On the Most Likely Convex Hull of Uncertain Points

Subhash Suri, Kevin Verbeek, and Hakan Yıldız

University of California, Santa Barbara, USA

Abstract. Consider a set of d-dimensional points where the existence or the location of each point is determined by a probability distribution. The convex hull of this set is a random variable distributed over exponentially many choices. We are interested in finding the *most likely convex hull*, namely, the one with the maximum probability of occurrence. We investigate this problem under two natural models of uncertainty: the *point* (also called the *tuple*) model where each point (site) has a fixed position s_i but only exists with some probability π_i, for $0 < \pi_i \leq 1$, and the *multipoint* model where each point has multiple possible locations or it may not appear at all. We show that the most likely hull under the point model can be computed in $O(n^3)$ time for n points in $d = 2$ dimensions, but it is NP–hard for $d \geq 3$ dimensions. On the other hand, we show that the problem is NP–hard under the multipoint model even for $d = 2$ dimensions. We also present hardness results for approximating the probability of the most likely hull. While we focus on the most likely hull for concreteness, our results hold for other natural definitions of a probabilistic hull.

1 Introduction

We study the problem of computing the *most likely convex hull* of n uncertain points. The problem is fundamental in its own right, extending the notion of minimal convex enclosure to probabilistic input, but is also motivated by a number of applications dealing with noisy data. Before formalizing the problem, let us mention some motivating scenarios for our problem. In *movement ecology* [12, 13], scientists track the movements of a group of animals using sensors with the goal of inferring their natural "home range". The ecologists have long known that the smallest convex polygon containing all possible locations visited by the animals is a gross overestimation of the home range, due to the outlier problem, and instead have begun to consider probability-based isopleths. The most likely hull is one possible tool in this analysis: use a discrete set of landmarks (points), assign probability to each based on the frequency of the animals' visits to the landmarks, and compute the most likely convex hull of this probabilistic set of points as the most probable home range. As another example, consider monitoring of a large geographic area for physical activity (e.g., earthquake tremors). After collecting data over a period of time, we want to estimate the most likely region of activity. Since the value of a prediction decreases sharply with the

rate of false positives, we want to find the tightest region for expected activity, and the most likely hull is a natural candidate. Finally, as a growing number of applications rely on machine learning and data mining for classification, we are inevitably forced to work with data whose attributes are inherently probabilistic. Computing meaningful geometric structures over these data is an interesting, and challenging, algorithmic problem. The most likely hull is a convenient vehicle to investigate these types of problems, although our methods and results are applicable more broadly, as discussed later.

In the *point* model of uncertain data,[1] the input is a pair (S, Π), where $S = \{s_1, s_2, \ldots, s_n\}$ is a set of n points (sites) in the d-dimensional space, and $\Pi = \{\pi_1, \pi_2, \ldots, \pi_n\}$ is a probability vector with the interpretation that site s_i is active (namely, present) with probability π_i. The probabilities π_i are mutually independent. Thus, a random instance of (S, Π) includes each point s_i with an independent probability π_i. The convex hull of (S, Π) is a random variable, which assumes values over the convex hulls of the (at most) 2^n possible subsets. We are interested in computing the *most likely convex hull* for (S, Π).

The *multipoint* model generalizes the point model to incorporate *locational uncertainty*. The ith point of the input is described as $\left(\{s_i^1, \pi_i^1\}, \ldots, \{s_i^{k_i}, \pi_i^{k_i}\}\right)$, with the interpretation that the point appears at the position s_i^j with probability π_i^j, for $j = 1, 2, \ldots, k_i$. Different points can have a different number of possible locations k_i, but for simplicity we assume that the total number of locations is linear. Finally, we allow $\sum_{j=1}^{k_i} \pi_i^j < 1$ to include the possibility that the ith point does not exist at all, thus achieving a strict generalization of the point model.

Our first result shows that the most likely hull of 2-dimensional points in the point model can be found in $O(n^3)$ time. We then show that the problem becomes NP-hard for dimensions $d \geq 3$. We also show that approximating the probability of the most likely hull is provably hard. In particular, computing a hull whose likelihood is within factor $2^{-O(n^{1-\epsilon})}$ of the optimal is NP–hard. This is nearly tight because a factor-(2^{-n}) approximate hull is easily computed by a simple greedy algorithm. Under the multipoint model, we show that the most likely hull problem is NP-hard even in two dimensions, and also inapproximable to a factor better than $2^{-O(n^{1-\epsilon})}$ unless P=NP. While we focus on the most likely hull as a natural and concrete example, our algorithms and techniques apply more broadly to other possible ways of defining a probabilistic convex hull.

Related Work. Uncertainty in geometric computing has been studied in a few different ways. In [16, 17], Löffler and van Kreveld have considered problems on "imprecise" objects: each object, such as a point, can be anywhere inside a simple geometric region. For instance, given a set of imprecise points, one can ask for the maximum possible area of the convex hull of these points. However, this line of

[1] The point model is also called the *tuple* model in database research,and has been used for studying clustering, ranking etc. of uncertain multi-attribute objects, modeled as points in d-space.

research looks at the worst-case behavior, and not the stochastic behavior, which is the main focus of our work. In a more closely related and interesting work [14], Jørgensen, Löffler and Phillips, develop a general framework for geometric shape-fitting problems and describe how the solutions to these problems vary with respect to the uncertainty in the points. Another line of research has focused on uncertainty caused by the finite machine precision [15, 18, 19]. The goal there is to achieve robustness under bounded precision, and not to compute structures that are most representative under a probability distribution. There also has been extensive research in the database community on clustering and ranking of uncertain data [4, 5, 10] and on range searching and indexing [1–3].

2 Two-Dimensional Most Likely Hull in the Point Model

In this section, we describe a dynamic programming algorithm for computing the most likely hull of n points in the plane under the point model of uncertainty. For simplicity, we assume that no three points are collinear, but the algorithm is easily modified to handle such degeneracies. We begin with some general technical facts related to convex hulls of uncertain points in the point model.

Let (S, Π) denote the input to the uncertain convex hull problem in d-space. A subset $A \subseteq S$ occurs as an outcome of a probabilistic experiment with probability $\pi(A)$ given by

$$\pi(A) \;=\; \prod_{s_i \in A} \pi_i \;\times\; \prod_{s_i \notin A} \overline{\pi_i}$$

where we use the notation $\overline{\pi_i} = (1 - \pi_i)$. Given an outcome A, its convex hull is denoted as $\mathcal{CH}(A)$. For a convex polytope C, we define its *likelihood*, denoted $\mathcal{L}(C)$, as the probability that C is the convex hull of the random outcome of a probabilistic experiment on (S, Π). In other words,

$$\mathcal{L}(C) \;=\; \Pr\big[\,\mathcal{CH}(A) \equiv C\,\big] = \sum_{\substack{A \subseteq S \\ \mathcal{CH}(A) \equiv C}} \pi(A)$$

The *most likely hull* of (S, Π) is the polytope C with the maximum value of $\mathcal{L}(C)$. Our first lemma shows that $\mathcal{L}(C)$ can be written as a product of two factors where the first factor involves only the *vertices* of C, and not all the sites that fall inside C. Please see Appendix A for a proof.

Lemma 1. *Let C be a convex polytope, $V \subseteq S$ be its vertex set, and $S_{out} \subseteq S$ the set of sites lying outside C. Then, we have the following:*

$$\mathcal{L}(C) \;=\; \prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i},$$

Likelihood Contributions of Edges. We now describe how to find the most likely hull for a 2-dimensional input under the point model. Our algorithm computes, for each site s_i, the most likely hull with s_i as its lowest (minimum y-coordinate) vertex, and then outputs the best hull over all choices of s_i. For ease

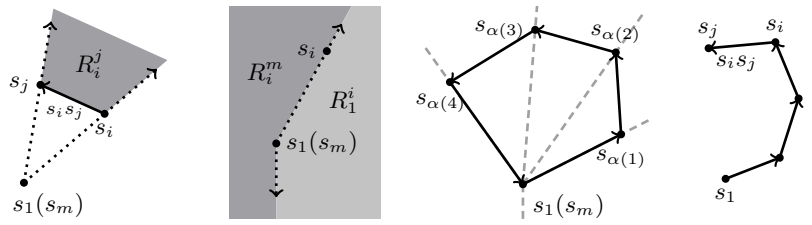

Fig. 1

of reference, let us call a convex polygon with s_i as its lowest vertex, a *hull rooted at* s_i. We decompose the likelihood of a convex hull into several components, each associated with an edge of the hull. The key to the computational efficiency is to ensure that the component associated with an edge *does not depend on the hull* in which the edge participates. Geometrically, we associate a *wedge shaped region* with each edge, depending only on the choice of the lowest vertex, and define the contribution based only on the sites contained in this wedge. We now discuss this in more details.

Suppose we want to compute the most likely hull rooted at s_1. Without loss of generality, let $s_2, \ldots, s_{m-1}$ be the sequence of sites (all lying above s_1) in the counter-clockwise order around s_1, for $(m-1) \leq n$. Any hull rooted at s_1 has a subsequence of $s_1, \ldots, s_{m-1}$ as its vertex set. Finally, for notational convenience, we add an artificial site $s_m = s_1$ (a copy of the root point) with probability zero.

Given two sites s_i and s_j, with $1 \leq i < j \leq m$, we use $s_i s_j$ to denote the *directed edge* drawn from s_i to s_j. To each directed edge $s_i s_j$, we associate a region of space R_i^j. For an edge not involving s_1 or its copy s_m, namely $s_i s_j$, for $1 < i < j < m$, R_i^j is the region bounded by the segment $s_i s_j$ and the rays $\overrightarrow{s_1 s_i}$ and $\overrightarrow{s_1 s_j}$. See Figure 1a for illustration. For edges with the first endpoint at s_1, namely $s_1 s_i$, for $1 < i < m$, R_1^i is the region bounded (on its left) by the downward ray extending from s_1 and the ray $\overrightarrow{s_1 s_i}$. The complementary region of R_1^i is also important, and we call it R_i^m, associated with the edge $s_i s_m$, which is the reverse edge of $s_1 s_i$. See Figure 1b.

We now define the *contribution* of the directed edge $s_i s_j$, denoted $\mathcal{C}(s_i s_j)$, as π_i times the probability that none of the sites in the region R_i^j (except s_i and s_j) are present, including the sites that may lie below s_1. That is,

$$\mathcal{C}(s_i s_j) = \pi_i \times \prod_{s_k \in R_i^j} \overline{\pi_k}$$

The following lemma shows how these edge contributions help us compute the likelihood of a convex hull C.

Lemma 2. *Let C be a hull rooted at s_1, with vertices $s_1, s_{\alpha(1)}, \ldots, s_{\alpha(\ell)}$ in the counter-clockwise order. Then,*

$$\mathcal{L}(C) = \mathcal{C}(s_1 s_{\alpha(1)}) \times \mathcal{C}(s_{\alpha(1)} s_{\alpha(2)}) \times \cdots \times \mathcal{C}(s_{\alpha(\ell-1)} s_{\alpha(\ell)}) \times \mathcal{C}(s_{\alpha(\ell)} s_m)$$

Proof. Partition the space outside C into the regions $R_1^{\alpha(1)}, R_{\alpha(1)}^{\alpha(2)}, \ldots, R_{\alpha(\ell-1)}^{\alpha(\ell)}, R_{\alpha(\ell)}^{m}$ by drawing a downward ray from s_1 and drawing rays $\overrightarrow{s_1 s_{\alpha(j)}}$ for each $1 \leq j \leq \ell$. (See Figure 1c for an example.) Then, by Lemma 1, it is easy to see that the $\mathcal{L}(C)$ is the product of the contributions of the edges of C. □

The contribution of each edge can be computed in constant time after an $O(n^2)$-time preprocessing, using a modified version of a triangle query data structure of [11]. We give the details of this structure in Appendix B.

The Dynamic Programming Algorithm. Our dynamic programming algorithm computes, for each edge $s_i s_j$, the convex chain whose edges yield the *maximum product of contributions* under the following constraints:

1. The sequence of vertices in the chain is a subsequence of $s_1, \ldots, s_m$.
2. The first vertex of the chain is s_1.
3. The last edge of the chain is $s_i s_j$. (See Figure 1d for an example.)

We denote this maximum chain by $\mathcal{T}(s_i s_j)$. With a slight abuse of notation, we also use $\mathcal{T}(s_i s_j)$ to denote the product of the edge contributions of this chain. Clearly, all chains of the form $\mathcal{T}(s_i s_m)$ correspond to polygons rooted at s_1, and the one with the maximum contribution is the most likely hull we want. Our dynamic programming formulation is fairly standard, and similar style of algorithms have been used in the past for computing largest convex subsets [9, 6] and monochromatic islands [7].

We now describe an optimal substructure property crucial for our dynamic programming algorithm. Consider a chain $\mathcal{T}(s_i s_j)$. This, by definition, has the maximum likelihood of all chains terminating with the edge $s_i s_j$. If we remove the last vertex s_j of $\mathcal{T}(s_i s_j)$, and the corresponding edge $s_i s_j$, then the remaining chain should be the optimal chain terminating at s_i *that can be extended to* s_j *without violating convexity*. In other words, the remaining chain is the maximum among all chains $\mathcal{T}(s_k s_i)$ (where $1 \leq k < i$) such that the path $s_k \to s_i \to s_j$ is a left turn. This implies the following recurrence:

$$\mathcal{T}(s_i s_j) = \begin{cases} \mathcal{C}(s_1 s_j) & \text{if } i = 1 \\ \mathcal{C}(s_i s_j) \times \max\limits_{\substack{1 \leq k < i \\ s_k \to s_i \to s_j \text{ is a left turn}}} \Big(\mathcal{T}(s_k s_i)\Big) & \text{otherwise} \end{cases}$$

We use this recurrence to compute all the chains $\mathcal{T}(s_i s_j)$ as follows. We begin by setting $\mathcal{T}(s_1 s_i)$ to $\mathcal{C}(s_1 s_i)$ for all $1 < i \leq m$. Then, we process all sites s_i in increasing order of i. When we process a site s_i, we compute all chains $\mathcal{T}(s_i s_j)$ by using the previously computed chains. This can be done in $O(n)$ time as follows. Let S_{prec} be the set of sites $\{s_1, \ldots, s_{i-1}\}$ and S_{succ} be the set $\{s_{i+1}, \ldots, s_m\}$. Let $s_{\beta(1)}, \ldots, s_{\beta(\ell)}$ be the sites in S_{prec} in counter-clockwise order around s_i.[2] For each site $s_{\beta(u)}$ in S_{prec}, we define s_u^* to be the site s_k among the sequence

[2] This counter-clockwise order for all sites s_i can be precomputed in $O(n^2 \log n)$ time.

$s_{\beta(1)}, \ldots, s_{\beta(u)}$ that maximizes $\mathcal{T}(s_k s_i)$. The site s_u^* can be computed for all sites $s_{\beta(u)}$ with a linear sweep of the sites in S_{prec} in order.

For each site $s_{\beta(u)}$ in S_{prec}, we set the value $\mathcal{T}(s_i s_j)$ to $\mathcal{C}(s_i s_j) \times \mathcal{T}(s_u^* s_j)$ for all sites s_j in S_{succ} inside the wedge bounded by the lines $\overleftrightarrow{s_{\beta(u)} s_i}$ and $\overleftrightarrow{s_{\beta(u+1)} s_i}$. (See Figure 2a.) Notice that the sites in this wedge are the sites that form a left turn when connected to $s_{\beta(1)}, \ldots, s_{\beta(u)}$ through s_i (the condition in the recurrence relation). Note that, by considering the sites $s_{\beta(u)}$ in radial order around s_i, we can locate each site in the wedge of interest in constant time.

The processing of a single point s_i takes $O(n)$ time, and thus we can find the most likely hull rooted at s_1 in $O(n^2)$ time, and the global most likely hull of P in $O(n^3)$ time. The algorithm needs $O(n^2)$ space, dominated by the storage of the $\mathcal{T}(\cdot)$ values.

Theorem 1. *The most likely convex hull of an uncertain point set defined by n sites in the point model can be computed in $O(n^3)$ time and in $O(n^2)$ space.*

3 Hardness of the 3-Dimensional Most Likely Hull

We now show that computing the most likely hull in 3 or more dimensions is NP-hard in the point model. In particular, we give a reduction from the vertex cover problem in penny graphs to the 3-dimensional most likely hull problem.

A *penny graph* is a graph $G = (V, E)$ along with an embedding $\rho : V \to \mathbb{R}^2$ such that $\|\rho(u) - \rho(v)\|_2 = 2$ if $(u, v) \in E$, and $\|\rho(u) - \rho(v)\|_2 > 2$ if $(u, v) \notin E$, where $\|.\|_2$ denotes L_2 norm. In other words, a penny graph admits a planar drawing where vertices are represented as unit disks with pairwise disjoint interiors, and two disks make contact if and only if there is an edge between the two corresponding vertices. We denote the centers of the unit disks by the points $p_1, \ldots, p_n$, and the point of contact between two adjacent disks with centers p_i and p_j by p_{ij}. The following simple observation about the penny graph embedding (whose proof is given in Appendix C) will be critical in our reduction. See Figure 2b for an illustration.

Lemma 3. *$\|p_k - p_{ij}\|_2 \geq \sqrt{3}$, for all $k \neq i, j$.*

The vertex cover problem for penny graphs is to find the smallest subset $U \subseteq V$ of vertices such that every edge of the graph has an endpoint in U. This problem was shown to be NP-hard in [8]. Our reduction relies on the following simple but important property of the most likely hull in the point model, whose proof is included in Appendix D.

Lemma 4. *Any point (s_i, π_i) with $\pi_i \geq 1/2$ is in the most likely hull.*

The Reduction. Consider an instance of the vertex cover problem for a penny graph G, with $p_1, \ldots, p_n$ being the disk centers of the embedding of G. We create an instance of the most likely hull problem in three dimensions, as follows. All the sites lie on one of the two paraboloids, $\mathcal{P}_1 : z = x^2 + y^2$ or $\mathcal{P}_2 : z = x^2 + y^2 - 2$. In

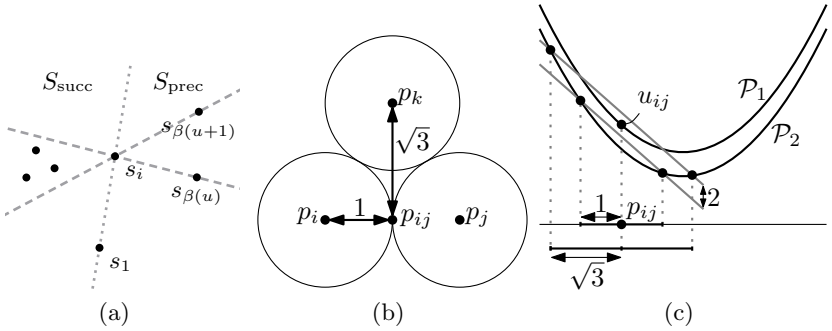

Fig. 2

particular, for each disk center p_i, we create a site u_i by vertically lifting p_i onto the paraboloid $\mathcal{P}_2$. All these points are assigned a fixed probability $\pi_i = \alpha < \frac{1}{2}$.

The sites on $\mathcal{P}_1$ are associated with the contact points p_{ij} but are not a direct lifting of the contact points themselves. Instead, for each contact point $p_{ij} = (x_{ij}, y_{ij})$, we define four new points $p_{ij}^N = (x_{ij}, y_{ij} + \delta)$, $p_{ij}^E = (x_{ij} + \delta, y_{ij})$, $p_{ij}^S = (x_{ij}, y_{ij} - \delta)$, and $p_{ij}^W = (x_{ij} - \delta, y_{ij})$, for some $\delta > 0$. (We set the value of δ later.) Next, we add a set X_{ij} of m arbitrary points inside the quadrilateral formed by p_{ij}^e ($e \in \{N, E, S, W\}$). We lift each of the p_{ij}^e onto $\mathcal{P}_1$ to obtain a site u_{ij}^e, for $e \in \{N, E, S, W\}$, and each of these points is assigned a probability of 1. Finally, the subsets X_{ij} are lifted onto $\mathcal{P}_1$ to get subsets Y_{ij}, and each of these points are assigned a fixed probability $\beta > \frac{1}{2}$. All these points, lying on the paraboloids $\mathcal{P}_1$ and $\mathcal{P}_2$, along with their associated probabilities form the input for our most likely hull problem.

The main idea of the reduction is that we want to "cover" each set Y_{ij} by putting either u_i or u_j on the most likely hull. In the penny graph, this corresponds to covering the edge associated with the contact point p_{ij} by the vertex associated with p_i or p_j. We now describe this relation in more depth, starting with a well-known lemma about the lifting transform. A proof of this lemma is included in Appendix E for reference.

Lemma 5. *Consider a point $p \in \mathbb{R}^2$, and let $u(p)$ be its vertical projection (lifting) onto the paraboloid $\mathcal{P}_1$, and $H(p)$ the hyperplane tangent to $\mathcal{P}_1$ at $u(p)$. Then, the vertical projection $u(p')$ of all points $p' \in \mathbb{R}^2$ at distance r from p lies on a hyperplane parallel to $H(p)$ whose vertical distance from $H(P)$ is r^2.*

The points u_i's (liftings of p_i's) lie on $\mathcal{P}_2$, which is a vertical downward shift of $\mathcal{P}_1$. Now, if u_{ij} is the point obtained by lifting p_{ij} to $\mathcal{P}_1$, then by Lemma 5 the points u_i and u_j are vertically 1 unit below the tangent plane of $\mathcal{P}_1$ at u_{ij}, while the points u_k ($k \neq i, j$) are at least vertically 1 unit above this plane by Lemma 3 (see Figure 2c). If we treat $\mathcal{P}_1$ as an "obstacle", then u_i and u_j can "see" u_{ij} from below, while the points u_k ($k \neq i, j$) cannot. Thus there exists a small enough $\delta > 0$ such that Y_{ij} is contained in the convex hull of u_{ij}^e ($e \in \{N, E, S, W\}$)

with either u_i or u_j, but not with u_k ($k \neq i, j$). The following lemma, whose proof is in Appendix F, describes a sufficient upper-bound on δ.

Lemma 6. *If $\delta < \sqrt{3} - \sqrt{2}$, then the points u_i and u_j can see the entire quadrilateral on $\mathcal{P}_1$ formed by u_{ij}^e ($e \in \{N, E, S, W\}$) from below, but no u_k ($k \neq i, j$) can see any part of the quadrilateral from below.*

Theorem 2. *Computing the most likely hull in three dimensions is NP-hard.*

Proof. We show that computing the likelihood of the most likely hull is NP-hard. Given an instance of the vertex cover problem for penny graphs, we construct an instance of the most likely hull problem in three dimensions as described above (e.g., with $\delta = 0.25$). We choose m, α, and β such that $\beta^m < \alpha$, and $\alpha < 0.5 < \beta$; e.g., $m = 3$, $\alpha = 0.25$, and $\beta = 0.6$. By Lemma 4 all points on $\mathcal{P}_1$ must be on or inside the most likely hull, and so we only need to choose which points u_i ($1 \leq i \leq n$) are on the most likely hull. No point from a set Y_{ij} can be on the most likely hull because then we could add either u_i or u_j to the hull and increase the likelihood of the hull, since $\beta^m < \alpha$. Thus, the likelihood of the most likely hull is determined by the number κ of points u_i ($1 \leq i \leq n$) that are on the most likely hull, and its likelihood is $\alpha^\kappa(1-\alpha)^{n-\kappa}$. Every point u_i on the most likely hull corresponds to a vertex of the penny graph, and by construction and Lemma 6, these vertices form a vertex cover of the penny graph. Thus the penny graph has a vertex cover of size κ if and only if the likelihood of the most likely hull is at least $\alpha^\kappa(1-\alpha)^{n-\kappa}$. Finally, it is easy to see that the construction can be performed in polynomial time. □

The proof above directly implies that there exists no polynomial-time $(\frac{\alpha}{1-\alpha})$-approximation algorithm to compute the likelihood of the most likely hull unless $P = NP$. Although we can change the value of α to obtain a stronger bound, we give a more general argument below.

Inapproximability. The likelihood of a hull is a product of terms. We show that, under mild conditions, NP-hard optimization problems of this form cannot be approximated well by a multiplicative factor, unless $P = NP$.

Let $\mathcal{O} = (\mathcal{I}, \mathcal{F}, f)$ be an optimization problem where $\mathcal{I}$ is the set of instances, $\mathcal{F}$ is a function over $\mathcal{I}$ such that $\mathcal{F}(I)$ describes the set of feasible solutions for instance I, and f is an optimization function over all feasible solutions. For an instance $I \in \mathcal{I}$, let $|I|$ denote the size of I. We say that $\mathcal{O}$ is *product composable* if, given any collection of problem instances $I_1, \ldots, I_k \in \mathcal{I}$, we can construct a new instance $I^* \in \mathcal{I}$ in polynomial time (w.r.t. $|I^*|$) satisfying the following:

1. $|I^*| = \sum_{i=1}^k |I_i|$.
2. There is a bijection between $\mathcal{F}(I^*)$ and $\mathcal{F}(I_1) \times \ldots \times \mathcal{F}(I_k)$ such that for each solution $S \in \mathcal{F}(I^*)$ with the matching tuple $(S_1, \ldots, S_k)$, $f(S) = \prod_{1 \leq i \leq k} f(S_i)$.
3. Given a solution $S \in \mathcal{F}(I^*)$, one can construct the solutions in its matching tuple in polynomial time.

In other words, we can form a new instance I^* by combining the instances $I_1, \ldots, I_k$ in an independent way. We now state the following lemma, whose proof is given in Appendix G.

Lemma 7. *If a maximization problem $\mathcal{O}$ is product composable and cannot be approximated within a constant $c < 1$ in polynomial time, then there exists no polynomial-time $2^{-O(n^{1-\epsilon})}$-approximation algorithm for $\mathcal{O}$, where n is the size of the instance and $\epsilon > 0$.*

Although the most likely hull problem is not product composable itself, this property only needs to hold for a subproblem. The subproblem formed by the instances used in our NP-hardness reduction is product composable, which easily follows from the construction. We defer a detailed explanation of this property to the full version of the paper.

Corollary 1. *For any $\epsilon > 0$, there exists no polynomial-time $2^{-O(n^{1-\epsilon})}$-approximation algorithm for the most likely hull problem in three dimensions, unless P=NP.*

Finally we observe that one can trivially achieve 2^{-n}-approximation of the most likely hull problem as follows: simply take the convex hull of all sites with probability at least $\frac{1}{2}$.

4 Most Likely Hull in the Multipoint Model

In this section, we show that computing the most likely hull in the multipoint model is NP–hard even for two dimensions. (The technical definition of the most likely hull under the multipoint model differs slightly from that of the point model, but the following abridged description should be accessible without a need for those details. A more complete formal description is included in Appendix H.) Our proof uses a reduction from 3-SAT.

Consider a 3-SAT instance (V, U) where V is the set of the variables and U is the set of clauses. We first construct $6|U|$ points on the unit circle. We call these points the *anchors* and use them as permanent points (i.e., points with probability 1) in our hull problem instance. Between each pair of consecutive anchors, we place a single point on the unit circle that we call a *spike*. (See Figure 3a.) We assign an independent existence probability of $\frac{1}{2}$ to each spike. As we will explain shortly, the main idea of our construction is that the most likely hull includes all spikes in its interior if and only if the 3-SAT instance is satisfiable.

For each variable v, we construct two additional sets of points, one corresponding to the case that v is true and one corresponding to the case that v is false. In particular, for each clause u that v appears in positive form, we construct a point p_v^u covering a single spike, at the intersection of the lines tangent to the unit circle at the two anchors next to the spike. We assign each p_v^u a probability of $\frac{1}{2}$ but this probability is dependent, as we will put p_u^v in the same tuple with another point in the rest of the construction. We construct all points

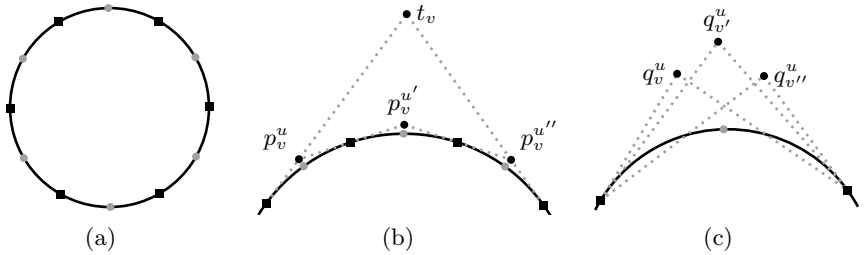

Fig. 3: (a) Anchors (black squares) and spikes (gray circles) on the unit circle. (b) Construction of t_v. (c) The three points constructed for clause u.

p_u^v for a single variable v over a consecutive sequence of spikes, and then put a single point t_v covering the constructed points. (See Figure 3b.)

We apply the same construction for all clauses that v appears in negated form. This creates an additional set of points p_u^v, all of which we cover with a single point f_v as we did for t_v. We put t_v and f_v to the same probabilistic tuple and assign each a probability of $\frac{1}{2}$. That is, in a probabilistic experiment, either t_v or f_v is present (with equal probability), but not both. Existence of t_v is meant to imply that v is assigned true, whereas the existence of f_v is meant to imply that v is assigned false.

Finally, for each clause u, we construct three additional points covering a single spike. These points are constructed in such a way that: (1) they do not cover any other spike, and (2) they are in convex position with respect to each other and the two anchors next to the covered spike. Each of these points corresponds to a distinct variable v that appears in the clause. We denote the point associated with variable v by q_v^u. (See Figure 3c.) We put each q_v^u to the same probability tuple as the previously constructed point p_v^u and assign it probability $\frac{1}{2}$. That is, in an experiment, either q_v^u or p_v^u exists (with equal probability), but not both. The following lemma, whose proof is in Appendix I implies that the most likely hull covers spikes if possible.

Lemma 8. *The most likely hull has likelihood* $(1/2)^{3|U|+|V|}$ *if and only if it contains all spikes in its interior. Otherwise, its likelihood is at most* $(1/2)^{3|U|+|V|+1}$.

We now describe how the satisfiability of the 3-SAT instance relates to our construction. Consider a variable v. Notice that, if the most likely hull covers all spikes below t_v, then either t_v or all points p_v^u below t_v appears in the hull as a vertex. If t_v appears in the hull, then the hull is can pass through the points q_v^u (which are in the same probabilistic tuples with points p_v^u), and cover spikes representing the clauses that v appears in positive form. This corresponds to the case that v is assigned true and all corresponding clauses are satisfied. Similar notion also applies to f_v and the clauses that v appears in negated form. If all spikes are covered, then all clauses are satisfied and so is the 3-SAT instance. Combining this idea with Lemma 8, we deduce the following lemma. A formal proof of this lemma is included in Appendix J.

Lemma 9. *The 3-SAT instance is satisfiable if and only if the most likely hull has likelihood* $(1/2)^{3|U|+|V|}$.

Theorem 3. *Computing the most likely hull in the multipoint model is NP-hard.*

Lemma 8 in fact implies a stronger result: It is NP-hard to compute the likelihood of the most likely hull within any factor $c > \frac{1}{2}$. By construction, the problem instances that we create are product composable. Then, by Lemma 7, we can state the following theorem.

Theorem 4. *For any* $\epsilon > 0$*, there exists no polynomial-time* $2^{-O(n^{1-\epsilon})}$*-approximation algorithm for the most likely hull problem in the multipoint model unless P=NP.*

5 Extensions and Concluding Remarks

Making sense of probabilistic (uncertain) data is a complex and challenging task. Even for simple numerical data, elementary statistics such as mean, median, or mode serve a useful first order approximation. For multi-dimensional spatial data, however, there are no universally agreed upon summaries of similar generality. Our work is an attempt to explore some natural geometric structures, and their complexity, over probabilistic data. For "convexity" of uncertain data, one possibility is to compute the distribution over the entire space: for each point of the space, compute the probability that it is inside the convex hull. In a different work, we are also exploring that direction but (i) a full distribution is inevitably quite expensive to compute (requiring a worst-case space complexity $\Omega(n^{d^2})$), and (ii) the distribution still does not lend itself to a simple and "intuitive" description of a convex hull.

Therefore, algorithms for computing or estimating succinct summary hulls are a useful tool in the analysis of uncertain geometric data. While we focused exclusively on the Most Likely Hull as a natural analog of the expected value for numerical data, our techniques are applicable to several other ways of defining the "best" hull. Any useful definition of the likely hull must include a penalty function for misclassifying points, *both false positives and false negatives*. If only false negatives (points outside the hull) are penalized, then the convex hull of *all* the points has the best score. Our dynamic programming algorithm for the point model in 2 dimensions can be extended for several natural scoring functions.

For instance, one simple scoring function measures the *agreement* on the "in" and "out" classification. A convex hull C splits the point set into two parts: inside and outside. We can measure the "quality" $Q(C)$ of a hull C by its expected agreement with a random hull's classification: the number of points of S whose classification (in or out) is the same for both C and the hull of a random outcome. Both our dynamic programming algorithm for computing the hull in 2 dimensions, and the hardness in 3 dimensions, under the point model carry over to this "Symmetric Difference Hull" definition. Similarly, another scoring function for measuring the fraction of points correctly classified counts

the number of points in the random outcome that lie in C plus the number of non-sample points that lie outside C. All our results hold for this model as well.

In summary, we believe that the study of geometric structures over probabilistic data is a fundamental problem, and our results are only a first, but promising, step. One can ask similar questions about many basic geometric structures, including Voronoi diagrams, Delaunay triangulations, shortest paths, range queries, and maxima.

References

1. P. Afshani, P. K. Agarwal, L. Arge, K. G. Larsen, and J. M. Phillips. (Approximate) uncertain skylines. *Theory Comput. Syst.*, 52(3):342–366, 2013.
2. P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi. Indexing uncertain data. In *PODS*, pages 137–146, 2009.
3. P. K. Agarwal, S.-W. Cheng, and K. Yi. Range searching on uncertain data. *ACM Transactions on Algorithms*, 8(4):43, 2012.
4. C. C. Aggarwal. *Managing and Mining Uncertain Data*, volume 35 of *Advances in Database Systems*. Kluwer, 2009.
5. C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. *IEEE Trans. Knowl. Data Eng.*, 21(5):609–623, 2009.
6. D. Avis and D. Rappaport. Computing the largest empty convex subset of a set of points. In *Proc. of the 1st Symp. on Comput. Geometry*, pages 161–167, 1985.
7. C. Bautista-Santiago, J. M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, J. Urrutia, and I. Ventura. Computing optimal islands. *Op. Res. Letters*, 39(4):246–251, 2011.
8. M. R. Cerioli, L. Faria, T. O. Ferreira, and F. Protti. On minimum clique partition and maximum independent set on unit disk graphs and penny graphs: Complexity and approximation. *Electronic Notes in Discrete Mathematics*, 18:73–79, 2004.
9. V. Chvátal and G. Klincsek. Finding largest convex subsets. *Congresus Numeratium*, 29:453–460, 1980.
10. G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proc. 27th Symp. on Principles of Database Systems*, pages 191–200, 2008.
11. D. Eppstein, M. Overmars, G. Rote, and G. Woeginger. Finding minimum area k-gons. *Discrete & Computational Geometry*, 7(1):45–58, 1992.
12. W. M. Getz, S. Fortmann-Roe, P. C. Cross, A. J. Lyons, S. J. Ryan, and C. C. Wilmers. Locoh: Nonparameteric kernel methods for constructing home ranges and utilization distr ibutions. *PLoS ONE*, 2(2), 02 2007.
13. W. M. Getz and C. C. Wilmers. A local nearest-neighbor convex-hull construction of home ranges and utilization distributions. *Ecography*, 27(4):489–505, 2004.
14. A. Jørgensen, M. Löffler, and J. M. Phillips. Geometric computations on indecisive and uncertain points. *CoRR*, abs/1205.0273, 2012.
15. L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C.-K. Yap. Classroom examples of robustness problems in geometric computations. *Comput. Geom.*, 40(1), 2008.
16. M. Löffler. *Data Imprecision in Computational Geometry*. PhD thesis, Utrecht University, 2009.
17. M. Löffler and M. van Kreveld. Largest and smallest convex hulls for imprecise points. *Algorithmica*, 56:235–269, 2010.
18. D. Salesin, J. Stolfi, and L. J. Guibas. Epsilon geometry: Building robust algorithms from imprecise computations. In *Symp. on Comput. Geom.*, pages 208–217, 1989.
19. C.-K. Yap and S. Pion. Special issue on robust geometric algorithms and their implementations. *Comput. Geom.*, 33(1-2), 2006.

A Proof of Lemma 1

Let S_{in} denote the sites contained by C (possibly on the boundary). Then,

$$\begin{aligned}
\mathcal{L}(C) &= \sum_{A \subseteq S \ \wedge \ \mathcal{CH}(A)=C} \pi(A) \\
&= \sum_{V \subseteq A \subseteq S_{in}} \pi(A) \\
&= \sum_{V \subseteq A \subseteq S_{in}} \left(\prod_{s_i \in A} \pi_i \times \prod_{s_i \notin A} \overline{\pi_i} \right) \\
&= \sum_{\substack{A = V \uplus A' \\ A' \subseteq (S_{in} \setminus V)}} \left(\prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i} \times \prod_{s_i \in A'} \pi_i \times \prod_{s_i \in (S_{in} \setminus V) \setminus A'} \overline{\pi_i} \right) \\
&= \prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i} \times \sum_{A' \subseteq S_{in} \setminus V} \left(\prod_{s_i \in A'} \pi_i \times \prod_{s_i \in (S_{in} \setminus V) \setminus A'} \overline{\pi_i} \right) \\
&= \prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i} \times \prod_{s_i \subseteq S_{in} \setminus V} (\pi_i + \overline{\pi_i}) \\
&= \prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i}
\end{aligned}$$

B Computing Edge Contributions

In this section, we describe how to compute the contribution of each edge in constant time after an $O(n^2)$-time preprocessing. The main idea is to utilize a modified version of a triangle query structure by [11]. In particular, we have the following lemma from [11].

Lemma 10. *Given a set P of n points in the plane, one can preprocess P in $O(n^2)$ time and space, so that the number of points in P contained by a given query triangle (with corners among P) can be reported in constant time.*

It is trivial to modify this data structure so that, under an assignment of weights to the set of points, one can report the product of the weights of the points in the query triangle. In particular, we have the following lemma.

Lemma 11. *Let P be a set of n points in the plane such that each point is assigned a weight. One can preprocess P in $O(n^2)$ time and space, so that the product of the weights of all points in P contained by a given query triangle (with corners among P) can be reported in constant time.*

We now show how to query edge contributions using this data structure. Recall that S is the set of all sites, and we want to compute edge contributions

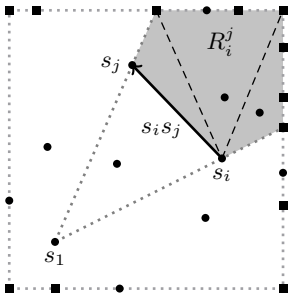

Fig. 4: Triangulating R_i^j inside the bounding box of S. The black circles are the sites in S. The black squares the points in U and V.

with respect to lowest vertex s_1. Let U be the set of the four corners of the bounding box of S. Moreover, let V be the set of points produced by intersecting the bounding box of S with the downward ray extending from s_1 and the rays $s_1 s_i$ for all s_i. Clearly, $|V| \leq n$. We construct an instance of the weighted triangle query structure on $S \cup U \cup V$. In this structure, we define the weight of each point s_i in S as its corresponding complementary probability, i.e., $\overline{\pi_i}$. For all points in U and V, we define the weight as 1.

Given an edge $s_i s_j$, we can compute its contribution as follows. The region R_i^j restricted to the bounding box of S is a polygon of constant complexity. We triangulate this polygon, and for each triangle, query the product of the weights of the points in the triangle. (See Figure 4.) The results of these queries, when multiplied, gives the product of complementary probabilities of all sites in R_i^j, which is what we need to compute $\mathcal{C}(s_i s_j)$.

C Proof of Lemma 3

Consider the triangle formed by p_i, p_j, and p_k. By Heron's formula, the area A of this triangle is at least $\sqrt{3}$ (the sides have length at least 2). Alternatively, the area can be computed as $A = bh/2$, where $b = \|p_i - p_j\| = 2$ and h is the height of triangle. Thus we get that $\|p_k - p_{ij}\| \geq h = A \geq \sqrt{3}$.

D Proof of Lemma 4

For the sake of contradiction assume that a site s_k has probability $\pi_k > \frac{1}{2}$ and is outside the most likely hull C. Let $V \subseteq S$ be the set of sites that appear on C as a vertex, and let S_{out} be the set of sites outside C. Now consider adding s_k to C. For the resulting hull C', let V' be the set of vertices of C', and let S'_{out} be the set of sites outside C'. Note that $V' \subseteq V \cup \{s_k\}$ and $S'_{out} \subseteq S_{out} \setminus \{s_k\}$. From Lemma 1 we can obtain:

$$\mathcal{L}(C') = \prod_{s_i \in V'} \pi_i \times \prod_{s_i \in S'_{out}} \overline{\pi_i} \geq \frac{\pi_k}{\overline{\pi_k}} \prod_{s_i \in V} \pi_i \times \prod_{s_i \in S_{out}} \overline{\pi_i} = \frac{\pi_k}{\overline{\pi_k}} \mathcal{L}(C) > \mathcal{L}(C)$$

This implies that C is not the most likely hull, contradicting the initial assumption.

E Proof of Lemma 5

Every plane parallel to $H(p)$ can be represented by the equation $Ax + By + C(z - h) = 0$, where (A, B, C) (with $C \neq 0$) is the normal of the plane with length 1, and h is the vertical shift from origin. By intersecting such a plane with $\mathcal{P}_1$ we obtain the equation $Ax + By + C(x^2 + y^2 - h) = 0$, which we can rewrite as $(x + \frac{A}{2C})^2 + (y + \frac{B}{2C})^2 = h + \frac{A^2+B^2}{4C^2}$. This equation describes a circle with center $(-\frac{A}{2C}, -\frac{B}{2C})$ (independent of h) and radius $r^2 = h + \frac{A^2+B^2}{4C^2}$. Since the plane is tangent to $\mathcal{P}_1$ when $r = 0$, the result follows.

F Proof of Lemma 6

Let $p \in \mathbb{R}^2$ be a point inside the quadrilateral formed by p_{ij}^e ($e \in \{N, E, S, W\}$) and let $q \in \mathcal{P}_1$ be the point obtained by lifting p. By definition, $\|p_i - p\| \leq 1 + \delta$ (same for p_j), and by Lemma 3 $\|p_k - p\| \geq \sqrt{3} - \delta$ for $k \neq i, j$. We need that u_i and u_j are below the tangent plane of $\mathcal{P}_1$ at q, and u_k ($k \neq i, j$) is above this plane. Since $\mathcal{P}_2$ is 2 units below $\mathcal{P}_1$ and by Lemma 5, u_i is below the tangent plane if and only if $\|p_i - p\| < \sqrt{2}$. The analogue holds for u_j. Similarly, we need $\|p_k - p\| > \sqrt{2}$ for all u_k. Consequently, we obtain two bounds on δ, namely $\delta < \sqrt{2} - 1$ and $\delta < \sqrt{3} - \sqrt{2}$, of which the latter is the strongest.

G Proof of Lemma 7

By changing the constant in the big O notation, we can rewrite the approximation factor as $2^{-O(n^{1-\epsilon})} = c^{O(n^{1-\epsilon})}$. For the sake of contradiction, assume that there is a polynomial-time $c^{O(n^{1-\epsilon})}$-approximation algorithm of $\mathcal{O}$, and that its output for instance I is given by the function $A(I) \in \mathcal{F}(I)$. For any instance I, let $Opt(I)$ denote its optimal solution. Now, consider any instance I of $\mathcal{O}$ and let $n = |I|$. Since $\mathcal{O}$ is product composable, we can construct an instance I^* containing $m = n^k$ copies of I. We get $|I^*| = N = n^{k+1}$ and by the bijection property of product composability $f(Opt(I^*)) = f(Opt(I))^m$. Let $(S_1, \ldots, S_m)$ (where each $S_i \in \mathcal{F}(I)$) be the matching tuple of $A(I^*)$ in the bijection. At least one solution in this tuple, say S_1, satisfies $f(S_1) \geq f(A(I^*))^{1/m}$. By assumption, $f(A(I^*)) \geq c^{O(n^{1-\epsilon})} \cdot f(Opt(I^*))$. It follows that

$$f(S_1) \geq f(A(I^*))^{1/m} \geq c^{\frac{O(n^{1-\epsilon})}{m}} \cdot f(Opt(I^*))^{1/m} = c^{\frac{O(N^{1-\epsilon})}{m}} \cdot f(Opt(I))$$

Since $m = N^{\frac{k}{k+1}}$ we can choose, for any $\epsilon > 0$, a large enough k such that $m = \omega(N^{1-\epsilon})$. For such an assignment, S_1 is computable in polynomial time (in n) and $f(S_1) \geq c \cdot f(Opt(I))$. This contradicts with the premise that there is no polynomial-time c-approximation algorithm for $\mathcal{O}$.

H Most Likely Hull Definition in the Multipoint Model

In the multipoint model, the ith point of the input is described by a tuple $((s_i^1, \pi_i^1), \ldots, (s_i^{k_i}, \pi_i^{k_i}))$, and the interpretation is that the ith point appears at the position s_i^j with probability π_i^j, for $j = 1, 2, \ldots, k_i$. If the sum of probabilities for the ith point (i.e., $\sum_{1 \leq i \leq k_i} \pi_i^j$) is not 1 (in which case it is strictly less than 1), then it is possible that the ith point does not appear at all in a probabilistic experiment.

We use S to denote the set of all sites as usual, i.e., $S = \{s_i^j\}$. For a subset $A \subseteq S$, we denote the probability that A is the outcome of a probabilistic experiment by $\pi(A)$. Similarly to the point model, the definition of $\pi(A)$ involves a product of existence probabilities for all sites in A. The sites that are not in A, however, contribute to $\pi(A)$ in a different way. Specifically, let s_i^j be a site that is not in A. If A contains another $s_i^{j'}$ site from the probabilistic tuple of the ith point, then the non-existence probability of s_i^j is irrelevant to $\pi(A)$, because existence of $s_i^{j'}$ already implies non-existence of s_i^j. If there is no such site $s_i^{j'}$, then no site from the tuple of the ith point is in A. In that case, we just consider the probability that ith point does not exist at all, which is $1 - \sum_{1 \leq j \leq k_i} \pi_i^j$. Finally, notice that if A contains two sites from the same probabilistic tuple, then it cannot be the outcome of an experiment. This implies the following definition for $\pi(A)$:

$$\pi(A) = \begin{cases} 0 & \text{if there are two distinct sites } s_i^j \text{ and } s_i^{j'} \text{ in } A \\ \displaystyle\prod_{s_i^j \in A} \pi_i^j \times \prod_{i \,|\, \nexists j . s_i^j \in A} \left(1 - \sum_{1 \leq j \leq k_i} \pi_i^j\right) & \text{otherwise} \end{cases}$$

The definition for the most likely hull follows from $\pi(A)$ as in the point model case. That is, the most likely convex hull is the polytope C which maximizes the likelihood function $\mathcal{L}(C)$, which is defined as

$$\mathcal{L}(C) \;=\; \Pr\big[\,\mathcal{CH}(A) \equiv C\,\big] = \sum_{\substack{A \subseteq S \\ \mathcal{CH}(A) \equiv C}} \pi(A)$$

I Proof of Lemma 8

Let C be the most likely hull. For ease of reference, let us say that the outcome A of a probabilistic experiment is compatible with C if $\mathcal{CH}(A) = C$. Notice that all experiment outcomes A compatible with C contain a particular configuration of the dependent point pairs. In particular, if C contains a point t_v as a vertex, then all compatible outcomes contain t_v and not f_v. Otherwise, all compatible outcomes contain f_v and not t_v. Similarly, if C contains q_v^u as a vertex, then all compatible outcomes contain q_v^u or p_v^u otherwise. The probability that these configurations exists in the outcome of an experiment is $(1/2)^{3|U|+|V|}$

because there are $3|U| + |V|$ dependent point pairs. If C contains all spikes in its interior, then the existence of spikes are irrelevant to the likelihood of C, thus $\mathcal{L}(C) = (1/2)^{3|U|+|V|}$. Otherwise, compatibility with C is also conditioned on either existence or non-existence of at least one spike. This implies $\mathcal{L}(C) \leq (1/2)^{3|U|+|V|+1}$.

J Proof of Lemma 9

We first show that if the most likely hull has likelihood $(1/2)^{3|U|+|V|}$ then the 3-SAT instance is satisfiable. Let C be the most likely hull with likelihood $(1/2)^{3|U|+|V|}$. By construction, C contains exactly one of the sites t_v and f_v as a vertex for each variable v. Consider the boolean assignment where we assign the variable v to true if t_v is a vertex, and to false if f_v is a vertex. We now argue that this assignment satisfies all clauses in the 3-SAT instance. Take any clause u. By Lemma 8, C contains all spikes in its interior. Consequently at least one point q_v^u is a vertex of C. Then, the dependent point p_v^u is not a vertex of C. By construction, C covers the underlying spike with t_v if v appears in positive form in u or with f_v if v appears in negated form. This implies that u is satisfied by the assignment of v.

We now prove the converse. Suppose that there is a satisfying variable assignment for the 3-SAT instance. We construct a subset Q of points as follows. We insert to Q t_v if v is assigned true and f_v if v is assigned false. Additionally, for each variable-clause pair (v, u) we insert q_v^u if v is a satisfying variable for u or p_v^u otherwise. Finally, we insert all anchor points. Observe that Q is a valid outcome of a probabilistic experiment. We now argue that the convex hull of Q covers all spikes and thus has likelihood $(1/2)^{3|U|+|V|}$ by Lemma 8. The spikes under all points $p_v^u \in Q$ are trivially covered. For each point $p_v^u \notin Q$, v is a satisfying variable for u, and thus the spike under p_v^u is covered by either t_v or f_v (whichever is the one above p_v^u). Finally, since all clauses are satisfied, each spike under a triplet of points q_v^u, $q_{v'}^u$ and $q_{v''}^u$ are also covered (at least by one of them). This completes the proof.