx \geq \frac{4\bar{\rho}}{\varepsilon}(1-\varepsilon/4) \geq \frac{2\bar{\rho}}{\varepsilon},\end{aligned}$$

which implies that $\bar{\rho} \leq \varepsilon\mathsf{Ed}(P,q)/2$. Substituting this in (2),

$$g(q) \leq (1+\varepsilon)\mathsf{Ed}(P,q).$$

□

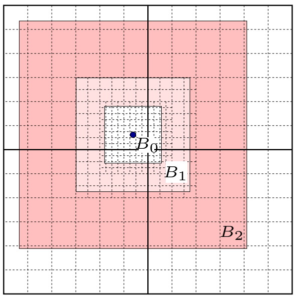


Figure 5. Covering B_l with four canonical squares and drawing an exponential grid composed of canonical squares.

Set $\overline{\rho} = \mathsf{Ed}(P,\overline{p})$. For a point $x\in\mathbb{R}^2$ and a value $r\geq 0$, let $B(x,r)$ denote the square of side length $2r$ centered at x. Let $l = \lceil\log_2(8/\varepsilon)\rceil$. For $0\leq i\leq l$, set $B_i = B(\overline{p},\overline{\rho}2^i)$; set $B_{-1}=\emptyset$. Finally, set $\rho_i = \varepsilon 2^i\overline{\rho}/8$.

We cover B_l by at most four congruent canonical squares $C_1,\ldots,C_4$ of side length at most $2\rho_l = \overline{\rho}2^{l+1}$. The union of $C_1,\ldots,C_4$ is also a square C; see Figure 5. We set

$$g_P(x) = \|x-\overline{p}\| + \overline{\rho}, \quad \forall x\notin C.$$

For $1\leq j\leq 4$ and $0\leq i\leq l$, we cover $C_j\cap(B_i\setminus B_{i-1})$ with canonical squares of size 2^{Δ_i} where $\Delta_i = \lfloor\log_2\rho_i\rfloor$; see Figure 5. For each such square $\square$, let $a_\square$ be its center and set

$$\delta_\square = \mathsf{Ed}(P,a_\square) + 4\cdot 2^{\Delta_i}.$$

Finally, we also cover $C\setminus B_l$ with canonical squares of size 2^{Δ_l} and set $\delta_\square$ as above. Let $\mathcal{B}$ be the resulting set of $O((1/\varepsilon^2)\log(1/\varepsilon))$ canonical squares. We construct a compressed quadtree $\mathcal{T}$ on $(\mathcal{B},C)$ as described in Section 2. It can be checked that each exposed node on $\mathcal{T}$ is a leaf and therefore the rectilinear subdividision of C induced by $\mathbb{M} = \mathbb{M}(\mathcal{B},C)$ is a hierarchical grid composed of canonical squares. If a square σ in $\mathbb{M}$ lies in multiple squares of $\mathcal{B}$, we set $\delta_\sigma = \delta_\square$ where $\square$ is the smallest square of $\mathcal{B}$ containing σ. Finally, for every $\sigma\in\mathbb{M}$, we set

$$g_P(x) = \delta_\sigma, \quad \forall x\in\sigma.$$

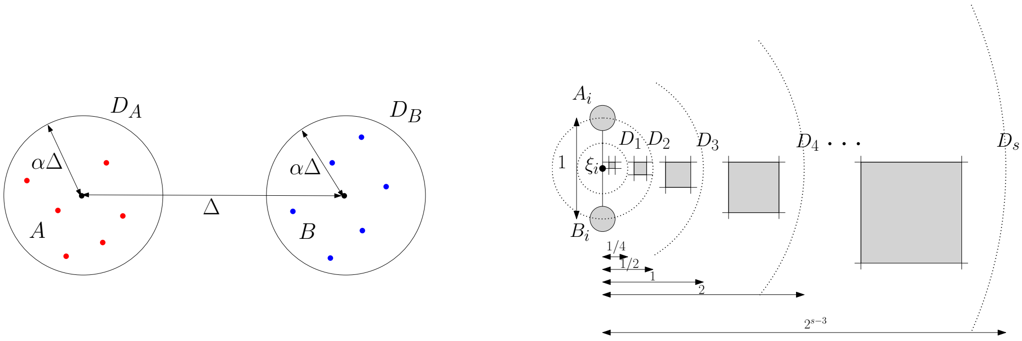
LEMMA 5.4. *Let P be an uncertain point in $\mathbb{R}^2$ with a pdf of description complexity k, and let $0<\varepsilon<1$ be a parameter. A function $g_P:\mathbb{R}^2\to\mathbb{R}_{\geq 0}$ can be constructed in $O((k/\varepsilon^2)\log(1/\varepsilon))$ time such that*

(i) g_P is piecewise constant inside a square C, which is the union of four canonical squares.

(ii) Each piece of g_P is defined over a canonical square, and the number of pieces is $O((1/\varepsilon^2)\log(1/\varepsilon))$.

(iii) $C\supseteq B[\overline{p},8\mathsf{Ed}(P,\overline{p})/\varepsilon]$ and $g_P(x) = ||x-\overline{p}|| + \mathsf{Ed}(P,\overline{p})$ for $x\notin C$.

(iv) $\mathsf{Ed}(P,x)\leq g_P(x)\leq(1+\varepsilon)\mathsf{Ed}(P,x)$ for all $x\in\mathbb{R}^2$.



(a) (b)

Figure 6. (a) A single pair (A, B) in an α-WSPD. (b) The $D_1, \ldots, D_s$ for a suitable $s = O(\log(1/\varepsilon))$ constructed for a pair (A_i, B_i) in a (1/8)-WSPD. D_j has radius 2^{j-3} and is covered by canonical squares of side length γ_j.

Proof. (i) and (ii) follow from the construction, and (iii) follows from Lemma 5.3, so we only need to prove (iv). We describe the proof for the case when $x \in B_0$, a similar argument holds when $x \in B_i \setminus B_{i-1}$, for $i \geq 1$. Suppose x lies in a grid cell τ of B_0. Then, using Lemma 5.1,

$$\begin{aligned} g_P(x) &= \mathsf{Ed}(P, a_\tau) + 4 \cdot 2^{\Delta_0} \\ &\geq \mathsf{Ed}(P, x) - \|x - a_\tau\| + 2\rho_0 \\ &\geq \mathsf{Ed}(P, x). \end{aligned}$$

On the other hand,

$$\begin{aligned} g_P(x) &\leq \mathsf{Ed}(P, x) + \|x - a_\tau\| + 4 \cdot 2^{\Delta_0} \\ &\leq \mathsf{Ed}(P, x) + 2\rho_0 + 4\rho_0 \\ &\leq \mathsf{Ed}(P, x) + \frac{3\varepsilon}{4}\overline{\rho} \\ &\leq (1 + \varepsilon)\mathsf{Ed}(P, x). \end{aligned}$$

□

Remark. We remark that a similar function can be constructed that approximates $\mathsf{Ed}(P, x)$ even when $\mathrm{d}(\cdot, \cdot)$ is any L_p metric.

5.2 Uncertain data

Let $\mathcal{P} = \{P_1, \cdots, P_n\}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a pdf of description complexity k. We describe a method for computing an ε-ENN of a query point $q \in \mathbb{R}^2$ in $\mathcal{P}$. For each $1 \leq i \leq n$, we construct the function $g_i : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$, using Lemma 5.4, such that $g_i(q) \leq (1 + \varepsilon/3)\mathsf{Ed}(P_i, q)$, for all $q \in \mathbb{R}^2$. Let C_i be the canonical square inside which g_i is a piecewise-constant function. Let $G = \{g_1, \ldots, g_n\}$. By definition, the minimization diagram $\mathbb{M}(G)$ of G is an ε-EVD($\mathcal{P}$). Hence, it suffices to construct $\mathbb{M}(G)$ and build an index on $\mathbb{M}(G)$ for point-location queries. The difficulty with this approach is that we do not have a near-linear upper bound on the complexity of $\mathbb{M}(G)$ even in $\mathbb{R}^2$. Moreover, the complexity of $\mathbb{M}(G)$ is $\Omega(n^{\lceil d/2 \rceil})$ in higher dimensions, so this approach will not be practical for $d \geq 3$. We circumvent this problem by using the ideas from Arya *et al.* [6] and constructing a different ε-EVD($\mathcal{P}$) of near-linear size.

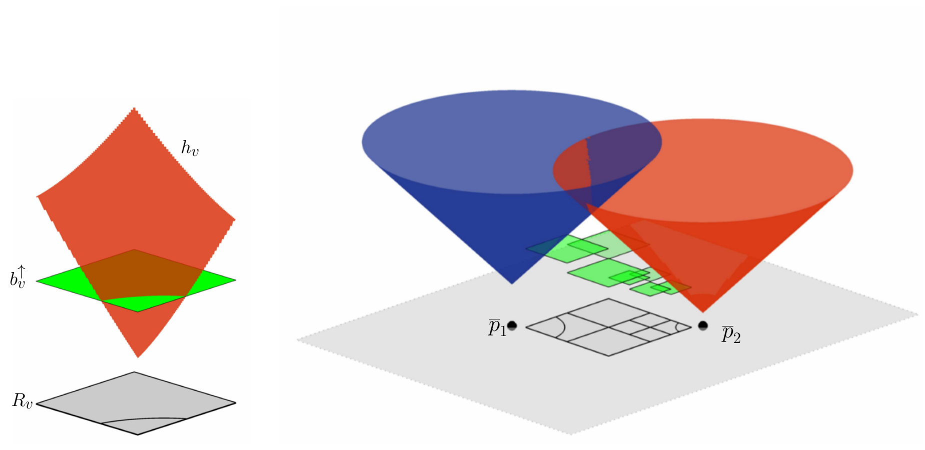
Here is the outline of the algorithm. We construct two sets $\mathcal{B}_{\text{in}}$ and $\mathcal{B}_{\text{out}}$ of canonical squares. Set $\mathcal{B} = \mathcal{B}_{\text{in}} \cup \mathcal{B}_{\text{out}}$. The size of $\mathcal{B}$, denoted by m, will be $O((n/\varepsilon^2)\log(1/\varepsilon))$, and we construct $\mathcal{B}$ in $O(n \log n + (n/\varepsilon^2)\log(1/\varepsilon))$ time. We build, in $O(m \log m)$ time, a compressed quad tree $\mathbb{T}$ of size $O(m)$ on $\mathcal{B}$, and preprocess it in additional $O(m)$ time so that for a point $q \in \mathbb{R}^2$, the exposed node of $\mathbb{T}$ containing q can be computed in $O(\log m)$ time. Let $\mathbb{M}$ be the planar subdivision induced by the exposed nodes of $\mathbb{T}$. We refine each cell of $\mathbb{M}$ into $O(1)$ faces to construct an ε-EVD of $\mathcal{P}$. More precisely, for a point $x \in \mathbb{R}^2$, let $\mathcal{P}_{\text{in}}[x] = \{P_i \mid x \in C_i$ and $\mathcal{P}_{\text{out}}[x] = \{P_i \mid x \notin C_i\}$. $\mathbb{T}$ has the property that for every exposed node v, $\mathcal{P}_{\text{in}}[x]$ and $\mathcal{P}_{\text{out}}[x]$ are the same for all points in the region R_v. We denote these sets by $\mathcal{P}_{\text{in}}[v]$ and $\mathcal{P}_{\text{out}}[v]$. We associate two representative points $P_v^{\text{in}} \in \mathcal{P}_{\text{in}}[v]$, $P_v^{\text{out}} \in \mathcal{P}_{\text{out}}[v]$ such that P_v^{in} is an ε-ENN of any point of R_v in $\mathcal{P}_{\text{in}}[v]$ and P_v^{out} is an ε-ENN of any point of R_v in $\mathcal{P}_{\text{out}}[v]$. If $P_v^{\text{in}} = P_i$, we store the canonical square $\square_v$ of the function g_i that contains R_v, and if $P_v^{\text{out}} = P_j$, we also store the centroid $\overline{p}_j$ of P_j at v.

For all $x \in R_v$, $g_i(x)$ is constant and $g_j(x) = \|x - \overline{p}_j\| + \mathsf{Ed}(P_j, \overline{p}_j)$. The minimization diagram of g_i and g_j within R_v, denoted by Σ_v, has $O(1)$ size; see Figure 7(a). We compute Σ_v for all exposed nodes of $\mathbb{T}$ and show that the planar subdivision induced by the Σ_v's is the desired ε-EVD of $\mathcal{P}$; Figure 7(b) shows a section of such a planar subdivision.

We first describe the computation of $\mathcal{B}_{\text{in}}$ and $\mathcal{B}_{\text{out}}$ followed by the computation of the representative points P_v^{in} and P_v^{out} for each exposed node v in $\mathbb{T}$. Finally, we describe how to construct an ε-EVD using the representative points.

Constructing $\mathcal{B}_{\text{in}}$. For $1 \leq i \leq n$, let $\mathcal{B}_i$ be the set of canonical squares that define the pieces of the piecewise-constant portion of g_i. Set $\mathcal{B}_{\text{in}} = \bigcup_{i=1}^n \mathcal{B}_i$. For each $\square \in \mathcal{B}_i$, we associate a value $\delta_\square$ with $\square$, which is $g_i(x)$ for any $x \in \square$. If a square $\square$ appears in multiple $\mathcal{B}_i$'s, we keep only one copy of $\square$ in $\mathcal{B}_{\text{in}}$ and $\delta_\square$ is the minimum of the values associated with the different copies of $\square$. For each square $\square \in \mathcal{B}_i$, we set $P_\square = P_i$.

Constructing $\mathcal{B}_{\text{out}}$. $\mathcal{B}_{\text{out}}$ is constructed using the algorithm by Arya *et al.* [6] for computing an ε-VD of a set S of N (certain) points. We therefore sketch their algorithm in $\mathbb{R}^2$. Two point sets $A, B \subset \mathbb{R}^2$ are called *α-well-separated* if A and B can be contained in disks D_A and D_B respectively, whose centers are at distance Δ and whose radii are at most $\alpha\Delta$; see Figure 6(a). A partition of $S \times S$ into a family $Z = \{(A_1, B_1), \ldots, (A_M, B_M)\}$ of α-well-separated pairs is called an *α-well-separated pair decomposition* (α-WSPD) of S. It is well known that an α-WSPD of S with $O(N/\varepsilon^2)$



(a)

(b)

Figure 7. (a) The minimization diagram, Σ_v (shown below) of $b_v^{\uparrow}$ and h_v for an exposed node v of $\mathbb{T}$. The complexity of Σ_v is $O(1)$. (b) A portion of the ε-EVD (shown below) obtained by replacing each cell of $\mathbb{M}$ by Σ_v. The corresponding $b_v^{\uparrow}$'s (raised squares) and h_v's (cones) are also shown.

pairs can be computed in $O((N/\varepsilon^2)\log N)$ time [18]. Arya *et al.* [6] first compute a $(1/8)$-WSPD Z of S. Let (A_i, B_i) be a pair in Z. Without loss of generality, assume that A_i and B_i are contained in disks of radii $1/8$ and the centers of these disks are at distance 1. Let ξ_i be the midpoint of these centers. They construct a family of $s = O(\log_2(1/\varepsilon))$ disks $D_1, \ldots, D_s$ centered at ξ_i where the radius of D_j is 2^{j-3}. Let $\gamma_j = 2^{j-c-\lceil \log_2(1/\varepsilon)\rceil}$ where $c \geq 3$ is a constant. Let $\mathcal{C}_j$ be the set of canonical squares of side length γ_j that intersect D_j; $|\mathcal{C}_j| = O(1/\varepsilon^2)$; see Figure 6(b). Set $\tilde{\mathcal{B}}_i = \bigcup_{1\leq j\leq s}\mathcal{C}_j$. They repeat the above procedure for each pair in Z. Let $\tilde{\mathcal{B}}$ be the overall set of canonical squares constructed; $|\tilde{\mathcal{B}}| = O((N/\varepsilon^2)\log(1/\varepsilon^2))$. $\tilde{\mathcal{B}}$ can be constructed in $O((N/\varepsilon^2)\log(1/\varepsilon) + N\log N)$ time. Next, they store $\tilde{\mathcal{B}}$ into a compressed quadtree to construct an ε-VD of S.

We adapt their procedure as follows. For $1 \leq i \leq n$, as before, let $\overline{p}_i$ be the centroid of P_i and C_i the square outside which $g_i(x) = \|x - \overline{p}_i\| + \mathsf{Ed}(P_i, \overline{p}_i)$. Set $\overline{P} = \{\overline{p}_i \mid 1 \leq i \leq n\}$. We execute the procedure of Arya *et al.* [6] on $\overline{P}$ and generate a set $\mathcal{B}_{\text{out}}$ of $O((n/\varepsilon^2)\log(1/\varepsilon))$ canonical squares.

Computing the representative points. For a point $x \in \mathbb{R}^2$, let $\overline{P}_{\text{out}}[x] = \{\overline{p}_i \mid P_i \in \mathcal{P}_{\text{out}}[x]\}$. Similar to the index in Section 4.1, we construct an index for answering stabbing queries which can find, for a query point q, which squares in $\mathcal{C} = \bigcup_{i=1}^{n} C_i$ do not contain q and thus, find $\overline{P}_{\text{out}}[q]$. This index stores a family of canonical subsets of $\overline{P}$ such that for any query point q, $\overline{P}_{\text{out}}[q]$ can be represented as the union of $O(\log^2 n)$ canonical subsets. For each of the canonical subsets, we also store an $(\varepsilon/12)$-VD from Arya *et al.* [6] of this subset. The total space required for the index is $O((n/\varepsilon^2)\log^2 n \log(1/\varepsilon))$ and it takes the same time to construct. For a query point q, we can now compute an $(\varepsilon/12)$-NN of q in $\overline{P}_{\text{out}}[q]$ in $O(\log^2 n \log(n/\varepsilon))$ time. This index is only needed for preprocessing and removed after representative points have been computed.

We build a compressed quadtree $\mathbb{T}$ on $\mathcal{B} = \mathcal{B}_{\text{in}} \cup \mathcal{B}_{\text{out}}$ as mentioned above. Let v be an exposed node of $\mathbb{T}$. If none of the ancestors of v (including v itself) stores a square of $\mathcal{B}_{\text{in}}$, P_v^{in} is undefined. Otherwise, among the squares $\square$ of $\mathcal{B}_{\text{in}}$ stored at the ancestors of v, let $\overline{\square}$ be the one with the smallest value of $\delta_{\square}$. We set $P_v^{\text{in}} = P_{\overline{\square}}$, $b_v = \overline{\square}$, and $b_v^{\uparrow}$ to the square in $\mathbb{R}^3$ obtained by lifting $\overline{\square}$ to the height $\delta_{\overline{\square}}$.

Next, we pick a point $x \in R_v$ and compute an $(\varepsilon/12)$-NN of x in $\overline{P}_{\text{out}}[x]$, say $\overline{p}_i$. We set $P_v^{\text{out}} = P_i$ and $\overline{p}_v = \overline{p}_i$. Let $h_v(x) = \|x - \overline{p}_i\| + \mathsf{Ed}(P_i, \overline{p}_i)$.

Finally, we compute the minimization diagram Σ_v of $b_v^{\uparrow}$ and h_v within R_v; see Figure 7(a). By replacing each cell R_v of $\mathbb{M}$ with Σ_v, we obtain the desired ε-EVD of $\mathcal{P}$, whose size is $O(m) = O((n/\varepsilon^2)\log(1/\varepsilon))$; Figure 7(b) shows a portion of such an ε-EVD. The total time spent in constructing this ε-EVD is $O((n/\varepsilon^2)\log^2 n \log(n/\varepsilon)\log(1/\varepsilon))$.

The correctness of the algorithm follows from the following lemma.

Lemma 5.5. *Let q be a point lying in R_v for an exposed node v of $\mathbb{T}$. Let P_q^{out} and P_q^{in} be the expected nearest neighbor of q in $\mathcal{P}_{\text{out}}[q]$ and $\mathcal{P}_{\text{in}}[q]$ respectively. Then,*

(i) $\mathsf{Ed}(P_v^{\text{in}}, q) \leq (1+\varepsilon)\mathsf{Ed}(P_q^{\text{in}}, q)$.

(ii) $\mathsf{Ed}(P_v^{\text{out}}, q) \leq (1+\varepsilon)\mathsf{Ed}(P_q^{\text{out}}, q)$.

Proof. (i) Let $\mathcal{B}_q = \{\square \in \mathcal{B}_{\text{in}} \mid P_i \in \mathcal{P}_{\text{in}}[q] \wedge q \in \square\}$. By construction, each square in $\mathcal{B}_q$ is stored at an ancestor of v in $\mathbb{T}$. Hence, $P_v^{\text{in}} = \arg\min_{P_i \in \mathcal{P}_{\text{in}}[q]} g_i(q)$. Now, (i) follows from Lemma 5.4.

(ii) By construction, the set $\{C_i \mid x \notin C_i\}$ is the same for all $x \in R_v$. Therefore, $\mathcal{P}_{\text{out}}[q] = \mathcal{P}_{\text{out}}[v]$ and $P_v^{\text{out}} \in \mathcal{P}_{\text{out}}[q]$. Let $\overline{p}_v$ and $\overline{p}_q$ be the centroids of P_v^{out} and P_q^{out} respectively. The argument in Arya *et al.* [6] implies that $\|\overline{p}_v - q\| \leq (1+\varepsilon/3)\|\overline{p}_q - q\|$. Hence,

$$\begin{aligned}\mathsf{Ed}(P_v^{\text{out}}, q) &\leq (1+\varepsilon/3)\|\overline{p}_q - q\| + \mathsf{Ed}(P_v^{\text{out}}, \overline{p}_v)\\ &\leq (1+\varepsilon/3)\|\overline{p}_q - q\| + \varepsilon/24\|\overline{p}_v - q\|\\ &\leq (1+\varepsilon/3)(1+\varepsilon/24)(\|\overline{p}_q - q\| + \mathsf{Ed}(P_q^{\text{out}}, \overline{p}_q)).\end{aligned}$$

Since $P_v \in \mathcal{P}_{\text{out}}[q]$, $\|\overline{p}_v - q\| \geq 24\mathsf{Ed}(P_v, \overline{p}_v)/\varepsilon$. Thus,

$$\mathsf{Ed}(P_v^{\text{out}}, q) \leq (1+\varepsilon/2)(1+\varepsilon/3)\mathsf{Ed}(P_q^{\text{out}}, q) \leq (1+\varepsilon)\mathsf{Ed}(P_q^{\text{out}}, q),$$

proving part (ii). □

Putting everything together, we conclude the following.

THEOREM 5.6. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a pdf of description complexity k, let $0 < \varepsilon < 1$ be a parameter and let $\mathrm{d}(\cdot,\cdot)$ be the Euclidean distance. An ε-EVD of $\mathcal{P}$ of size $O((n/\varepsilon^2)\log(1/\varepsilon))$ can be constructed in $O((n/\varepsilon^2)\log^2 n \log(n/\varepsilon)\log(1/\varepsilon))$ time. It can be processed in additional $O((n/\varepsilon^2)\log(1/\varepsilon)))$ time into an index of $O((n/\varepsilon^2)\log(1/\varepsilon))$ size so that an ε-ENN of a query point can be constructed in $O(\log(n/\varepsilon))$ time.*

Noting that for an uncertain point P, the function g_P that approximates $\mathsf{Ed}(P, x)$ under any L_p metric can be constructed in the same time, we also obtain the following.

THEOREM 5.7. *Let $\mathcal{P}$ be a set of n uncertain points in $\mathbb{R}^2$, each with a pdf of description complexity k, let $0 < \varepsilon < 1$ be a parameter. For any L_p metric, an ε-EVD of $\mathcal{P}$ of size $O((n/\varepsilon^2)\log(1/\varepsilon))$ can be constructed in $O((n/\varepsilon^2)\log^2 n \log(n/\varepsilon)\log(1/\varepsilon))$ time. It can be processed in $O((n/\varepsilon^2)\log(1/\varepsilon))$ additional time into an index of $O((n/\varepsilon^2)\log(1/\varepsilon))$ size so that an ε-ENN of a query point under the L_p metric can be constructed in $O(\log(n/\varepsilon))$ time.*

Remarks. (i) Note that we do not have to construct the minimization diagram Σ_v for each exposed node $v \in \mathbb{T}$. We can simply use $P_v^{\text{in}}, P_v^{\text{out}}, b_v^{\uparrow}$ and h_v stored at v to compute an ε-ENN of a query point.

(ii) The algorithm can be extended to higher dimensions. The size of the index becomes $O((n/\varepsilon^d)\log(1/\varepsilon))$, the preprocessing time become $O((n/\varepsilon^d)\log^d n\log(1/\varepsilon))$, and the query time remains the same.

5.3 Uncertain query

Let $\mathcal{P} = \{p_1, \cdots, p_n\}$ be a set of (certain) points in $\mathbb{R}^2$. For an uncertain query point Q of description complexity k and a parameter $0 < \varepsilon < 1$, we wish to compute its ε-ENN in $\mathcal{P}$. We preprocess $\mathcal{P}$ into a compressed quadtree $\mathbb{T}$ as described in Section 2. We also preprocess $\mathcal{P}$ for answering NN queries, by constructing its Voronoi diagram and preprocessing it for point-location queries. The size of the index is $O(n)$ and it can be built in $O(n \log n)$ time [16].

To answer a given query Q, we construct the function $g_Q : \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ using Lemma 5.4. Let $\mathcal{B}$ be the set of canonical squares defining g_Q. For each $\square \in \mathcal{B}$, we query $\mathbb{T}$ and report a point $p_\square \in \square \cap \mathcal{P}$ if there exists one. Among all the points reported, we return the point $p^* = \arg\min_{p_\square} g_Q(p_\square)$. If no point is reported, then we return the point of $\mathcal{P}$ that is closest to $\overline{q}$, the centroid of Q. The correctness of the algorithm follows from the Lemma 5.4. Querying each $\square \in \mathcal{B}$ takes $O(\log n)$ time, by Lemma 2.1, and the NN of $\overline{q}$ can be computed in $O(\log n)$ time, so we conclude the following:

THEOREM 5.8. *Let $\mathcal{P}$ be a set of n (certain) points in $\mathbb{R}^2$. An index of $O(n)$ size can be built on $\mathcal{P}$ in $O(n \log n)$ time so that for an uncertain query point Q with a pdf of description complexity k and for a parameter $0 < \varepsilon < 1$, an ε-ENN of Q can be computed in $O((k/\varepsilon^2)\log(1/\varepsilon)\log n)$ time.*

Remarks. (i) The algorithm can be extended to higher dimensions. The size and the preprocessing time remain the same, but the query time in $\mathbb{R}^d$ increases to $O((k/\varepsilon^d)\log(1/\varepsilon)\log n)$.

(ii) All pieces of the function g_Q need not be computed in the beginning itself. They can be constructed hierarchically while querying the compressed quadtree on $\mathcal{P}$. This does not affect the worst-case running time but it is more efficient in practice.

6. CONCLUSION

In this paper we considered the problem of answering NN queries under uncertainty. We used a probabilistic framework to model the uncertainty in the location of input data or query point, and presented indexing schemes of linear or near-linear size that answer exact or ε-approximate ENN queries in $\mathbb{R}^2$ in polylog(n) time under squared Euclidean, L_1, L_2, and L_∞ distance functions. As far as we know, these are the first methods to obtain such bounds. We conclude by mentioning a few open problems:

(i) What is the combinatorial complexity of EVD($\mathcal{P}$) when $d(\cdot,\cdot)$ is the Euclidean distance? Can a quadratic upper bound be proved? Although the algebraic complexity of a bisector is large, the combinatorial complexity, i.e., the number of vertices, can be small.

(ii) The expected distance is not a reliable indicator when the variance of the pdfs is not small. In this case, one is interested in computing a point that is the nearest neighbor with highest probability or the points that are the nearest neighbors with probability higher than a given threshold. Is there a linear-size index to answer these queries in sublinear time in the worst case? This problem seems hard even for very simple pdfs such as Gaussians.

7. REFERENCES

[1] P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi, Indexing uncertain data, *Proc. ACM Symposium on Principles of Database Systems*, 2009, pp. 137–146.

[2] P. K. Agarwal, S. Har-Peled, M. Sharir, and Y. Wang, Hausdorff distance under translation for points and balls, *ACM Transactions on Algorithms*, 6 (2010), 71:1–71:26.

[3] P. K. Agarwal and J. Matousek, Ray shooting and parametric search, *SIAM Journal on Computing*, 22 (1993), 794–806.

[4] C. C. Aggarwal, *Managing and Mining Uncertain Data*, Springer, 2009.

[5] A. Andoni and P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, *Communications of the ACM*, 51 (2008), 117–122.

[6] S. Arya, T. Malamatos, and D. M. Mount, Space-time tradeoffs for approximate nearest neighbor searching, *Journal of the ACM*, 57 (2009), 1:1–1:54.

[7] F. Aurenhammer and R. Klein, Voronoi diagrams, in: *Handbook of Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), Elsevier Science Publishers, Amsterdam, 2000, pp. 201–290.

[8] G. Beskales, M. A. Soliman, and I. F. IIyas, Efficient search for the top-k probable nearest neighbors in uncertain databases, *Proc. International Conference on Very Large Databases*, 1 (2008), 326–339.

[9] S. Cabello, Approximation algorithms for spreading points, *Journal of Algorithmss*, 62 (2007), 49–73.

[10] S. Cabello and M. J. van Kreveld, Approximation algorithms for aligning points, *Proc. 19th ACM Symposium on Computational Geometry*, 2003, pp. 20–28.

[11] R. Cheng, J. Chen, M. Mokbel, and C.-Y. Chow, Probabilistic verifiers: Evaluating constrained nearest-neighbor queries over uncertain data, *Proc. IEEE International Conference on Data Engineering*, 2008, pp. 973–982.

[12] R. Cheng, L. Chen, J. Chen, and X. Xie, Evaluating probability threshold k-nearest-neighbor queries over uncertain data, *Proc. 12th International Conference on Extending Database Technology: Advances in Database Technology*, 2009, pp. 672–683.

[13] R. Cheng, X. Xie, M. L. Yiu, J. Chen, and L. Sun, Uv-diagram: A voronoi diagram for uncertain data, *Proc. IEEE International Conference on Data Engineering*, 2010, pp. 796–807.

[14] K. L. Clarkson, Nearest-neighbor searching and metric space dimensions, *Nearest-Neighbor Methods for Learning and Vision: Theory and Practice*, (2006), 15–59.

[15] N. N. Dalvi, C. Ré, and D. Suciu, Probabilistic databases: diamonds in the dirt, *Communications of the ACM*, 52 (2009), 86–94.

[16] M. de Berg, M. van Kreveld, M. Overmars, and O. Schwarzkopf, *Computational Geometry: Algorithms and Applications*, Springer-Verlag, 2000.

[17] A. Guttman, R-trees: a dynamic index structure for spatial searching, *Proc. ACM SIGMOD International Conference on Management of Data*, 1984, pp. 47–57.

[18] S. Har-Peled, *Geometric Approximation Algorithms*, American Mathematical Society, 2011.

[19] M. Hua, J. Pei, W. Zhang, and X. Lin, Ranking queries on uncertain data: a probabilistic threshold approach, *Proc. ACM SIGMOD International Conference on Management of Data*, 2008, pp. 673–686.

[20] P. Indyk, Nearest neighbors in high-dimensional spaces, in: *Handbook of Discrete and Computational Geometry* (J. E. Goodman and J. O'Rourke, eds.), CRC Press LLC, 2004.

[21] M. Jooyandeh, A. Mohades, and M. Mirzakhah, Uncertain voronoi diagram, *Information Processing Letters*, 109 (2009), 709–712.

[22] P. Kamousi, T. M. Chan, and S. Suri, Closest pair and the post office problem for stochastic points, *Proc. 12th International Conference on Algorithms and Data Structures*, 2011, pp. 548–559.

[23] H.-P. Kriegel, P. Kunath, and M. Renz, Probabilistic nearest-neighbor query on uncertain objects, *Proc. 12th International Conference on Database Systems for Advanced Applications*, 2007, pp. 337–348.

[24] F. Li, B. Yao, and P. Kumar, Group enclosing queries, *IEEE Transactions on Knowledge and Data Engineering*, 23 (2011), 1526 –1540.

[25] H. Li, H. Lu, B. Huang, and Z. Huang, Two ellipse-based pruning methods for group nearest neighbor queries, *Proc. 13th Annual ACM International Workshop on Geographic Information Systems*, 2005, pp. 192–199.

[26] Y. Li, F. Li, K. Yi, B. Yao, and M. Wang, Flexible aggregate similarity search, *Proc. ACM SIGMOD International Conference on Management of Data*, 2011, pp. 1009–1020.

[27] X. Lian and L. Chen, Probabilistic group nearest neighbor queries in uncertain databases, *IEEE Transactions on Knowledge and Data Engineering*, 20 (2008), 809–824.

[28] V. Ljosa and A. Singh, Apla: Indexing arbitrary probability distributions, *Proc. IEEE International Conference on Data Engineering*, 2007, pp. 946 –955.

[29] M. Löffler and M. J. van Kreveld, Largest bounding box, smallest diameter, and related problems on imprecise points, *Computational Geometry*, 43 (2010), 419–433.

[30] Y. Luo, H. Chen, K. Furuse, and N. Ohbo, Efficient methods in finding aggregate nearest neighbor by projection-based filtering, *Proc. 12th International Conference on Computational Science and Its Applications*, 2007, pp. 821–833.

[31] D. Papadias, Q. Shen, Y. Tao, and K. Mouratidis, Group nearest neighbor queries, *Proc. IEEE International Conference on Data Engineering*, 2004, pp. 301 – 312.

[32] N. Sarnak and R. E. Tarjan, Planar point location using persistent search trees, *Communications of the ACM*, 29 (1986), 669–679.

[33] J. Sember and W. Evans, Guaranteed voronoi diagrams of uncertain sites, *Proc. 20th Canadian Conference on Computational Geometry*, 2008.

[34] M. Sharifzadeh and C. Shahabi, Vor-tree: R-trees with voronoi diagrams for efficient processing of spatial nearest neighbor queries, *Proc. International Conference on Very Large Databases*, 3 (2010), 1231–1242.

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

[36] G. Trajcevski, R. Tamassia, H. Ding, P. Scheuermann, and I. F. Cruz, Continuous probabilistic nearest-neighbor queries for uncertain trajectories, *Proc. 12th International Conference on Extending Database Technology: Advances in Database Technology*, 2009, pp. 874–885.

[37] M. J. van Kreveld, M. Löffler, and J. S. B. Mitchell, Preprocessing imprecise points and splitting triangulations, *SIAM Journal on Computing*, 39 (2010), 2990–3000.

[38] M. Yiu, N. Mamoulis, and D. Papadias, Aggregate nearest neighbor queries in road networks, *IEEE Transactions on, Knowledge and Data Engineering*, 17 (2005), 820 – 833.

[39] S. M. Yuen, Y. Tao, X. Xiao, J. Pei, and D. Zhang, Superseding nearest neighbor search on uncertain spatial databases, *IEEE Transactions on Knowledge and Data Engineering*, 22 (2010), 1041 –1055.