# Convex Hulls under Uncertainty *

Pankaj K. Agarwal † Sariel Har-Peled ‡ Subhash Suri § Hakan Yıldız ¶
Wuzhou Zhang ‖

March 3, 2015

**Abstract**

We study the convex-hull problem in a probabilistic setting, motivated by the need to handle data uncertainty inherent in many applications, including sensor databases, location-based services and computer vision. In our framework, the uncertainty of each input point is described by a probability distribution over a finite number of possible locations including a *null* location to account for non-existence of the point. Our results include both exact and approximation algorithms for computing the probability of a query point lying inside the convex hull of the input, time-space tradeoffs for the membership queries, a connection between Tukey depth and membership queries, as well as a new notion of $\beta$-hull that may be a useful representation of uncertain hulls.

## 1. Introduction

Convex hull is a fundamental structure in mathematics and computational geometry. Given a set of points $\mathcal{P}$ in $d$-space, the *convex hull* of $\mathcal{P}$ is the minimal convex polytope that contains all points in $\mathcal{P}$. Convex hulls have applications in a variety of areas including but not limited to computer graphics, image processing, pattern recognition, robotics, combinatorics and statistics. Owing to their importance in practice, the algorithms for computing convex hulls are well-studied. The

*A preliminary version of this paper appeared in the proceedings of 22nd European Symposium of Algorithms (ESA'14).

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. S. Har-Peled is supported by NSF grants CCF-09-15984 and CCF-12-17462. S. Suri and H. Yıldız are supported by NSF grants CCF-1161495 and CNS-1035917.

†Department of Computer Science; Duke University; Durham, NC, 27708, USA; pankaj@cs.duke.edu; http://www.cs.duke.edu/~pankaj/.

‡Department of Computer Science; University of Illinois; 201 N. Goodwin Avenue; Urbana, IL, 61801, USA; sariel@uiuc.edu; http://sarielhp.org/ .

§Department of Computer Science; University of California; Santa Barbara; CA, 93106, USA; suri@cs.ucsb.edu; http://www.cs.ucsb.edu/~suri.

¶Department of Computer Science; University of California; Santa Barbara; CA, 93106, USA; hakan@cs.ucsb.edu; http://www.cs.ucsb.edu/~hakan.

‖Department of Computer Science; Duke University; Durham, NC, 27708, USA; wuzhou@cs.duke.edu; http://www.cs.duke.edu/~wuzhou/.

convex hull of $n$ points can be computed in $O(n \log n)$ time [17,29] for $d = 2, 3$, and in $O(n^{\lfloor d/2 \rfloor})$ time [10] for $d > 3$. These bounds are worst-case optimal [7,10], and output-sensitive algorithms are also known [8,12,22,25,30]. See the survey [31] for an overview of known results. In many applications, such as sensor databases, location-based services or computer vision, the location and sometimes even the existence of the data is uncertain, but statistical information can be used as a probability distribution guide for data. This raises the natural computational question: what is a robust and useful convex hull representation for such an uncertain input, and how well can we compute it? We explore this problem under two simple models in which both the location and the existence (presence) of each point is described probabilistically, and study basic questions such as what is the probability of a query point lying inside the convex hull, or what does the probability distribution of the convex hull over the space look like.

**Uncertainty models.** We focus on two models of uncertainty: unipoint and multipoint. In the *unipoint model*, each input point has a fixed location but it only exists probabilistically. Specifically, the input $\mathcal{P}$ is a set of pairs $\{(p_1, \gamma_1), \ldots, (p_n, \gamma_n)\}$ where each $p_i$ is a point in $\mathbb{R}^d$ and each $\gamma_i$ is a real number in the range $(0, 1]$ denoting the probability of $p_i$'s existence. The existence probabilities of different points are independent; $P = \{p_1, \ldots, p_n\}$ denotes the set of sites in $\mathcal{P}$.

In the *multipoint model*, each point probabilistically exists at one of multiple possible sites. Specifically, $\mathcal{P}$ is a set of pairs $\{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$ where each $P_i$ is a set of $n_i$ points and each $\Gamma_i$ is a set of $n_i$ real values in the range $(0, 1]$. The set $P_i = \left\{p_i^1, \ldots, p_i^{n_i}\right\}$ describes the possible sites for the $i$th point of $\mathcal{P}$ and the set $\Gamma_i = \left\{\gamma_i^1, \ldots, \gamma_i^{n_i}\right\}$ describes the associated probability distribution. The probabilities $\gamma_i^j$ correspond to disjoint events and therefore sum to at most 1. By allowing the sum to be less than one, this model also accounts for the possibility of the point not existing (i.e. the *null* location)—thus, the multipoint model generalizes the unipoint model. In the multipoint model, $P = \bigcup_{i=1}^m P_i$ refers to the set of all sites and $n = |P|$.

**Our results.** The main results of our paper can be summarized as follows.

(A) We show (in Section 2) that the membership probability of a query point $q \in \mathbb{R}^d$, namely, the probability of $q$ being inside the convex hull of $\mathcal{P}$, can be computed in $O(n \log n)$ time for $d = 2$. For $d \geq 3$, assuming the input and the query point are in general position, the membership probability can be computed in $O(n^d)$ time. (This result is shown in Section 3.) The results hold for both unipoint and multipoint models.

(B) Next we describe two algorithms (in Section 4) to preprocess $\mathcal{P}$ into a data structure so that for a query point its membership probability in $\mathcal{P}$ can be answered quickly. The first algorithm constructs a *probability map* $\mathbb{M}(\mathcal{P})$, a partition of $\mathbb{R}^d$ into convex cells, so that all points in a single cell have the same membership probability. We show that $\mathbb{M}(\mathcal{P})$ has size $\Theta(n^{d^2})$, and for $d = 2$ it can be computed in optimal $O(n^4)$ time. The second one is a sampling-based Monte Carlo algorithm for constructing a near-linear-size data structure that can approximate the membership probability with high likelihood in sublinear time for any fixed dimension.

(C) We show (in Section 5) a connection between the membership probability and the Tukey depth, which can be used to approximate cells of high membership probabilities. For $d = 2$, this relationship also leads to an efficient data structure.

(D) Finally, we introduce the notion of *β-hull* (in Section 6) as another approximate representation for uncertain convex hulls in the multipoint model: for $\beta \in [0, 1]$, a convex set $C$ is called *β-dense* for $\mathcal{P}$, if $C$ contains at least $\beta$ fraction of each uncertain point. The $\beta$-hull of $\mathcal{P}$ is the intersection of all $\beta$-dense sets for $\mathcal{P}$. We show that for $d = 2$, the $\beta$-hull of $\mathcal{P}$ can be computed in $O(n \log^5 n)$ time.

**Related work.** There is extensive and ongoing research in the database community on uncertain data; see [13] for a survey. In the computational geometry community, the early work relied on deterministic models for uncertainty (see e.g. [16,23]), but more recently probabilistic models of uncertainty, which are closer to the models used in statistics and machine learning, have been explored [1,2,3,4,20,21,27,28,32,33]. The convex-hull problem over uncertain data has received some attention very recently. Suri *et al.* [32] showed that the problem of computing the most likely convex hull of a point set in the multipoint model is NP-hard. Even in the unipoint model, the problem is NP-hard for $d \geq 3$. They also presented an $O(n^3)$-time algorithm for computing the most likely convex hull under the unipoint model in $\mathbb{R}^2$. Zhao *et al.* [34] investigated the problem of computing the probability of each uncertain point lying on the convex hull, where they aimed to return the set of (uncertain) input points whose probabilities of being on the convex hull are at least some threshold. Jørgensen *et al.* [19] showed that the distribution of properties, such as areas or perimeters, of the convex hull of $\mathcal{P}$ may have $\Omega(\Pi_{i=1}^m n_i)$ complexity.

## 2. Membership Probability in the Plane

In this section, we describe how to compute the probability of a given point lying in the convex hull of a given uncertain point set in the plane. For simplicity, we assume that the input is non-degenerate, meaning that all possible point sites, including the query point $q$, are in general position: no two sites have the same coordinate along any dimension and no three sites are collinear. We defer the discussion of how to handle such degeneracies to Section 2.3. We begin our discussion with the unipoint case.

### 2.1. The Unipoint Model

Let $\mathcal{P} = \{(p_1, \gamma_1), \ldots, (p_n, \gamma_n)\}$ be a set of $n$ uncertain points in $\mathbb{R}^2$ under the unipoint model. We let $P$ denote the set of all sites of $\mathcal{P}$, namely, $\{p_1, \ldots, p_n\}$. Let $\mathsf{R}$ denote the outcome of the probabilistic experiment by choosing each $p_i$ with probability $\gamma_i$. $\mathsf{R}$ is a random subset of $P$. For a subset $B \subseteq P$, let $\gamma(B) = \Pr[\mathsf{R} = B]$. Then

$$\gamma(B) = \prod_{p_i \in B} \gamma_i \times \prod_{p_i \notin B} \overline{\gamma_i},$$

where $\overline{\gamma_i}$ is the complementary probability $(1 - \gamma_i)$. Given a query point $q$, we want to compute its membership probability, denoted by $\mu(q)$, the probability that $q$ lies in the convex hull of $\mathsf{R}$. Let $\mathrm{CH}(B)$ denote the convex hull of $B$. By definition, $\mu(q)$ can be written as

$$\mu(q) = \sum_{B \subseteq P \mid q \in \mathrm{CH}(B)} \gamma(B), \tag{1}$$

which unfortunately involves an exponential number of terms (possible subsets $B$). Our polynomial-time scheme for computing $\mu(q)$ builds on the following simple observation: $q$ is *outside* $\mathrm{CH}(\mathsf{R})$ if and only if $q$ is a vertex of the convex hull $\mathrm{CH}(\mathsf{R} \cup \{q\})$. For ease of reference, let $C$ denote

$\text{CH}(\mathsf{R} \cup \{q\})$ and $V$ denote the set of vertices of $C$. Then the probability we want is $\mu(q) = 1 - \Pr[q \in V]$.

If $\mathsf{R} = \emptyset$, then clearly $C = \{q\}$ and $q \in V$. Otherwise, $|V| \geq 2$ and $q \in V$ implies that $q$ is an endpoint of exactly two edges on the boundary of $C$.[1] In this case, we define the first edge following $q$ in the counter-clockwise order of $C$ as the *witness edge* of $q \notin \text{CH}(\mathsf{R})$. (See Figure 1a.) It is easy to see that $q \in V$ if and only if $\mathsf{R} = \emptyset$ or (exclusively) $qp_i$ is a witness edge for some $1 \leq i \leq n$. Let $\pi_i(q) = \Pr\big[\, qp_i \text{ is a witness edge} \,\big]$. Thus,

$$1 - \mu(q) = \Pr\big[\, q \in V \,\big] = \Pr\big[\, \mathsf{R} = \emptyset \,\big] + \sum_{p_i \in P} \pi_i(q). \tag{2}$$

Let $G_i \subseteq P$ be the set of sites lying to the right of the oriented line $\ell_i$, spanned by the vector $\overrightarrow{qp_i}$. We observe that $qp_i$ is a witness edge if and only if (i) $p_i \in \mathsf{R}$ and (ii) $\mathsf{R} \cap G_i = \emptyset$. Therefore

$$\pi_i(q) = \gamma_i \cdot \prod_{p_j \in G_i} \overline{\gamma_j},$$

This expression can be computed in $O(n)$ time. Since $\Pr\big[\, \mathsf{R} = \emptyset \,\big]$ can be computed in linear time, $\mu(q)$ can be computed in $O(n^2)$ time. The computation time can be improved to $O(n \log n)$ as described below.

**Improving the running time.** The main idea is to compute the witness edge probabilities in radial order around $q$. We sort $P$ in counter-clockwise order around $q$. Without loss of generality, assume that the circular sequence $\langle p_1, \ldots, p_n \rangle$ is the resulting order. (See Figure 1b.) We first compute $\pi_1(q)$ in $O(n)$ time. Next for $i > 1$, we compute $\pi_i(q)$ from $\pi_{i-1}(q)$ in $O(1)$ amortized time as follows: Let $W_i$ denote the set of sites in the open wedge bounded by the rays emanating from $q$ in directions $\overrightarrow{p_{i-1}q}$ and $\overrightarrow{p_i q}$. (See Figure 1c.) Notice that $G_i = G_{i-1} \cup \{p_{i-1}\} \setminus W_i$. It follows that

$$\pi_i(q) = \frac{\gamma_i}{\gamma_{i-1}} \cdot \frac{\overline{\gamma_{i-1}}}{\prod_{p_j \in W_i} \overline{\gamma_j}} \cdot \pi_{i-1}(q).$$

The amortized cost of a single update is $O(1)$ because each site of $P$ enters $G_i$ at most twice. Finally, notice that we can easily keep track of the set $W_i$ during our radial sweep, as changes to this set follow the same radial order.

**Theorem 2.1.** *Given a set of $n$ uncertain points in $\mathbb{R}^2$ under the unipoint model, the membership probability of a query point $q$ can be computed in $O(n \log n)$ time.*

## 2.2. The Multipoint Model

Let $\mathcal{P} = \{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$ be a set of uncertain points in the multipoint model, as defined in the previous section. Recall that $P = \bigcup_{i=1}^{m} P_i$ is the set of all sites. Let $\mathsf{R}$ be the outcome of the experimental outcome, in which exactly one point of $P_i$ is chosen randomly — $p_i^j$ is chosen with probability $\gamma_i^j$.

[1] If $B$ consists of a single site $p_i$, then $C$ is the line segment $qp_i$. In this case, we consider the boundary of $C$ to be a cycle formed by two edges: one going from $q$ to $p_i$, and one going from $p_i$ back to $q$.

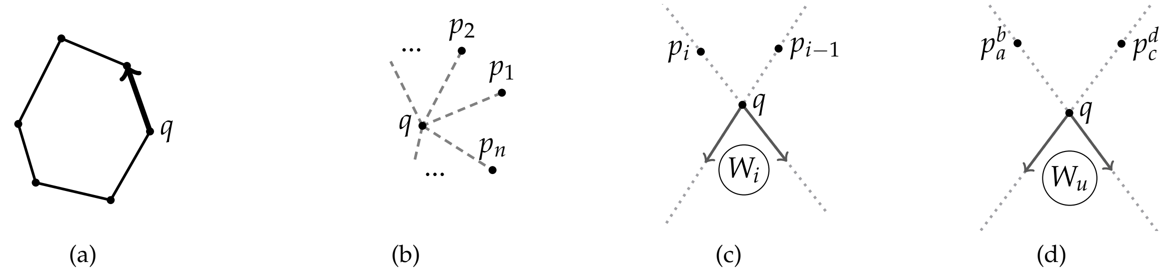


Figure 1: (a) A witness edge. (b) Sites in radial order around $q$. (c) The set $W_i$. (d) The set $W_u$.

For a subset $B \subseteq P$, let $\gamma(B) = \Pr[\mathsf{R} = B]$. Similarly to the unipoint model, the definition of $\gamma(B)$ involves a product of existence probabilities for all sites in $B$. The sites that are not in $B$, however, contribute to $\gamma(B)$ in a different way. Specifically, let $p_i^j$ be a site that is not in $B$. If $B$ contains another $p_i^l$ site from the $i$th point, then the non-existence probability of $p_i^j$ is irrelevant to $\gamma(B)$, because existence of $p_i^l$ already implies non-existence of $p_i^j$. If there is no such site $p_i^l$, then no site from the tuple of the $i$th point is in $B$. In that case, we just consider the probability that $i$th point does not exist at all, which is $1 - \sum_{1 \le j \le |P_i|} \gamma_i^j$. Finally, notice that if $B$ contains two sites from the same uncertain point, then it cannot be the outcome of an experiment.

For a point $q \in \mathbb{R}^2$, we again define $\mu(q) = \Pr[q \in \textsc{ch}(\mathsf{R})]$. We now describe how $\mu(q)$ is computed in the multipoint model. Let $C$ and $V$ be defined as above. As in the unipoint model, $q \in \textsc{ch}(\mathsf{R})$ if and only if $q \notin V$, thus $\mu(q) = 1 - \Pr[q \in V]$. Let $\pi_{ij}(q)$ denote the probability that $qp_i^j$ is a witness edge. We follow a similar strategy and decompose $\Pr[q \in V]$ as follows:

$$\Pr[q \in V] = \Pr[\mathsf{R} = \emptyset] + \sum_{\substack{1 \le i \le m \\ 1 \le j \le |P_i|}} \pi_{ij}(q).$$

The first term is trivial to compute in $O(n)$ time; $\pi_{ij}(q)$ is computed as follows. Let $G_{i,j}$ be the set of sites to the right of the oriented line spanning the vector $\overrightarrow{qp_i^j}$. As in the unipoint model, the segment $qp_i^j$ is a witness edge if and only if $p_i^j \in \mathsf{R}$ and $\mathsf{R} \cap G_{i,j} = \emptyset$. Hence,

$$\begin{aligned}
\pi_{ij}(q) &= \Pr\left[p_i^j \in \mathsf{R}\right] \times \Pr\left[\mathsf{R} \cap G_{i,j} = \emptyset \mid p_i^j \in \mathsf{R}\right] \\
&= \Pr\left[p_i^j \in \mathsf{R}\right] \times \prod_{1 \le k \le m} \Pr\left[\mathsf{R} \cap G_{i,j} \cap P_k = \emptyset \mid p_i^j \in \mathsf{R}\right] \\
&= \Pr\left[p_i^j \in \mathsf{R}\right] \times \prod_{\substack{1 \le k \le m \\ k \ne i}} \Pr\left[\mathsf{R} \cap P_k \cap G_{i,j} = \emptyset\right] \\
&= \gamma_i^j \times \prod_{\substack{1 \le k \le m \\ k \ne i}} \left(1 - \sum_{l \mid p_k^l \in G_{i,j}} \gamma_k^l\right).
\end{aligned}$$

This expression can be easily computed in $O(n)$ time. It follows that one can compute $\Pr[q \in V]$, thus $\mu(q)$, in $O(n^2)$ time.

As before, the computation time can be improved to $O(n \log n)$ by computing the witness edge probabilities in a radial order around $q$. Let the circular sequence $p'_1, p'_2, \ldots, p'_n$ be the counterclockwise order of all sites around $q$, where each $p'_u$ is a distinct site $p_a^b$. We first compute the probability that $qp'_1$ is the witness edge in $O(n)$ time and also remember the values of the intermediate factors used in the computation. (The factors inside the $\prod_{1 \le k \le m}$ expression.) Then, for increasing values of $u$ from 2 to $n$, we compute the probability that $qp'_u$ is the witness edge by updating the probability for $qp'_{u-1}$. As a first step to this update, we update the values of the intermediate factors. To be more specific, let $W_u$ denote the set of sites in the open wedge bounded by the rays emanating from $q$ in directions $\overrightarrow{p'_u q}$ and $\overrightarrow{p'_{u-1} q}$. Also, for simplicity, assume that $p'_u = p_a^b$ and $p'_{u-1} = p_c^d$. (See Figure 1d.) Notice that $G_{a,b} = G_{c,d} \cup \{p_c^d\} \setminus W_u$. Then, for each site $p_e^f$ in $W_u$, the $e$th factor increases by $\gamma_e^f$. Also, the $c$th factor decreases by $\gamma_c^d$. Finally, we temporarily set the value of the $a$th factor to 1 (to cover the case $k \neq i$ in the expression). Then, we can compute the witness edge probability for $qp'_u$ by multiplying the probability of $qp'_{u-1}$ with $\gamma_a^b / \gamma_c^d$ and the multiplicative change in each intermediate factor. The cost of a single update is $O(1)$ amortized, as each site can contribute to at most 4 updates as in the unipoint case.

**Theorem 2.2.** *Given a set $\mathcal{P}$ of uncertain points in the multipoint model with $n$ sites in total and a point $q$ in $\mathbb{R}^2$, the probability of $q$ being in the convex hull of $\mathcal{P}$ can be computed in $O(n \log n)$ time using linear space.*

### 2.3. Dealing with Degeneracies

In this section, we briefly explain how our algorithm for the planar case can be adapted to handle degeneracies. There are two main issues to be handled in the presence of degeneracies:

1. **A site may coincide with the query point $q$:** If this is the case, then the existence of such a site in $B$ implies that $q \in \text{CH}(B)$. We separately compute the probability that $q$ coincides with a site in $B$. The remaining portion of $\mu(q)$ is computed as before, however is conditioned on the non-existence of all sites that coincide with $q$. To be precise, we again compute the probability $1 - \Pr[q \in V]$, but this time on a reduced set of sites which does not involve the sites coinciding $q$. (In the multipoint model, this also requires adjusting the probabilities of the sites which belong to the same uncertain point with another site coinciding $q$.) Once this probability is computed, we further multiply it with the probability that no site coinciding $q$ exists in $B$.

2. $q$ **might be collinear with two or more other sites:** In this case, we need to re-define $G_i$ in a more careful manner. As before, let $\ell_i$ be the oriented line spanned by the vector $\overrightarrow{qp_i}$. Let $G_i^1 \subseteq P$ be the set of sites lying to the right of or on $\ell_i$. Let $G_i^2 \subseteq P$ be the set of sites lying on the segment $qp_i$. Then $G_i = G_i^1 - G_i^2$.

   A similar approach also applies to the multipoint model.

## 3. Membership Probability in $d$ Dimensions

In this section, we describe our algorithm for computing the membership probability for dimensions higher than two. This algorithm works correctly only for non-degenerate input. To be precise, we require that all sites, including the query point $q$, are in general position in the following

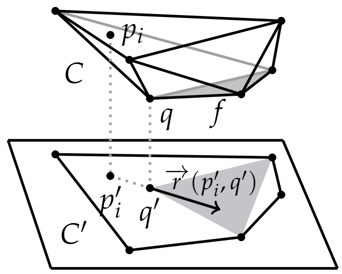


Figure 2: A three-dimensional example of an $p_i$-escaping face $f$ for $q$.

sense: for $2 \le k \le d$, the projection of no $k+1$ points of $P \cup \{q\}$ on a subspace spanned by any subset of $k$ coordinates lies on a $(k-1)$-hyperplane. We note that computing the membership probability for degenerate point sets in high dimensions is still an open problem. Our work can be considered a first step into understanding the complexity of the membership probability in high dimensions.

### 3.1. The Unipoint Model

The difficulty in extending the above to higher dimensions is an appropriate generalization of witness edges, which allow us to implicitly sum over exponentially many outcomes without over-counting. The following discussion explains the main idea we use for this generalization.

Let $B$ be an outcome, $C = \mathrm{CH}(B \cup \{q\})$ its convex hull, and $V$ the vertices of $C$. Let $\lambda(B \cup \{q\})$ denote the point with the lowest $x_d$-coordinate in $B \cup \{q\}$. Clearly, if $q$ is $\lambda(B \cup \{q\})$ then $q \in V$; otherwise, we condition the probability based on which point among $B$ is $\lambda(B \cup \{q\})$. Therefore, we can write

$$\Pr\Big[q \in V\Big] = \Pr\Big[q = \lambda(B \cup \{q\})\Big] + \sum_{1 \le i \le n} \Pr\Big[p_i = \lambda(B \cup \{q\}) \;\wedge\; q \in V\Big].$$

It is easy to compute the first term. We show below how to compute each term of the summation in $O(n^{d-1})$ time, which gives the desired bound of $O(n^d)$.

Consider an outcome $B$. Let $p_i$ be an arbitrary point in $B$. We use $p_i$ as a reference point known to be contained in the hull $C = \mathrm{CH}(B \cup \{q\})$. Let $B'$, $p_i'$ and $q'$ denote the projections of $B$, $p_i$ and $q$ respectively on the hyperplane $x_d = 0$, which we identify with $\mathbb{R}^{d-1}$. Let us define $C' = \mathrm{CH}(B' \cup \{q'\}) \subset \mathbb{R}^{d-1}$, and let $V'$ be the vertices of $C'$.

Let $\overrightarrow{r}(p_i', q')$ denote the open ray emanating from $q'$ in the direction of the vector $\overrightarrow{p_i'q'}$ (that is, this ray is moving "away" from $p_i'$). A facet $f$ of $C$ is a *$p_i$-escaping* facet for $q$, if $q$ is a vertex of $f$ and the projection of $f$ on $\mathbb{R}^{d-1}$ intersects $\overrightarrow{r}(p_i', q')$. See Figure 2 for a three-dimensional example. The following lemma is key to our algorithm. The points of $C$ projected into $\partial C'$ form the *silhouette* of $C$.

**Lemma 3.1.** *(A) $q$ has at most one $p_i$-escaping facet on $C$.*

*(B) The point $q$ is a non-silhouette vertex of the convex-hull $C$ if and only if $q$ has a (single) $p_i$-escaping facet on $C$.*

*Proof:* (A) If $q$ has a $p_i$-escaping facet then it is a vertex of the convex-hull $C$. Consider the union of facets adjacent to $q$, and observe that the projection of this "tent" can fold over itself in the projection only if $q$ is on the silhouette. Specifically, if $q$ is not on the silhouette then the claim immediately holds.

Otherwise, $q$ is on the silhouette, the open ray $\overrightarrow{r}(p_i', q')$ does not intersect $C'$, and there are no $p_i$-escaping facets.

(B) Follows immediately from (B), by observing that in this case, the projected "tent", surrounds $q'$, and as such one of the facets must be an escaping facets for $p_i$. ∎

Given a subset of sites $P_\alpha \subseteq P \setminus \{p_i\}$ of size $(d-1)$, define $f(P_\alpha)$ to be the $(d-1)$-dimensional simplex $\mathrm{CH}(P_\alpha \cup \{q\})$. Since $p_i = \lambda(B \cup \{q\})$ implies $p_i \in B$, we can use Lemma 3.1 to decompose the $i$th term as follows:

$$\Pr\Big[\, p_i = \lambda(B \cup \{q\}) \ \wedge\ q \in V \,\Big] = \Pr\Big[\, p_i = \lambda(B \cup \{q\}) \ \wedge\ q' \in V' \,\Big] + \sum_{\substack{P_\alpha \subseteq P \setminus \{p_i\} \\ |P_\alpha| = (d-1) \\ f(P_\alpha) \text{ is } p_i\text{-escaping for } q}} \Pr\Big[\, p_i = \lambda(B \cup \{q\}) \ \wedge\ f(P_\alpha) \text{ is a facet of } C \,\Big].$$

The first term is an instance of the same problem in $(d-1)$ dimensions (for the point $q'$ and the projection of $P$), and thus is computed recursively. For the second term, we compute the probability that $f(P_\alpha)$ is a facet of $C$ as follows. Let $G_1 \subseteq P$ be the subset of sites which are on the other side of the hyperplane supporting $f(P_\alpha)$ with respect to $p_i$. Let $G_2 \subseteq P$ be the subset of sites that are below $p_i$ along the $x_d$-axis. Clearly, $f(P_\alpha)$ is a facet of $C$ (and $p_i = \lambda(B \cup \{q\})$) if and only if all points in $P_\alpha$ and $p_i$ exist in $B$, and all points in $G_1 \cup G_2$ are absent from $B$. The corresponding probability can be written as

$$\gamma_i \times \prod_{p_j \in P_\alpha} \gamma_j \times \prod_{p_j \,\in\, G_1 \cup G_2} \overline{\gamma_j}\,.$$

This formula is valid only if $P_\alpha \cap G_2 = \emptyset$ and $p_i$ has a lower $x_d$-coordinate than $q$; otherwise we set the probability to zero. This expression can be computed in linear time, and the whole summation term can be computed in $O(n^d)$ time. Then, by induction, the computation of the $i$th term takes $O(n^d)$ time. Notice that the base case of our induction requires computing the probability $\Pr\big[\, p_i = \lambda(B \cup \{q\}) \ \wedge\ q^{(d-2)} \in V^{(d-2)} \,\big]$ (where ${}^{(d-2)}$ indicates a projection to $\mathbb{R}^2$). Computing this probability is essentially a two-dimensional membership probability problem on $q$ and $P$, but is conditioned on the existence of $p_i$ and the non-existence of all sites below $p_i$ along $d$th axis. Our two dimensional algorithm can be easily adapted to solve this variation in $O(n \log n)$ time as well.

Similar to the planar case, we can improve the computation time for the $i$th term to $O(n^{d-1})$ by considering the facets $f(P_\alpha)$ in radial order. In essence, the main idea is to fix $(d-2)$ sites of a given $P_\alpha$ and then change its $(d-1)$th site in radial order around the fixed sites in order to obtain different $P_\alpha$s, such that the probability for the next $P_\alpha$ is easily computed using the probability for the previous $P_\alpha$. In particular, let $L_\beta \subseteq P \setminus \{p_i\}$ be a subset of $(d-2)$ sites. Let $f_j$ denote the $(d-1)$-dimensional simplex $f(L_\beta \cup \{p_j\})$ where $p_j \notin L_\beta$ and $p_j \neq p_i$. (Note that $L_\beta \cup p_j$ is a possible $P_\alpha$ and $f_j = f(P_\alpha)$.) We can compute the probability that $f_j$ is a facet of $C$ for all facets $f_j$

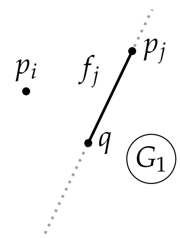


Figure 3: A facet $f_j$ projected to the orthogonal complement plane.

in constant amortized time as follows. We project all sites to the two-dimensional plane passing through $q$ and orthogonal to the $(d-2)$-dimensional hyperplane defined by $L_\beta \cup \{q\}$. (Such a plane is known as an orthogonal complement.) The hyperplane defined by $L_\beta \cup \{q\}$ projects onto $q$ on this plane. Moreover, each facet $f_j$ projects to a line segment extending from $q$. When we need to compute the probability that $f_j$ is a facet of $C$, the set $G_1$ includes the sites on the other side of the line supporting $f_j$'s projection with respect to $p_i$. (See Figure 3.) We compute probabilities for the facets $f_j$ based on their radial order around $q$. The probability for the next facet in the sweep can be computed by modifying the probability of the previous facet in constant amortized time as we have done for the planar case, as we can efficiently track how $G_1$ changes. It follows that the probability for all facets $f_j$ (based on a single $L_\beta$) can be computed in $O(n)$ time. By iterating through all possible $L_\beta$, we can compute the probability for all facets $f(P_\alpha)$ in $O(n^{d-1})$ time. As a final note, we point out that the total cost of all sorting required for the radial sweeps is $O(n^{d-1}\log n)$ which is less than the overall cost of $O(n^d)$.

**Theorem 3.2.** *Let $\mathcal{P}$ be a set of $n$ uncertain points in the unipoint model in $\mathbb{R}^d$, and let $q$ be a point in $\mathbb{R}^d$. If the input sites and $q$ are in general position, then the probability of $q$ being in the convex hull of $\mathcal{P}$ can be computed in $O(n^d)$ time using linear space.*

### 3.2. The Multipoint Model

As in the planar case, the $d$-dimensional algorithm easily extends to the multipoint model. As before, we compute $\mu(q)$ by computing the probability $\Pr[\,q \in V\,]$. Following the earlier strategy, we decompose it as

$$\Pr\Big[\,q \in V\,\Big] = \Pr\Big[\,q = \lambda(B)\,\Big] + \sum_{1\le i\le m}\left(\sum_{1\le j\le |P_i|}\Pr\Big[\,p_i^j = \lambda(B)\ \wedge\ q \in V\,\Big]\right).$$

It is trivial to compute the first term in $O(n)$ time. We now show how to compute each term inside the summations in $O(n^{d-1})$ time. This implies a total time of $O(n^d)$.

Clearly, Lemma 3.1 extends to the multipoint model, so we can use $p_i^j$-escaping facets to decompose our probability. Given a subset of sites $P_\alpha \subseteq P\backslash\left\{p_i^j\right\}$ of size $(d-1)$, define $f(P_\alpha)$ to be the $(d-1)$-dimensional simplex whose vertices are the points in $P_\alpha$ and $q$. Then,

$$\Pr\Big[ p_i^j = \lambda(B \cup \{q\}) \ \wedge \ q \in V \Big] = \Pr\Big[ p_i^j = \lambda(B \cup \{q\}) \ \wedge \ q' \in V' \Big]$$
$$+ \sum_{\substack{P_\alpha \subseteq P \setminus \left\{p_i^j\right\} \\ |P_\alpha| = (d-1) \\ f(P_\alpha) \text{ is } p_i^j\text{-escaping for } q}} \Pr\Big[ p_i^j = \lambda(B \cup \{q\}) \ \wedge \ f(P_\alpha) \text{ is a facet of } C \Big].$$

The first term is computed recursively. We compute each term of the summation as follows. Let $I_\alpha$ be the set of uncertain point indices of the sites in $P_\alpha$, i.e., $I_\alpha = \{u \mid \exists v \,.\, p_u^v \in P_\alpha\}$. As before, let $G_1 \subseteq P$ be the subset of sites which are on the other side of the hyperplane supporting $f(P_\alpha)$ with respect to $p_i^j$. Let $G_2 \subseteq P$ be the subset of sites that are below $p_i^j$ along the $x_d$-axis. As done for the unipoint model, we write the desired probability as the probability that all points in $P_\alpha$ and $p_i^j$ exist in $B$, and all points in $G_1 \cup G_2$ are absent from $B$. This probability is clearly zero, if any of the following conditions hold:

- $P_\alpha \cap G_2 \neq \emptyset$.
- $p_i^j$ has a higher $x_d$-coordinate than $q$.
- $P_\alpha$ contains any two sites from the same uncertain point $P_k$.
- $P_\alpha$ contains any site from $P_i$.

Otherwise, we can write the probability as follows:

$$\begin{aligned}
&\Pr\Big[ p_i^j \in B \ \wedge \ P_\alpha \subseteq B \ \wedge \ B \cap (G_1 \cup G_2) = \emptyset \Big] \\
&= \Pr\Big[ p_i^j \in B \Big] \times \Pr\Big[ P_\alpha \subseteq B \mid p_i^j \in B \Big] \times \\
&\qquad \Pr\Big[ B \cap (G_1 \cup G_2) = \emptyset \mid p_i^j \in B \ \wedge \ P_\alpha \subseteq B \Big] \\
&= \Pr\Big[ p_i^j \in B \Big] \times \Pr\Big[ P_\alpha \subseteq B \Big] \times \\
&\qquad \Pr\Big[ B \cap (G_1 \cup G_2) = \emptyset \mid p_i^j \in B \ \wedge \ P_\alpha \subseteq B \Big] \\
&= \Pr\Big[ p_i^j \in B \Big] \times \Pr\Big[ P_\alpha \subseteq B \Big] \times \\
&\qquad \prod_{\substack{1 \le u \le m \\ u \neq i \\ u \notin I_\alpha}} \Big( \Pr[\, P_u \cap B \cap (G_1 \cup G_2) = \emptyset \,] \Big) \\
&= \gamma_i^j \times \prod_{u,v \,\mid\, p_u^v \in P_\alpha} \gamma_u^v \times \prod_{\substack{1 \le u \le m \\ u \neq i \\ u \notin I_\alpha}} \left( 1 - \sum_{v \,\mid\, p_u^v \in (G_1 \cup G_2)} \gamma_u^v \right).
\end{aligned}$$

The expression takes linear time to compute and thus the summation term can be computed in $O(n^d)$ time. Then, by induction, the computation of the term for the site $p_i^j$ takes $O(n^d)$ time. As

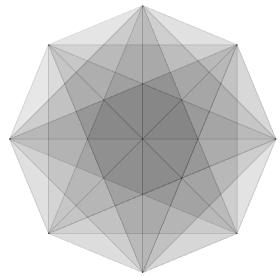

Figure 4: The probability map for a set of uncertain points arranged as vertices of a regular polygon.

before, we can improve this computation time to $O(n^{d-1})$ by considering the facets $f(P_\alpha)$ in radial order. This implies a total complexity of $O(n^d)$ for the algorithm.

**Theorem 3.3.** *Let $\mathcal{P}$ be a set of $n$ uncertain points in the multipoint model in $\mathbb{R}^d$ and a point $q \in \mathbb{R}^d$ for $d \geq 3$, such that $q$ and the sites of $\mathcal{P}$ are in general position, the probability of $q$ being in the convex hull of $\mathcal{P}$ can be computed in $O(n^d)$ time.*

# 4. Membership Queries

In this section, we describe two algorithms – one deterministic and one Monte Carlo – for preprocessing a set of uncertain points for efficient membership-probability queries. We begin with the deterministic algorithm, which is based on a structure called the probability map.

## 4.1. Probability Map

The *probability map* $\mathbb{M}(\mathcal{P})$ is the subdivision of $\mathbb{R}^d$ into maximal connected regions so that the membership probability is the same for all query points in a region. The following lemma gives a tight bound on the size of $\mathbb{M}(\mathcal{P})$.

**Lemma 4.1.** *The worst-case complexity of the probability map of a set of uncertain points in $\mathbb{R}^d$ is $\Theta(n^{d^2})$, under both the unipoint and the multipoint model, where $n$ is the total number of sites in the input.*

*Proof:* We prove the result for the unipoint model, as the extension to the multipoint model is straightforward. For the upper bound, consider the set $H$ of $O(n^d)$ hyperplanes formed by all $d$-tuples of points in $\mathcal{P}$. In the arrangement $\mathcal{A}(H)$ formed by these planes, each (open) cell has the same value of $\mu(q)$. This arrangement, which is a refinement of $\mathbb{M}(\mathcal{P})$, has size $O((n^d)^d) = O(n^{d^2})$, establishing the upper bound.

For the lower bound, consider the problem in two dimensions; extension to higher dimensions is straightforward. We choose the sites to be the vertices $p_1, \ldots, p_n$ of a regular $n$-gon, where each site exists with probability $\gamma$, $0 < \gamma < 1$. See Figure 4. Consider the arrangement $\mathcal{A}$ formed by the line segments $p_i p_j$, $1 \leq i < j \leq n$, and treat each face as relatively open. If $\mu(f)$ denotes the membership probability for a face $f$ of $\mathcal{A}$, then for any two faces $f_1$ and $f_2$ of $\mathcal{A}$, where $f_1$ bounds $f_2$ (i.e., $f_1 \subset \partial f_2$), we have $\mu(f_1) \geq \mu(f_2)$, and $\mu(f_1) > \mu(f_2)$ if $\gamma < 1$. Thus, the size of the arrangement $\mathcal{A}$ is also a lower bound on the complexity of $\mathbb{M}(\mathcal{P})$. This proves that the worst-case complexity of $\mathbb{M}(\mathcal{P})$ in $\mathbb{R}^d$ is $\Theta(n^{d^2})$. ∎

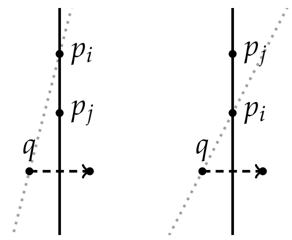


Figure 5: The cases to consider for computing the probability of $C'$ from $C$.

We now describe an algorithm for computing the probability map $\mathbb{M}(\mathcal{P})$. For simplicity, we describe the algorithm for the unipoint model, and then briefly explain how to extend it to the multipoint model. Let $H$ be the set of $O(n^2)$ lines passing through a pair of sites in $\mathcal{P}$, and let $\mathcal{A}(H)$ be the arrangement of $H$. $\mathcal{A}(H)$ contains $O(n^4)$ cells, edges and vertices. By Lemma 4.1, $\mathcal{A}(H)$ is a refinement of $\mathbb{M}(\mathcal{P})$. We first construct $\mathcal{A}(H)$ in $O(n^4)$ time, using an algorithm by Edelsbrunner et al. [15]. Since the membership probability of all points on an edge or cell is the same, let $\mu(f)$ denote this quantity for a vertex, edge, or cell $f$ of $\mathcal{A}(H)$.

Next, we compute the membership probability of one of the cells in the arrangement, say $C$, in $O(n \log n)$ time (cf. Theorem 2.1). We compute the membership probabilities of the vertices, edges and cells neighboring $C$, in $O(1)$ time per each, by modifying $\mu(C)$.[2] We then apply the same process for all cells neighboring $C$. By repeatedly expanding into the neighboring cells, we can compute the probabilities for all of the arrangement in $O(n^4)$ time.

We now show how to compute $\mu(C')$ by using the already computed $\mu(C)$, where $C'$ is one of the neighboring cells of $C$. We later explain how this algorithm can be adapted to compute the probability of neighboring edges and vertices.

Without loss of generality, assume that $C$ and $C'$ are separated by a vertical line $\ell$ passing through the sites $p_i$ and $p_j$ and $C$ is to the left of $C'$. Notice that the common edge of $C$ and $C'$, contained in $\ell$, does not contain $p_i$ or $p_j$. Now imagine that a point $q$ moves through this boundary, crossing from $C$ to $C'$. It is easy to see that the change in the membership probability of $q$ is due to the changes in witness edge probabilities of the segments $qp_i$ and $qp_j$, as other sites are irrelevant. Let $G_i(C)$ denote the set of sites lying to the right of the line $\overrightarrow{qp_i}$ for some $q \in C$. By construction, $G_i(C)$ is the same for all $q \in C$. Similarly we define $G_i(C')$, $\pi_i(C) = \Pr\left[\, qp_i \text{ is a witness edge } \mid q \in C \,\right]$ and $\pi_i(C') = \Pr\left[\, qp_i \text{ is a witness edge } \mid q \in C' \,\right]$. We describe the change in the witness edge probability of $qp_i$. The probability of $qp_j$ changes analogously. The change in the probability of $qp_i$ happens differently for two cases (See Figure 5):

(a) **$p_i$ is above $p_j$:** In this case $G_i(C') = G_i(C) \setminus \{p_j\}$, therefore $\pi_i(C') = \pi_i(C)/\overline{\gamma_j}$.

(b) **$p_i$ is below $p_j$:** In this case $G_i(C') = G_i(C) \cup \{p_j\}$ and $p_j \notin G_i(C)$, therefore $\pi_i(C') = \pi_i(C) \cdot \overline{\gamma_j}$.

The changes clearly require constant time operations, and thus the membership probability of $C'$ can be computed in $O(1)$ time.

[2]For ease of presentation, we assume that the arrangement is non-degenerate. It is straightforward to apply our technique on degenerate arrangements by using standard techniques (such as perturbation) to create a non-degenerate arrangement. We note that, even if we perturb the points to create a non-degenerate arrangement, we still use the old coordinates of the points and utilize the degeneracy handling rules of Section 2.3 while computing probabilities.

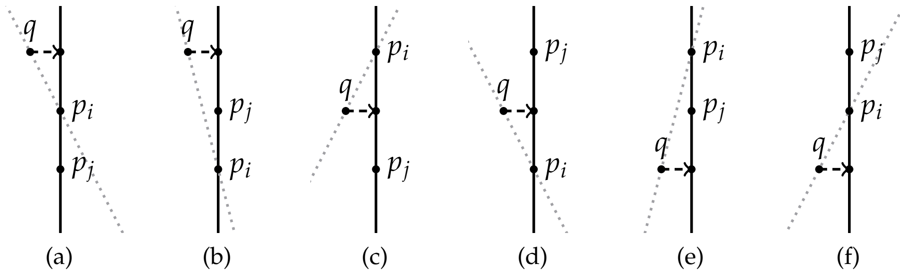


Figure 6: The cases to consider for computing the probability of $e$ from $C$.

We now describe how to compute $\mu(e)$ by using the already computed $\mu(C)$, where $e$ is one of the bounding edges of $C$. Notice that any query point $q$ on an edge $e$ is degenerate with respect to the uncertain points. Therefore, we have to make use of the degeneracy handling rules from Section 2.3. Without loss of generality, assume that $e$ is on a vertical line passing through the sites $p_i$ and $p_j$ and C is to the left of $e$. Notice that $e$ is only a segment of the vertical line and does not contain $p_i$ or $p_j$. Now imagine that a point $q$ moves from $C$ onto $e$. Again, the change in the membership probability of $q$ is due to the changes in witness edge probabilities of the segments $qp_i$ and $qp_j$. Similarly to $G_i(C)$ and $\pi_i(C)$, we define $G_i(e)$, and $\pi_i(e) = \Pr\big[\, qp_i \text{ is a witness edge } \mid q \in e \,\big]$. We describe the change in the witness edge probability of $qp_i$, the change for $qp_j$ is analogous. We consider six different cases based on the vertical order of the points $q$,$p_i$ and $p_j$ (See Figure 6):

(a) **Order** $q, p_i, p_j$**:** In this case $G_i(e) = G_i(C)$, therefore $\pi_i(e) = \pi_i(C)$.

(b) **Order** $q, p_j, p_i$**:** In this case $G_i(e) = G_i(C)$, therefore $\pi_i(e) = \pi_i(C)$.

(c) **Order** $p_i, q, p_j$**:** In this case $G_i(e) = G_i(C)$, therefore $\pi_i(e) = \pi_i(C)$.

(d) **Order** $p_j, q, p_i$**:** In this case $G_i(e) = G_i(C) \cup \{p_j\}$ and $p_j \notin G_i(C)$, therefore $\pi_i(e) = \pi_i(C) \cdot \overline{\gamma_j}$.

(e) **Order** $p_i, p_j, q$**:** In this case $G_i(e) = G_i(C) \setminus \{p_j\}$, therefore $\pi_i(e) = \pi_i(C)/\overline{\gamma_j}$.

(f) **Order** $p_j, p_i, q$**:** In this case $G_i(e) = G_i(C) \cup \{p_j\}$ and $p_j \notin G_i(C)$, therefore $\pi_i(e) = \pi_i(C) \cdot \overline{\gamma_j}$.

For all cases, the change to the witness edge probability is easily computed in $O(1)$ time.

Finally, we explain how to compute the probability of the vertices. Let $v$ be a vertex of $\mathcal{A}(H)$, which is the common endpoint of two edges $e_1$ and $e_2$ of a cell $C$. Then $\mu(v)$ is computed by applying the same probability changes that is applied for $e_1$ and $e_2$. In other words,

$$\mu(v) = \frac{\mu(e_1) \cdot \mu(e_2)}{\mu(C)}.$$

The only exception to this is when $v$ coincides a site $p_i$. In that case, we compute the membership probability of $v$ from scratch in $O(n \log n)$ time. Since the number of such vertices is linear, it does not increase our overall cost of $O(n^4)$.

The extension of our technique to the multipoint model is straightforward. The only major difference is that we need to remember (similar to what is done in Section 2.2) the intermediate factors when computing cell probabilities, as updating the witness edge probabilities requires updating these factors first. The total cost of a single update remains $O(1)$ because it requires updating one intermediate factor of two witness edge probabilities.

**Theorem 4.2.** *Given a set $\mathcal{P}$ of uncertain points in the plane with $n$ sites in total, the probability map $\mathbb{M}(\mathcal{P})$ can be computed in $O(n^4)$ time.*

Once $\mathbb{M}(\mathcal{P})$ is computed, it can be preprocessed in $O(n^4)$ time into a data structure of size $O(n^4)$ so that the vertex, edge, or face of $\mathbb{M}(\mathcal{P})$ containing a query point can be found in $O(\log n)$ time. We thus conclude the following.

**Theorem 4.3.** *Let $\mathcal{P}$ be a set of uncertain points in $\mathbb{R}^2$, with a total of $n$ sites. $\mathcal{P}$ can be preprocessed in $O(n^4)$ time into a data structure of size $O(n^4)$ so that for any point $q \in \mathbb{R}^d$, $\mu(q)$ can be computed in $O(\log n)$ time.*

**Remark.** For $d \geq 3$, due to our general position assumption, we can compute the membership probability only for $d$-faces of $\mathbb{M}(\mathcal{P})$, and not for the lower-dimensional faces. In that case, by utilizing a point-location technique in [9], one can build a structure that can report the membership probability of a query point (inside a $d$-face) in $O(\log n)$ time, with a preprocessing cost of $O(n^{d^2+d})$.

## 4.2. Monte Carlo Algorithm

The size of the probability map may be prohibitive even for $d = 2$, so we describe a simple, space-efficient Monte Carlo approach for quickly approximating the membership probability, within absolute error. Fix a parameter $s > 1$, to be specified later. The preprocessing consists of $s$ rounds, where the algorithm creates an outcome $A_j$ of $\mathcal{P}$ in each round $j$. Each $A_j$ is preprocessed into a data structure so that for a query point $q \in \mathbb{R}^d$, we can determine whether $q \in \mathrm{CH}\big(A_j\big)$.

For $d \leq 3$, we can build each $\mathrm{CH}\big(A_j\big)$ explicitly and use linear-size point-location structures with $O(\log n)$ query time. For $d = 2$, we also apply fractional cascading to $\mathrm{CH}\big(A_j\big)$, for $1 \leq j \leq s$, so that for a point $q \in \mathbb{R}^2$, all values of $j \leq s$ for which $q \in \mathrm{CH}\big(A_j\big)$ can be reported in a total of $O(\log n + s)$ time. This leads to total preprocessing time $O(sn \log n)$ and space $O(sn)$. For $d \geq 4$, we use the data structure in [25] for determining whether $q \in \mathrm{CH}\big(A_j\big)$, for all $1 \leq j \leq s$. For a parameter $t$ such that $n \leq t \leq n^{\lfloor d/2 \rfloor}$ and for any constant $\sigma > 0$, using $O(st^{1+\sigma})$ space and preprocessing, it can compute in $O(\frac{sn}{t^{1/\lfloor d/2 \rfloor}} \log^{2d+1} n)$ time whether $q \in \mathrm{CH}\big(A_j\big)$ for every $j$.

Given a query point $q \in \mathbb{R}^d$, we check whether $q \in \mathrm{CH}\big(A_j\big)$, for every $j \leq s$. If $q$ lies in $k$ of them, we return $\widehat{\mu}(q) = k/s$ as our estimate of $\mu(q)$. Thus, the query time is $O(\frac{sn}{t^{1/\lfloor d/2 \rfloor}} \log^{2d+1} n)$ for $d \geq 4$, $O(s \log n)$ for $d = 3$, and $O(\log n + s)$ for $d = 2$.

It remains to determine the value of $s$ so that $|\mu(q) - \widehat{\mu}(q)| \leq \varepsilon$ for all queries $q$, with probability at least $1 - \delta$. For a fixed $q$ and outcome $A_j$, let $X_i$ be the random indicator variable, which is 1 if $q \in \mathrm{CH}\big(A_j\big)$ and 0 otherwise. Since $\mathsf{E}[X_i] = \mu(q)$ and $X_i \in \{0, 1\}$, using a Chernoff-Hoeffding bound [26] on

$$\widehat{\mu}(q) = k/s = (1/s) \sum_i X_i,$$

we observe that

$$\Pr[\, |\widehat{\mu}(q) - \mu(q)| \geq \varepsilon \,] \leq 2 \exp(-2\varepsilon^2 s) \leq \delta'.$$

By Lemma 4.1, we need to consider $O(n^{d^2})$ distinct queries. If we set $1/\delta' = O(n^{d^2}/\delta)$ and $s = O((1/\varepsilon^2)\log(n/\delta))$, we obtain the following theorem.

**Theorem 4.4.** *Let $\mathcal{P}$ be a set of uncertain points in $\mathbb{R}^d$ under the multipoint model with a total of $n$ sites, and let $\varepsilon, \delta \in (0,1)$ be parameters. $\mathcal{P}$ can be preprocessed into a data structure so that with probability at least $1-\delta$, for any query point $q \in \mathbb{R}^2$, $\widehat{\mu}(q)$ satisfying $|\mu(q) - \widehat{\mu}(q)| \le \varepsilon$ and $\widehat{\mu}(q) > 0$ can be returned.*

- *For $d = 2$, the preprocessing time, the size, and the query time of the data structure are $O(\frac{n}{\varepsilon^2}\log\log\frac{n}{\delta}\log n)$, $O(\frac{n}{\varepsilon^2}\log\frac{n}{\delta})$, and $O(\frac{1}{\varepsilon^2}\log\frac{n}{\delta})$, respectively.*
- *For $d = 3$, the preprocessing time, the size, and the query time of the data structure are $O(\frac{n}{\varepsilon^2}\log\log\frac{n}{\delta}\log n)$, $O(\frac{n}{\varepsilon^2}\log\frac{n}{\delta})$, and $O(\frac{1}{\varepsilon^2}\log(\frac{n}{\delta})\log n)$, respectively.*
- *For $d \ge 4$, the preprocessing time, the size, and the query time of the data structure are $O((t^{1+\sigma}/\varepsilon^2)\log\frac{n}{\delta})$, $O((t^{1+\sigma}/\varepsilon^2)\log\frac{n}{\delta})$, and $O(\frac{n}{t^{1/\lfloor d/2\rfloor}\varepsilon^2}\log\frac{n}{\delta}\log^{2d+1} n)$, respectively, where $t$ is a parameter and $n \le t \le n^{\lfloor d/2\rfloor}$, for any constant $\sigma > 0$.*

## 5. Tukey Depth and Convex Hull

The membership probability is neither a convex nor a continuous function, as suggested by the example in the proof of Lemma 4.1. In this section, we establish a helpful structural property of membership-probability function, intuitively showing that the probability stabilizes once we go deep enough into the "region". Specifically, we show a connection between the Tukey depth of a point $q$ with its membership probability; in two dimensions, this also results in an efficient data structure for approximating $\mu(q)$ quickly within a small absolute error.

**Estimating $\mu(q)$.** Let $Q$ be a set of weighted points in $\mathbb{R}^d$. For a subset $A \subseteq Q$, let $w(A)$ be the total weight of points in $A$. Then the *Tukey depth* of a point $q \in \mathbb{R}^d$ with respect to $Q$, denoted by $\tau(q, Q)$, is $\min w(Q \cap H)$ where the minimum is taken over all halfspaces $H$ that contain $q$.[3] If $Q$ is obvious from the context, we use $\tau(q)$ to denote $\tau(q, Q)$. Before bounding $\mu(q)$ in terms of $\tau(q, Q)$, we prove the following lemma.

**Lemma 5.1.** *Let $Q$ be a finite set of points in $\mathbb{R}^d$. For any $p \in \mathbb{R}^d$, there is a set $\mathcal{S} = \{S_1, \dots, S_T\}$ of $d$-simplices formed by $Q$ such that (i) each $S_i$ contains $p$ in its interior; (ii) no pair of them shares a vertex; and (iii) $T \ge \lceil \tau(p, Q)/d \rceil$.*

*Proof:* If $\tau(p, Q) > 0$, then $p \in \mathrm{CH}(Q)$, and by Carathéodory Theorem [14], there is a $d$-simplex $S$ with its $d+1$ vertices in $Q$ such that $p \in S$. Remove the vertices of $S$ from $Q$, and repeat the argument. Let $S_1, \dots, S_T$ be the resulting simplices. Observe that at most $d$ vertices of $S$ can be in an halfspace passing through $p$, which implies that the Tukey depth of $p$ drops by at most $d$ after each iteration of this algorithm. Hence $T \ge \lceil \tau(p, Q)/d \rceil$. ∎

We now use Lemma 5.1 to bound $\mu(p)$ in terms of $\tau(p, P)$, but we first need a definition. Let $X$ be a set of $n$ points in $\mathbb{R}^d$. A subset $N \subseteq X$ is called an *$\varepsilon$-net*, for $\varepsilon \in [0,1]$, if for every halfspace

[3]If the points in $Q$ are unweighted, then $\tau(q, Q)$ is simply the minimum number of points that lie in a closed halfspace that contains $q$.

$h$ with $|h \cap X| \geq \varepsilon n$, $N \cap h \neq \emptyset$. Haussler and Welz [18] proved that a random subset of $X$ of size $\frac{cd}{\varepsilon} \ln \frac{1}{\varepsilon\delta}$ is an $\varepsilon$-net with probability at least $1 - \delta$; here $c$ is a constant.[4] Their argument can be adapted to prove that if each point of $X$ is chosen with probability $p(\varepsilon) \geq \frac{cd}{\varepsilon n} \ln \frac{1}{\varepsilon\delta}$, then the resulting subset is an $\varepsilon$-net with probability at least $1 - \delta$.

**Theorem 5.2.** *Let $\mathcal{P}$ be a set of $n$ uncertain points in the uniform unipoint model, that is, each point is chosen with the same probability $\gamma > 0$. Let $P$ be the set of sites in $\mathcal{P}$. There is a constant $c > 0$ such that for any point $p \in \mathbb{R}^d$ with $\tau(p, P) \geq t$, we have $(1-\gamma)^t \leq 1 - \mu(p) \leq d \exp\left(-\frac{\gamma t}{cd^2}\right)$.*

*Proof:* For the first inequality, fix a closed halfspace $H$ that contains $t$ points of $P$. If none of these $t$ points is chosen then $p$ does not appear in the convex hull of the outcome, so $1 - \mu(p) \geq (1-\gamma)^t$.

Next, let $p$ be a point with $\tau(p, P) \geq t$, $\mathcal{S}$ be the set of simplices of Lemma 5.1, and $V$ be its set of vertices, where $T \geq \lceil t/d \rceil$. The uniform model chooses each point of $V$ with probability $\gamma$. Let $R \subseteq V$ be the set of chosen points. If $\gamma = \frac{cd(d+1)}{|V|} \ln \frac{d}{\delta}$, then $R$ is a $(\frac{1}{d+1})$-net with probability at least $1 - \delta$. Since any halfspace $H$ containing $p$ contains at least one vertex of each simplex in $\mathcal{S}$, $|H \cap R| \geq \frac{|V|}{d+1}$. Therefore with probability at least $1 - \delta$, every halfspace containing $p$ contains at least one point of $R$. Consequently, $\mu(p) \geq 1 - \delta$. Since $\gamma = \frac{cd(d+1)}{|V|} \ln \frac{d}{\delta} = \frac{cd}{T} \ln \frac{d}{\delta} \leq \frac{cd^2}{t} \ln \frac{d}{\delta}$, $\delta \leq d \exp(-\frac{\gamma t}{cd^2})$, as desired.

**Data structure.** Let $\mathcal{P}$ be a set of uncertain points in the uniform unipoint model in $\mathbb{R}^2$, i.e., each point appears with probability $\gamma$. Let $P$ denote the set of all sites of $\mathcal{P}$. We now describe a data structure to estimate $\mu(q)$ for a query point $q \in \mathbb{R}^2$, within additive error $1/n$. We fix a parameter $t_0 = \frac{c}{\gamma} \ln n$ for some constant $c > 0$. Let $\mathcal{T} = \left\{x \in \mathbb{R}^2 \mid \tau(x, \mathcal{P}) \geq t_0\right\}$ be the set of all points whose Tukey depth in $\mathcal{P}$ is at least $t_0$. If $t_0 \leq n/3$, $\mathcal{T} \neq \emptyset$ and $\mathcal{T}$ is a convex polygon with $O(n)$ vertices [24]. We assume that $\gamma \geq \frac{3c}{n} \ln n$, so that $\mathcal{T} \neq \emptyset$.

By Theorem 5.2, $\mu(q) \geq 1 - 1/n^{c'}$ for all points $q \in \mathcal{T}$, where $c'$ is a constant dependent on $c$. For a point $q \in \mathbb{R}^2$, let $\hat{\mu}(q) = \Pr\left[q \in \mathrm{CH}(\mathcal{T} \cup \mathsf{R})\right]$, where $\mathsf{R}$, as earlier, is the random subset of $\mathcal{P}$ obtained by choosing each point of $\mathcal{P}$ with probability $\gamma$. Note that $\hat{\mu}(q) \geq \mu(q)$. We describe a data structure for computing $\hat{\mu}(q)$ and argue that $\hat{\mu}(q) - \mu(q) \leq 1/n^{c'}$ for some constant $c'$ that depends on $c$.

We construct $\mathcal{T}$ and preprocess $P$ for halfspace range reporting queries [11]. $\mathcal{T}$ can be computed in time $O(n \log^3 n)$ [24], and constructing the half-plane range reporting data structure takes $O(n \log n)$ time [11]. So the total preprocessing time is $O(n \log^3 n)$, and the size of the data structure is linear.

A query is answered as follows. Given a query point $q \in \mathbb{R}^2$, we first test in $O(\log n)$ time whether $q \in \mathcal{T}$. If the answer is yes, we simply return 1 as $\mu(q)$. If not, we compute in $O(\log n)$ time the two tangents $\ell_1, \ell_2$ of $\mathcal{T}$ from $q$. For $i = 1, 2$, let $\xi_i = \ell_i \cap \mathcal{T}$, and let $h_{\ell_i}$ be the *closed* half-plane bounded by $\ell_i$ that does not contain $\mathcal{T}$. Without loss of generality, assume that $\mathcal{T}$ lies to the left (resp. right) of the vector $\overrightarrow{q\xi_1}$ (resp. $\overrightarrow{q\xi_2}$). (See Figure 7.) Set $P_q = P \cap (h_{\ell_1} \cup h_{\ell_2})$ and $n_q = |P_q|$.

By querying the half-plane range reporting data structure with each of these two tangent lines, we compute the set $P_q$ in time $O(\log n + n_q)$. Let $P_q^0 = P_q \cap (h_{\ell_1} \setminus h_{\ell_2})$. Note that for a point

[4] Haussler and Welz [18] defined $\varepsilon$-net for general range spaces and proved the bound on the size of $\varepsilon$-nets for range spaces with finite VC-dimension but we need their result for this special case.

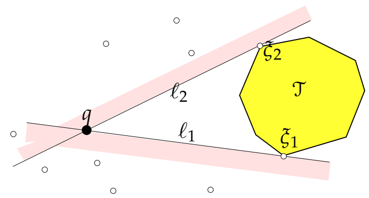


Figure 7: Illustration for the halfplanes $h_{\ell_1}$ and $h_{\ell_2}$.

$p \in P \setminus P_q^0$, $\overrightarrow{qp}$ cannot be a witness edge of $q \notin \text{CH}(\mathsf{R} \cup \mathcal{T})$. Adapting Eq. (2) in Section 2, we can write

$$1 - \hat{\mu}(q) = \sum_{p_i \in P_q^0} \hat{\pi}_i(q). \tag{3}$$

where $\hat{\pi}_i(q) = \Pr\left[\, qp_i \text{ is a witness edge} \,\right]$.

Note that for any $p_i \in P_q^0$, the points lying to the right of the line $\overrightarrow{qp_i}$ belong to $P_q$. Therefore $\hat{\pi}_i(q)$ can be computed by just considering the points of $P_q$. By sorting $P_q$ around $q$ and then performing an angular sweep, as in Section 2, $\hat{\pi}_i(q)$, for all $p_i \in P_q$, can be computed in a total of $O(n_q \log n_q)$ time. Hence, $\hat{\mu}(q)$ can be computed in $O(n_q \log n_q)$ time.

The correctness of the algorithm follows from the following lemma.

**Lemma 5.3.** *For any point $q \notin \mathcal{T}$, $|\hat{\mu}(q) - \mu(q)| \leq 1/n^{c-1}$, where $c$ is the hiding constant of proportionality in to.*

*Proof:* We first note that $\pi_i(q) = \hat{\pi}_i(q)$ for all $p_i \in P_q^0$. Therefore Eq. (2) and Eq. (3) imply

$$\hat{\mu}(q) - \mu(q) = \sum_{p_i \notin P_q^0} \pi_i(q) = \sum_{p_i \notin P_q^0} \gamma(1-\gamma)|G_i|,$$

where $G_i \subseteq P$, as defined in Section 2, is the set of points lying to the right of $\overrightarrow{qp_i}$.

For $p_i \notin P_q^0$, the halfplane lying to the right of the line $\overrightarrow{qp_i}$ intersects $\mathcal{T}$, therefore, by definition of $\mathcal{T}$, $|G_i| \geq t_0$. Hence,

$$\begin{aligned} \hat{\mu}(q) - \mu(q) &\leq \sum_{p_i} \gamma(1-\gamma)^{t_0} \\ &\leq n \cdot \exp(-\gamma t_0) \\ &\leq n \cdot \exp(-\gamma \frac{c}{\gamma} \ln n) \\ &= 1/n^{c-1}. \end{aligned}$$

■

The efficiency of the algorithm follows from the following lemma.

**Lemma 5.4.** *For any point $q \notin \mathcal{T}$, $n_q < 4t_0 = O(\gamma^{-1} \log n)$.*

*Proof:* Fix any edge $e = (u, v)$ of $\mathcal{T}$. Let $\ell_e$ be the bounding line of $e$, and $h_{\ell_e}^-$ be the *open* half-plane bounded by $\ell_e$ that does not contain $\mathcal{T}$. By definition of $\mathcal{T}$, $|P \cap h_{\ell_e}^-| < t_0$.

Next we show that $|P \cap h_{\ell_1}| < 2t_0$. Let $e_1 = (\xi_1, u)$ and $e'_1 = (\xi_1, u')$ be the two edges adjacent to the vertex $\xi_1$ of $\mathcal{T}$. Then $|P \cap h^-_{\ell_{e_1}}| \leq t_0 - 1$ and $|P \cap h^-_{\ell_{e'_1}}| \leq t_0 - 1$. Noticing that $P \cap h_{\ell_1} \subseteq P \cap (h^-_{\ell_{e_1}} \cup h^-_{\ell_{e'_1}} \cup \{\xi_1\})$, we have

$$|P \cap h_{\ell_1}| \leq |P \cap (h^-_{\ell_{e_1}} \cup h^-_{\ell_{e'_1}} \cup \{\xi_1\})| \leq |P \cap h^-_{\ell_{e_1}}| + |P \cap h^-_{\ell_{e'_1}}| + 1 < 2t_0.$$

Similarly, $|P \cap h_{\ell_2}| < 2t_0$. The lemma now follows from that $n_q = |P \cap (h_{\ell_1} \cup h_{\ell_2})| \leq |P \cap h_{\ell_1}| + |P \cap h_{\ell_2}| < 4t_0$.

By Lemma 5.4, $n_q = O(\gamma^{-1} \log n)$, so the query takes $O(\gamma^{-1} \log(n) \log\log n)$ time. We thus obtain the following.

**Theorem 5.5.** *Let $\mathcal{P}$ be a set of $n$ uncertain points in $\mathbb{R}^2$ in the unipoint model, where each point appears with probability $\gamma$. Given a constant $c > 0$, $\mathcal{P}$ can be preprocessed in $O(n \log^3 n)$ time into a linear-size data structure so that, for any point $q \in \mathbb{R}^2$, a value $\hat{\mu}(q)$ satisfying $|\hat{\mu}(q) - \mu(q)| \leq 1/n^c$ can be computed in $O(\gamma^{-1} \log(n) \log\log n)$ time, provided that $\gamma = \Omega(\frac{\ln n}{n})$ with the hiding constant of proportionality depending on $c$.*

# 6. $\beta$-Hull

In this section, we consider the multipoint model, i.e., $\mathcal{P} = \{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$. A convex set $C \subseteq \mathbb{R}^2$ is called *$\beta$-dense* with respect to $\mathcal{P}$ if it contains $\beta$-fraction of each $(P_i, \Gamma_i)$, i.e., $\sum_{p_i^j \in C} \gamma_i^j \geq \beta$ for all $i \leq m$. The *$\beta$-hull* of $\mathcal{P}$, denoted by $\mathrm{CH}_\beta(\mathcal{P})$, is the intersection of all convex $\beta$-dense sets with respect to $\mathcal{P}$. See Figure 8(a) for an example. Note that for $m = 1$, $\mathrm{CH}_\beta(\mathcal{P})$ is the set of points whose Tukey depth is at least $1 - \beta$. We first prove an $O(n)$ upper bound on the complexity of $\mathrm{CH}_\beta(\mathcal{P})$ and then describe an algorithm for computing it.

**Theorem 6.1.** *Let $\mathcal{P} = \{(P_1, \Gamma_1), \ldots, (P_m, \Gamma_m)\}$ be a set of $m$ uncertain points in $\mathbb{R}^2$ under the multipoint model with $P = \bigcup_{i=1}^m P_i$ and $|P| = n$. For any $\beta \in [0, 1]$, $\mathrm{CH}_\beta(\mathcal{P})$ has $O(n)$ vertices.*

*Proof:* We call a convex $\beta$-dense set $C$ *minimal* if there is no convex $\beta$-dense set $C'$ such that $C' \subset C$. A convex $\beta$-dense set $C$ is minimal if $C = \mathrm{CH}(P \cap C)$. Therefore $C$ is a convex polygon whose vertices are a subset of $P$. Obviously $\mathrm{CH}_\beta(\mathcal{P})$ is the intersection of minimal convex $\beta$-dense sets. Therefore each edge of $\mathrm{CH}_\beta(\mathcal{P})$ lies on a line passing through a pair of points of $P$. Let $L$ be the set of lines supporting the edges of $\mathrm{CH}_\beta(\mathcal{P})$. We prove that $|L| \leq 2n$.

Fix a point $p \in P$. We claim that $L$ contains at most two lines that pass through $p$. Indeed if $p \in \mathrm{int}(\mathrm{CH}_\beta(\mathcal{P}))$, then no line of $L$ passes through $p$; if $p \in \partial(\mathrm{CH}_\beta(\mathcal{P}))$, then at most two lines of $L$ pass through $p$; and if $p \notin \mathrm{CH}_\beta(\mathcal{P})$, then the only lines of $L$ that can pass through $p$ are the two tangents of $\mathrm{CH}_\beta(\mathcal{P})$ from $p$. Hence at most two lines of $L$ pass through $p$, as claimed. ∎

**Algorithm.** We describe the algorithm for computing the upper boundary $\mathcal{U}$ of $\mathrm{CH}_\beta(\mathcal{P})$. The lower boundary of $\mathrm{CH}_\beta(\mathcal{P})$ can be computed analogously. We call a line $\ell$ passing through a point $p \in P_i$ *$\beta$-tangent* of $P_i$ at $p$ if the *above* open half-plane $h_\ell^{\mathrm{above}-}$ bounded by $\ell$ contains less than $\beta$-fraction of points of $P_i$ but the *below* closed half-plane $h_\ell^{\mathrm{below}}$ contains at least $\beta$-fraction of points.

It will be easier to work in the dual plane. The dual of a point $p = (a, b)$ is the line $p^* : y = ax - b$, and the dual of a line $\ell : y = mx + c$ is the point $\ell^* = (m, -c)$. The point $p$ lies above/below/on the line $\ell$ if and only if the dual point $\ell^*$ lies above/below/on the dual line $p^*$. Set $P_i^* = \left\{ p_i^{j*} \mid p_i^j \in P_i \right\}$ and $P^* = \bigcup_{i=1}^m P_i^*$. For a point $q \in \mathbb{R}^2$ and for $i \le m$, let $\kappa(q, i) = \sum \gamma_i^j$, where the summation is taken over all points $p_i^j \in P_i$ such that $q$ lies below the dual line $p_i^{j*}$. We define the $\beta$-level $\Lambda_i$ of $P_i^*$ to be the upper boundary of the region $\left\{ q \in \mathbb{R}^2 \mid \kappa(q, i) \ge \beta \right\}$. $\Lambda_i$ is an $x$-monotone polygonal chain composed of the edges of the arrangement $\mathcal{A}(P_i^*)$; the dual line of a point on $\Lambda_i$ is a $\beta$-tangent line of $P_i$. Let $\Lambda$ be the lower envelope of $\Lambda_1, \ldots, \Lambda_m$. See Figure 8(b).

Let $\ell$ be the line supporting an edge of $\mathcal{U}$. It can be proved that the dual point $\ell^*$ is a vertex of $\Lambda$: first, $\ell$ is a supporting line for some minimal convex $\beta$-dense set, hence $\ell$ is a $\beta$-tangent line for some $P_i$ and contains at least $\beta$-fraction of each $(P_i, \Gamma_i)$, that is, $\ell^* \in \Lambda$; second, as in the proof of Theorem 6.1, $\ell$ passes through a pair of points of $P$, hence $\ell^*$ is an intersection vertex of two lines of $P^*$. Next, let $q$ be a vertex of $\mathcal{U}$, then $q$ cannot lie above any $\beta$-tangent line of any $P_i$ (since for any $\beta$-tangent line $\ell_0$, there exists a convex $\beta$-dense set with $\ell_0$ bounding its upper part, and $\mathcal{U}$ has to lie no above $\ell_0$), which implies that the dual line $q^*$ passes through a pair of vertices of $\Lambda$ and does not lie below any vertex of $\Lambda$. Hence, each vertex of $\mathcal{U}$ corresponds to an edge of the upper boundary of the convex hull of $\Lambda$. See Figure 8(c). This observation suggests that $\mathcal{U}^*$, the dual of $\mathcal{U}$, can be computed by adapting an algorithm for computing the convex hull of a level in an arrangement of lines [6, 24]. We begin by describing a simple procedure, which will be used as a subroutine in the overall algorithm.

**Lemma 6.2.** *Given a line $\ell$, the intersection points of $\ell$ and $\Lambda$ can be computed in $O(n \log n)$ time.*

*Proof:* We sort the intersections of the lines of $P^*$ with $\ell$ in $O(n \log n)$ time. Let $\langle q_1, \ldots, q_u \rangle$, $u \le n$, be the sequence of these intersection points. For every $i \le m$, $\kappa(q_1, i)$ can be computed in a total of $O(n)$ time. Given $\left\{ \kappa(q_{j-1}, i) \mid 1 \le i \le m \right\}$, $\left\{ \kappa(q_j, i) \mid 1 \le i \le m \right\}$ can be computed in $O(1)$ time because if $q_j$ lies on a line of $P_i^*$, then only $\kappa(q_j, i)$ is different from $\kappa(q_{j-1}, i)$. A point $q_j \in \Lambda$ if $q_j \in \Lambda_i$ for some $i$ and lies below $\Lambda_i'$ for all other $i'$. This completes the proof of the lemma. ∎

The following two procedures can be developed by plugging Lemma 6.2 into the parametric-search technique [6, 24].

(A) Given a point $q$, determine whether $q$ lies below $\mathcal{U}^*$ or return the tangent lines of $\mathcal{U}^*$ from $q$.

(B) Given a line $\ell$, compute the edges of $\mathcal{U}^*$ that intersect $\ell$.

Using Lemma 6.2 and the parametric search technique described in [6, Section 2], (A) can be performed in $O(n \log^2 n)$ time. Using (A) as a procedure and the parametric search technique described in [6, Section 2], (B) can be done in $O(n \log^4 n)$ time.[5] Given (B), we can now compute $\mathcal{U}^*$ as follows.

We fix a parameter $r > 1$.[6] We compute a $(1/r)$-cutting $\Xi = \{\Delta_1, \ldots, \Delta_u\}$ of $P^*$, where $u = O(r^2)$. For each $\Delta \in \Xi$, we do the following. Using (B) we compute the edges of $\mathcal{U}^*$ that intersect $\partial\Delta$. We can then deduce whether $\Delta$ contains more than one vertex of $\mathcal{U}^*$. If $\partial\Delta$ is intersected by two disjoint edges $e_1$ and $e_2$ of $\mathcal{U}^*$ each of which has one vertex inside $\Delta$, then the answer is yes,

[5] In [6, Section 2], the running time for (B) was stated as $O(n \operatorname{polylog}(n))$ since the authors did not aim for the most efficient implementation.

[6] A $(1/r)$-*cutting* of $P^*$ is a triangulation $\Xi$ of $\mathbb{R}^2$ such that each triangle of $\Xi$ crosses at most $n/r$ lines of $P^*$.

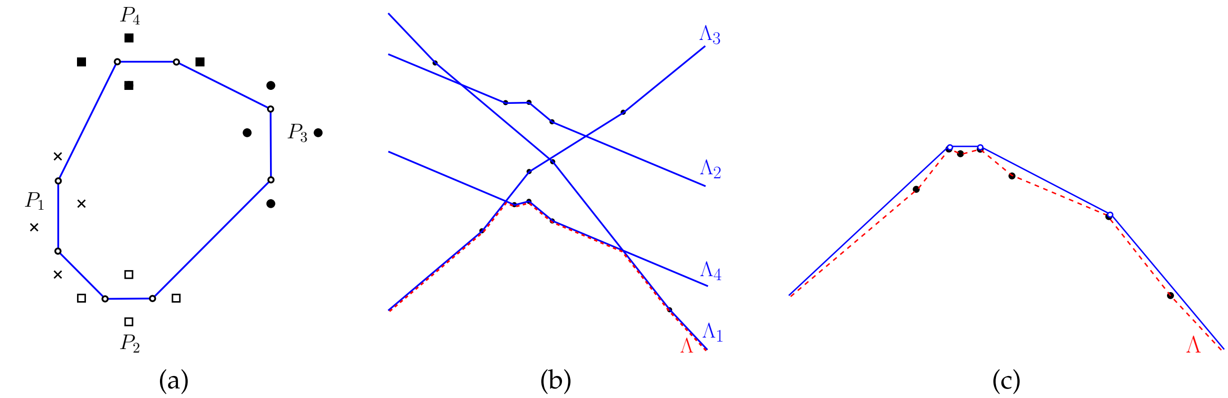


Figure 8: (a) An example of $\beta$-hull. $\mathcal{P}$ has 4 uncertain points (marked with different shapes); each uncertain point has 4 possible locations with probability 0.25 each. Here $\beta = 0.75$. (b) The $\beta$-levels $\Lambda_i$ (blue solid) of $P_i^*$, and their lower envelope $\Lambda$ (red dashed). (c) The upper boundary (blue solid) of the convex hull of $\Lambda$ (red dashed).

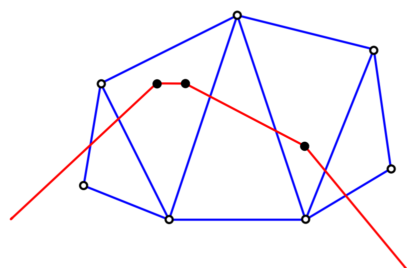

Figure 9: A $(1/r)$-cutting $\Xi$ with respect to the upper boundary of the convex hull of $\Lambda$.

and we solve the problem recursively in $\Delta$ with $\boldsymbol{\beta}_\Delta$ and a subset $L_\Delta$ of lines of $P^*$. See Figure 9. $\boldsymbol{\beta}_\Delta$ and $L_\Delta$ are defined as follows.

$\boldsymbol{\beta}_\Delta = \langle \beta_1^\Delta, \ldots, \beta_m^\Delta \rangle$ is the vector in which $\beta_i^\Delta$ indicates that $\Lambda_i$ is the $\beta_i^\Delta$-level of $P_i^* \cap L_\Delta$. For each $\Delta \in \Xi$, let $\kappa(\Delta) = \sum \gamma_i^j$ where the summation is taken over all points $p_i^j \in P_i$ such that $\Delta$ lies below the dual line $p_i^{j*}$. For $i \le m$, we set $\beta_i^\Delta = \beta_i^{\Delta'} - \kappa(\Delta)$, where $\beta_i^{\Delta'}$ is the value of $\beta_i^\Delta$ from the previous call. In the initial call, we use $\boldsymbol{\beta} = \langle \beta, \ldots, \beta \rangle$.

Let $P_\Delta^* \subset P^*$ be the set of lines that cross $\Delta$, and let $v_1$ (resp. $v_2$) be the vertex of $e_1$ (resp. $e_2$) that lies inside $\Delta$, respectively. $L_\Delta$ is the set of lines $\ell$ in $P_\Delta^*$ such that:[7]

(i) $\ell$ crosses the segment $v_1 v_2$, or

(ii) $\ell$ lies above the segment $v_1 v_2$ and the slope of $\ell$ is between the slopes of $e_1$ and $e_2$.

Let $n_\Delta = |L_\Delta|$. Since $\mathcal{U}^*$ is convex, each line $\ell$ crosses $\mathcal{U}^*$ at most twice for case (i), and each line $\ell$ satisfies case (ii) only once. Furthermore, if case (ii) holds for a triangle $\Delta$, then case (i) does not hold for any triangle. Hence, $\sum_{\Delta \in \Xi} n_\Delta \le 2n$.

[7] It can happen that $\partial\Delta$ is intersected by two disjoint edges $e_1$ and $e_2$ of $\mathcal{U}^*$ each of which has one vertex inside $\Delta$, and another two disjoint edges $e_3$ and $e_4$ of $\mathcal{U}^*$ each of which has one vertex inside $\Delta$. One can define $L_\Delta$ in this case similarly, and it does not affect our analysis.

Let $T(n_\Delta, \mu_\Delta)$ denote an upper bound on the running time of the recursive call within $\Delta$, where $n_\Delta = |L_\Delta|$, and $\mu_\Delta$ is the number of vertices of $\mathcal{U}^*$ lying inside $\Delta$. The overall running time will be $T(n, \mu)$, where $\mu = O(n)$ is the complexity of $\mathcal{U}^*$. In the initial call, we assume $\mu > 1$, otherwise it is trivial. The following recurrence can be derived.

$$T(n,\mu) = \begin{cases} \sum_{\Delta\in\Xi} T(n_\Delta, \mu_\Delta) + O(n\log^4 n) & \text{if } \mu > 1, \\ O(n\log^4 n) & \text{if } \mu \le 1. \end{cases}$$

where $n_\Delta \le n/r, \sum_{\Delta\in\Xi} n_\Delta \le 2n$ and $\sum_{\Delta\in\Xi} \mu_\Delta \le \mu = O(n)$. The above recurrence solves to $T(n,\mu) = O(n\log^5 n)$. We conclude the following.

**Theorem 6.3.** *Given a set $\mathcal{P}$ of uncertain points in $\mathbb{R}^2$ under the multipoint model with a total of $n$ sites, and a parameter $\beta \in [0,1]$, the $\beta$-hull of $\mathcal{P}$ can be computed in $O(n\log^5 n)$ time.*

**Remarks.** (1) The procedure $(B)$ can be performed in $O(n\log^3 n)$ expected time by using randomized search (see e.g. [5]) instead of parametric search. Consequently, the $\beta$-hull of $\mathcal{P}$ can be computed in $O(n\log^4 n)$ expected time.

(2) Let $k = \max_{1\le i\le m} |P_i|$. Note that the $\beta$-level $\Lambda_i$ has $O(k^2)$ complexity, and the lower envelope $\Lambda$ of $\Lambda_1, \dots, \Lambda_m$ has $O(mk^2) = O(nk)$ complexity. Thus, the upper hull of $\Lambda$, hence $\mathcal{U}^*$ and the $\beta$-hull of $\mathcal{P}$, can be computed in $O(nk\log n)$ time. This approach is more straightforward than using parametric search, and improves the running time for computing $\beta$-hull of $\mathcal{P}$ in Theorem 6.3 when $k = O(\log^4 n)$.

# 7. Conclusion

In this paper we studied the convex-hull problem in a probabilistic setting. We presented efficient algorithms for computing the probability of a point lying inside the convex hull of a set of uncertain points, and we also presented data structures for answering membership-probability queries. There are a few natural open problems:

(i) Extend our membership-probability algorithm in high dimensions to handle degeneracies in the input.

(ii) The size of the probability map is quite high. Is there a small-size approximate probability map? More precisely, given a parameter $\varepsilon > 0$, can we compute a small-size subdivision of $\mathbb{R}^d$ and associate a number $\hat{\mu}_f$ with each cell of the subdivision so that for all points $q \in f$, $|\mu(q) - \hat{\mu}_f| \le \varepsilon$. What is the size of such a subdivision and how quickly can it be computed?

(iii) Can the data structure described in Section 5 be extended to higher dimensions?

# References


[1] P. Afshani, P. Agarwal, L. Arge, K. Larsen, and J. Phillips. (Approximate) uncertain skylines. In *Proc. 14th IEEE Int. Conf. Data Theory*, pages 186–196, 2011.

[2] P. K. Agarwal, B. Aronov, S. Har-Peled, J. M. Phillips, K. Yi, and W. Zhang. Nearest neighbor searching under uncertainty II. In *Proc. 32nd ACM Sympos. Principles Database Syst.*, pages 115–126, 2013.

[3] P. K. Agarwal, S. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *Proc. 28th ACM Sympos. Principles Database Syst.*, pages 137–146, 2009.

[4] 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.

[5] P. K. Agarwal and M. Sharir. Efficient algorithms for geometric optimization. *ACM Comput. Surv.*, 30(4):412–458, 1998.

[6] P. K. Agarwal, M. Sharir, and E. Welzl. Algorithms for center and Tverberg points. *ACM Trans. Algs.*, 5(1):5:1–5:20, 2008.

[7] M. Ben-Or. Lower bounds for algebraic computation trees. In *Proc. 15th Annu. ACM Sympos. Theory Comput.*, pages 80–86, 1983.

[8] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom.*, 16:361–368, 1996.

[9] B. Chazelle. Cutting hyperplanes for divide-and-conquer. *Discrete Comput. Geom.*, 9(1):145–158, 1993.

[10] B. Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

[11] B. Chazelle, L. J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25(1):76–90, 1985.

[12] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Comput. Geom.*, 5(1):27–32, 1995.

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

[14] J. Eckhoff. Helly, radon, and Carathéodory type theorems. In P. M. Gruber and J. M. Wills, editors, *Handbook of Convex Geometry*, pages 389–448. North-Holland, 1993.

[15] H. Edelsbrunner, J. O'Rourke, and R. Seidel. Constructing arrangements of lines and hyperplanes with applications. In *Proc. 24th Annu. IEEE Sympos. Foundat. Comp. Sci.*, pages 83–91, 1983.

[16] E. Ezra and W. Mulzer. Convex hull of points lying on lines in $o(n \log n)$ time after preprocessing. *Comput. Geom.*, 46(4):417 – 434, 2013.

[17] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 1(4):132–133, 1972.

[18] D. Haussler and E. Welzl. epsilon-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[19] A. Jørgensen, M. Löffler, and J. Phillips. Geometric computations on indecisive points. In *Proc. 12th Workshop Algs. Data Struct.*, pages 536–547, 2011.

[20] P. Kamousi, T. Chan, and S. Suri. Stochastic minimum spanning trees in euclidean spaces. In *Proc. 27th Annu. ACM Sympos. Comput. Geom.*, pages 65–74, 2011.

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

[22] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15(1):287–299, 1986.

[23] M. Löffler. *Data Imprecision in Computational Geometry*. PhD thesis, Dept. Computer Sci., 2009.

[24] J. Matoušek. Computing the center of planar point sets. In J. E. Goodman, R. Pollack, and W. Steiger, editors, *Computational Geometry: Papers from the DIMACS Special Year*, pages 221–230. Amer. Math. Soc., 1991.

[25] J. Matoušek and O. Schwarzkopf. Linear optimization queries. In *Proc. 8th Annu. ACM Sympos. Comput. Geom.*, pages 16–25, 1992.

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

[27] P. Pérez-Lantero. Area and perimeter of the convex hull of stochastic points. *CoRR*, abs/1412.5153, 2014.

[28] J. Phillips. *Small and Stable Descriptors of Distributions for Geometric Statistical Problems*. PhD thesis, Dept. Computer Sci., 2009.

[29] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20(2):87–93, 1977.

[30] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proc. 18th Annu. ACM Sympos. Theory Comput.*, pages 404–413, 1986.

[31] R. Seidel. Convex hull computations. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 495–512. CRC Press, 2004.

[32] S. Suri, K. Verbeek, and H. Yıldız. On the most likely convex hull of uncertain points. In *Proc. 21st Annu. European Sympos. Algs.*, pages 791–802, 2013.

[33] H. Wang and W. Zhang. The $\tau$-skyline for uncertain data. In *Proc. 26th Canad. Conf. Comput. Geom.*, pages 326–331, 2014.

[34] Z. Zhao, D. Yan, and W. Ng. A probabilistic convex hull query tool. In *Proc. 15th Int. Conf. Ext. Database Tech.*, pages 570–573, 2012.