Shape Fitting on Point Sets with Probability Distributions

Maarten Löffler

Jeff Phillips

Technical Report UU-CS-2009-013
June 2009

Department of Information and Computing Sciences
Utrecht University, Utrecht, The Netherlands
www.cs.uu.nl

lSSN: 0924-3275

Department of Information and Computing Sciences
Utrecht University
P.O. Box 80.089
3508 TB Utrecht
The Netherlands

Shape Fitting on Point Sets with Probability Distributions

Maarten Löffler
Utrecht University
loffler@cs.uu.nl

Jeff Phillips
Duke University
jeffp@cs.duke.edu

June 25, 2009

Abstract

We consider problems on data sets where each data point has uncertainty described by an individual probability distribution. We develop several frameworks and algorithms for calculating statistics on these uncertain data sets. Our examples focus on geometric shape fitting problems. We prove approximation guarantees for the algorithms with respect to the full probability distributions. We then empirically demonstrate that our algorithms are simple and practical, solving for a constant hidden by asymptotic analysis so that a user can reliably trade speed and size for accuracy.

1 Introduction

This paper deals with data sets where each "data point" is actually a distribution of where a data point may be. We focus on geometric problems on this data, however, many of the ideas, data structures, and algorithms extend to non-geometric problems. Since the input data we consider has uncertainty, given by probability distributions, we argue that computing exact answers may not be worth the effort. Furthermore, many problems we consider may not have compact closed form solutions. As a result, we produce approximate answers.

Sensed Data

In gathering data there is a trade-off between quantity and accuracy. The drop in the price of hard drives and other storage costs has shifted this balance towards gathering enormous quantities of data, yet with noticeable and sometimes intentional imprecision. However, often as a benefit from the large data sets, models are developed to describe the pattern of the data error.

Let us take as an example Light Detection and Ranging (LIDAR) data gathered for Geographic Information Systems (GIS) [27], specifically height values at millions of locations on a terrain. Each data point (x, y, z) has an x-value (longitude), a y-value (latitude), and a z-value (height). This data set is gathered by a small plane flying over a terrain with a laser aimed at the ground measuring the distance from the plane to the ground. Error can occur due to inaccurate estimation of the plane's altitude and position or artifacts on the ground distorting the laser's distance reading. But these errors are well-studied and can be modeled by replacing each data point with a probability distribution of its actual position. Greatly simplifying, we could represent each data point as a 3-variate normal distribution centered at its recorded value.

Similarly, large data sets are gathered and maintained for many other applications. In robotic mapping [39, 16] error models are provided for data points gathered by laser range finders and other sources. In data mining [1, 5] original data (such as published medical data) are often perturbed by a known model to preserve anonymity. In spatial databases [20, 37, 13] large data sets may be summarized as probability distributions to store them more compactly. Sensor networks [15] stream in large data sets collected by cheap and thus inaccurate sensors. In protein structure determination [35] every atom's position is imprecise due to inaccuracies in reconstruction techniques and the inherent flexibility in the protein. In summary, there are many large data sets with modeled errors and dynamic updates.

However, much raw data is not immediately given as a set of probability distributions, rather as a set of points, each essentially drawn from a probability distribution itself. Approximate algorithms may treat this data as exact, construct an approximate answer, and then postulate that since the raw data is not exact and has inaccuracies, the approximation errors made by the algorithm may be similar to the inaccuracies of the imprecise input data. This is a very dangerous postulation, as demonstrated by the following example.

Example. *Consider a robot trying to determine the boundary of a convex room. Its strategy is to use a laser range finder to get data points on objects in the room (hopefully boundary walls), and then take the convex hull of these points.*

However, large errors may occur if the room has windows; a few laser scans may not bounce off the window, and thus return data points (say, 100 *meters) outside the room. Standard techniques (e.g., α-kernels) would include those points in the convex hull, but may allow some approximation (say, up to* 10 *meters). Hence, the outlier data points could still dramatically warp the shape of the room, outside the error tolerance.*

However, an error model on these outlier data points, through regression to the mean, would assign some probability to them being approximately correct and some probability to them actually being inside (or much closer to) the true room. An algorithm which took this error model into account would assign some

probability of a shape near the true room shape and some probability to the oblong room that extends out through the window.

It is clear from this example, that an algorithm can only provide answers as good as the raw data *and* the models for error on that data. This paper is not about how to construct error models, but how to take error models into account. While many existing algorithms produce approximations with respect only to the raw input data, algorithms in this paper approximate with respect to the raw input data and the error models associated with them.

Other geometric error models. The input for a typical computational geometry problem is a set P of n points in $\mathbb{R}^2$, or more generally $\mathbb{R}^d$. Traditionally, such a set of points is assumed to be known exactly, and indeed, in the 1980s and 1990s such an assumption was often justified because much of the input data was hand-constructed for computer graphics or simulations. However, in many modern applications the input is sensed from the real world, and such data is inherently imprecise. Therefore, there is a growing need for methods that are able to deal with imprecision.

An early model to quantify imprecision in geometric data, motivated by finite precision of coordinates, is ε*-geometry*, introduced by Guibas *et al.* [18]. In this model, the input is given by a traditional point set P, where the imprecision is modeled by a single extra parameter ε. The true point set is not known, but it is certain that for each point in P there is a point in the disk of radius ε around it. This model has proven fruitful and is still often used due to its simplicity. To name a few examples, Guibas *et al.* [19] define *strongly convex* polygons: polygons that are guaranteed to stay convex, even when the vertices are perturbed by ε. Bandyopadhyay and Snoeyink [7] compute the set of all potential simplices in $\mathbb{R}^2$ and $\mathbb{R}^3$ that could belong to the Delaunay triangulation. Held and Mitchell [23] and Löffler and Snoeyink [28] study the problem of preprocessing a set of imprecise points under this model, so that when the true points are specified later some computation can be done faster.

A more involved model for imprecision can be obtained by not specifying a single ε for all the points, but allowing a different radius for each point, or even other shapes of imprecision regions. This allows for modeling imprecision that comes from different sources, independent imprecision in different dimensions of the input, etc. This extra freedom in modeling comes at the price of more involved algorithmic solutions, but still many results are available. Nagai and Tokura [32] compute the union and intersection of all possible convex hulls to obtain bounds on any possible solution, as does Ostrovsky-Berman and Joskowicz [33] in a setting allowing some dependence between points. Van Kreveld and Löffler [40] study the problem of computing the smallest and largest possible values of several geometric extent measures, such as the diameter or the radius of the smallest enclosing ball, where the points are restricted to lie in given regions in the plane. Kruger [25] extends some of these results to higher dimensions.

These models, in general, give worst case bounds on error, for instance upper and lower bounds on the radius of the minimum enclosing ball. When the error is derived entirely from precision errors, this information can be quite useful (as much of theoretical computer science is based on worst case bounds). However, when data is sensed, the maximum error range used as input are often manufactured by truncating a probability distribution, so the probability that a point is outside that range is below some threshold. Since the above models usually produce algorithms and answers very dependent on boundary cases, these artificial (and sometimes arbitrary) thresholds play large roles in the answers. Furthermore, the true location of the data points are often not near the boundary of the error range, but near the center. Hence, it makes more sense to use the original probability distributions, and then if needed, we can apply a threshold based on probability to the final solution. This ensures that the truncation errors have not accumulated.

This paper studies the computation of extent measures on uncertain point sets governed by probability distributions. Unsurprisingly, directly using the probability distribution error model creates harder algorithmic problems, and many questions may be impossible to answer exactly under this model. But since the

data is imprecise to begin with, it is also reasonable to construct approximate answers. Our algorithms have approximation guarantees with respect to the original distributions, not an approximation of them. This model of uncertain data has been studied in the database community but for different types of problems (e.g. indexing[38, 24] and nearest neighbor[12]) and approximation guarantees. We focus on computing statistics on uncertain point sets, specifically shape fitting problems in a way that allows the uncertain data problem to be reduced to well-studied techniques on discrete point sets.

1.1 Problem Statement

Let $\mu_p : \mathbb{R}^d \to \mathbb{R}^+$ describe the probability distribution of a point p where the integral $\int_{q \in \mathbb{R}^d} \mu_p(q)\, dq = 1$. Let $\mu_P : \mathbb{R}^d \times \mathbb{R}^d \times \ldots \times \mathbb{R}^d \to \mathbb{R}^+$ describe the distribution of a point set P by the joint probability over each $p \in P$. For brevity we write the space $\mathbb{R}^d \times \ldots \times \mathbb{R}^d$ as $\mathbb{R}^{dn}$. For this paper we will assume $\mu_P(q_1, q_2, \ldots, q_n) = \prod_{i=1}^{n} \mu_{p_i}(q_i)$, so the distribution for each point is independent, although this restriction can be easily circumvented.

Given a distribution μ_P we ask a variety of shape fitting questions about the uncertain point set. For instance, what is the radius of the smallest enclosing ball or what is the smallest axis-aligned bounding box of an uncertain point set. In the presence of imprecision, the answer to such a question is not a single value or structure, but also a *distribution* of answers. The focus of this paper is not just how to answer such shape fitting questions about these distributions, but how to concisely represent them. As a result, we introduce two types of approximate distributions as answers, and a technique to construct coresets for these answers.

ε-Quantizations. Let $f : \mathbb{R}^{dn} \to \mathbb{R}^k$ be a function on a fixed point set. Examples include the radius of the minimum enclosing ball where $k = 1$ and the width of the minimum enclosing axis-aligned rectangle along the x-axis and y-axis where $k = 2$. Define the "dominates" binary operator $\preceq$ so that $(p_1, \ldots, p_k) \preceq (v_1, \ldots, v_k)$ is true if for every coordinate $p_i \leq v_i$. Let $\mathbb{X}_f(v) = \{Q \in \mathbb{R}^{dn} \mid f(Q) \preceq v\}$. For a query value v define,

$$F_{\mu_P}(v) = \int_{Q \in \mathbb{X}_f(v)} \mu_P(Q)\, dQ.$$

Then F_{μ_P} is the cumulative density function of the distribution of possible values that f can take[1]. Ideally, we would return the function F_{μ_P} so we could quickly answer any query exactly, however, it is not clear how to calculate $F_{\mu_P}(v)$ exactly for even a single query value v. Rather, we introduce a data structure, which

[1]For a function f and a distribution of point sets μ_P, we will always represent the cumulative density function of f over μ_P by F_{μ_P}.

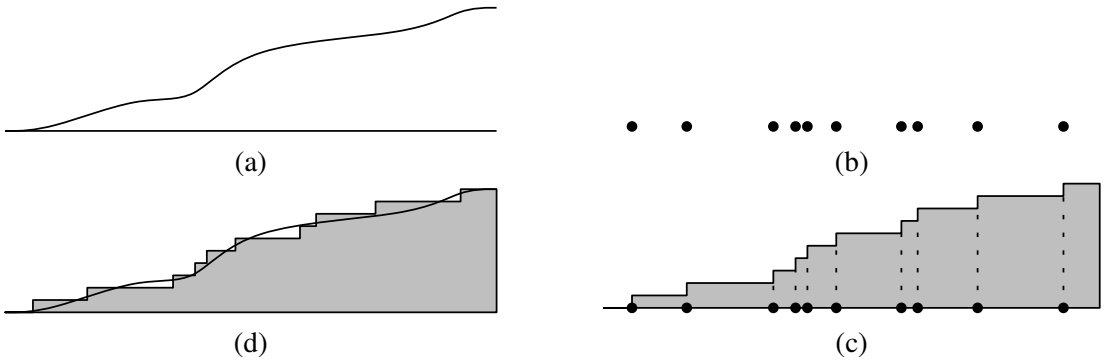
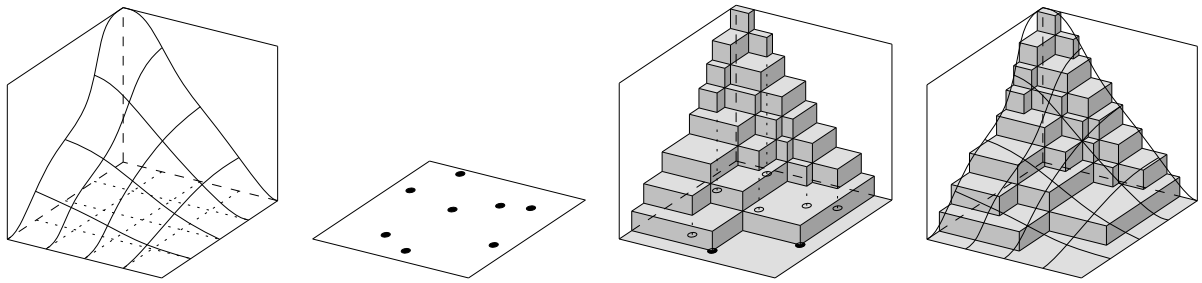


Figure 1: (a) The true form of a monotonically increasing function from $\mathbb{R} \to \mathbb{R}$. (b) The ε-quantization R as a point set in $\mathbb{R}$. (c) The inferred curve h_R in $\mathbb{R}^2$. (d) Overlay of the two images.

we call an ε-quantization, to answer any such query approximately and efficiently, illustrated in Figure 1 for $k = 1$. An *ε-quantization* is a point set $R \subset \mathbb{R}^k$ which induces a function h_R where $h_R(v)$ describes the fraction of points in R that v dominates. Let $R_v = \{r \in R \mid r \preceq v\}$. Then $h_R(v) = |R_v|/|R|$. For an isotonic (monotonically increasing in each coordinate) function F_{μ_P} and any value v, an ε-quantization, R, guarantees that

$$|h_R(v) - F_{\mu_P}(v)| \leq \varepsilon.$$

More generally (and, for brevity, usually only when $k > 1$), we say R is a k-variate ε-quantization. An example of a 2-variate ε-quantization is shown in Figure 2. The space required to store the data structure for R is dependent only on ε and k, not on $|P|$ or μ_P.



(a) (b) (c) (d)

Figure 2: (a) The true form of a 2-variate function. (b) The ε-quantization R as a point set in $\mathbb{R}^2$. (c) The inferred surface h_R in $\mathbb{R}^3$. (d) Overlay of the two images.

$(\varepsilon, \delta, \alpha)$-Kernels. Rather than compute a new data structure for each measure we are interested in, we can also compute a single data structure (a coreset) that allows us to answer many types of questions. For an isotonic function $F_{\mu_P} : \mathbb{R}^+ \to [0, 1]$, an *$(\varepsilon, \alpha)$-quantization* data structure M describes a function $h_M : \mathbb{R}^+ \to [0, 1]$ so for any $x \in \mathbb{R}^+$, there is an $x' \in \mathbb{R}^+$ such that (1) $|x - x'| \leq \alpha x$ and (2) $|h_M(x) - F_{\mu_P}(x')| \leq \varepsilon$. An *$(\varepsilon, \delta, \alpha)$-kernel* is a data structure that can produce an (ε, α)-quantization, with probability at least $1 - \delta$, for F_{μ_P} where f measures the width in any direction and whose size depends only on ε, α, and δ. The notion of (ε, α)-quantizations is generalized to a k-variate version, as are $(\varepsilon, \delta, \alpha)$-kernels, in Section 2.2.

Shape inclusion probabilities. A *summarizing shape* of a point set $P \subset \mathbb{R}^d$ is a Lebesgue-measureable subset of $\mathbb{R}^d$ that is determined by P. Examples include the smallest enclosing ball, the minimum-volume axis-aligned bounding box, or the convex hull. We consider some class of shapes $\mathcal{S}$ and the summarizing shape $S(P) \in \mathcal{S}$ is the shape from $\mathcal{S}$ that is optimized in some aspect with respect to P. For a family of summarizing shapes $\mathcal{S}$ we can study the *shape inclusion probability function* $s_{\mu_P} : \mathbb{R}^d \to [0, 1]$ (or sip function), where $s_{\mu_P}(q)$ describes the probability that a query point $q \in \mathbb{R}^d$ is included in the summarizing shape[2]. There does not seem to be a closed form for many of these functions. Rather we calculate an ε-sip function $\hat{s} : \mathbb{R}^d \to [0, 1]$ such that $\forall_{q \in \mathbb{R}^d} |s_{\mu_P}(q) - \hat{s}(q)| \leq \varepsilon$. The space required to store an ε-sip function depends only on ε and the complexity of the summarizing shape.

1.2 Contributions

We describe simple and practical randomized algorithms for the computation of ε-quantizations, $(\varepsilon, \delta, \alpha)$-kernels, and ε-sip functions. Let $T_f(n)$ be the time it takes to calculate a summarizing shape of a set of n

[2]For technical reasons, if there are (degenerately) multiple optimal summarizing shapes, we say each is equally likely to be the summarizing shape of the point set.

points $Q \subset \mathbb{R}^d$, which generates a statistic $f(Q)$ (e.g., radius of smallest enclosing ball). We can calculate an ε-quantization of F_{μ_P}, with probability at least $1-\delta$, in time $O(T_f(n)(1/\varepsilon^2)\log(1/\delta))$. For univariate ε-quantizations the size is $O(1/\varepsilon)$, and for k-variate ε-quantizations the size is $O(k^2(1/\varepsilon)\log^{2k}(1/\varepsilon))$. We can calculate an $(\varepsilon,\delta,\alpha)$-kernel of size $O((1/\alpha^{(d-1)/2})\cdot(1/\varepsilon^2)\log(1/\delta))$ in $O((n+(1/\alpha^{d-3/2}))(1/\varepsilon^2)\log(1/\delta))$ time. With probability at least $1-\delta$, we can calculate an ε-sip function of size $O((1/\varepsilon^2)\log(1/\delta))$ in time $O(T_f(n)(1/\varepsilon^2)\log(1/\delta))$.

All of these randomized algorithms are simple and practical, as demonstrated by a series of experimental results. In particular, we show that the constant hidden by the big-O notation is in practice at most 0.5 for all algorithms.

This paper describes results for shape fitting problems for distributions of point sets in $\mathbb{R}^d$, in particular, we will use the smallest enclosing ball and the axis-aligned bounding box as running examples in the algorithm descriptions. The concept of ε-quantizations extends to many other problems with uncertain data. In fact, variations of our randomized algorithm will work for a more general array of problems.

1.3 Preliminaries: ε-Samples and α-Kernels

ε-Samples. For a set P let $\mathcal{A}$ be a set of subsets of P. In our context usually P will be a point set and the subsets in $\mathcal{A}$ could be induced by containment in a shape from some family of geometric shapes. For some examples of $\mathcal{A}$, let $\mathcal{B}_r$ describe all subsets of P determined by containment in some ball of radius r; let $\mathcal{R}_d$ describe all subsets of P defined by containment in some d-dimensional axis-aligned box; let $\mathcal{H}$ describe all subsets of P defined by containment in some halfspace. We use $\mathcal{A}$ generically to represent one such family of ranges.

The pair $(P,\mathcal{A})$ is called a *range space*. We say that $Q \subset P$ is an *ε-sample* of $(P,\mathcal{A})$ if

$$\forall_{R\in\mathcal{A}} \left| \frac{\phi(R\cap Q)}{\phi(Q)} - \frac{\phi(R\cap P)}{\phi(P)} \right| \leq \varepsilon,$$

where $|\cdot|$ takes the absolute value and $\phi(\cdot)$ returns the measure of a point set. In the discrete case $\phi(Q)$ returns the cardinality of Q. We say $\mathcal{A}$ *shatters* a set S if every subset of S is equal to $R\cap S$ for some $R\in\mathcal{A}$. The cardinality of the largest discrete set $S\subseteq P$ that $\mathcal{A}$ can shatter is known as the *VC-dimension* of $(P,\mathcal{A})$.

When $(P,\mathcal{A})$ has constant VC-dimension ν, we can create an ε-sample Q of $(P,\mathcal{A})$, with probability $1-\delta$, by uniformly sampling $O((1/\varepsilon^2)(\nu+\log(1/\delta)))$ points from P [41, 26]. There exist deterministic techniques to create ε-samples [29, 11] of size $O(\nu(1/\varepsilon^2)\log(1/\varepsilon))$ in time $O(\nu^{3\nu}n((1/\varepsilon^2)\log(\nu/\varepsilon))^\nu)$. There exist ε-samples of smaller sizes [31], but direct, efficient constructions are not known. When P is a point set in $\mathbb{R}^d$ and the family of ranges $\mathcal{Q}_k$ is determined by inclusion of convex shapes whose sides have one of k predefined normal directions, such as the set of axis-aligned boxes, then an ε-sample for $(P,\mathcal{Q}_k)$ of size $O((k/\varepsilon)\log^{2k}(1/\varepsilon))$ can be constructed in $O((n/\varepsilon^3)\log^{6k}(1/\varepsilon))$ time [34]. If $(P,\mathcal{A})$ has VC-dimension ν, this also implies that $(P,\mathcal{A})$ contains at most $|P|^\nu$ sets.

For a range space $(P,\mathcal{A})$ the *dual range space* is defined $(\mathcal{A},P^*)$ where P^* is all subsets $\mathcal{A}_p\subseteq\mathcal{A}$ defined for an element $p\in P$ such that $\mathcal{A}_p=\{A\in\mathcal{A}\mid p\in A\}$. If $(P,\mathcal{A})$ has VC-dimension ν, then $(\mathcal{A},P^*)$ has VC-dimension $\leq 2^{\nu+1}$. Thus, if the VC-dimension of $(\mathcal{A},P^*)$ is constant, then the VC-dimension of $(P,\mathcal{A})$ is also constant [30]. Hence, the standard ε-sample theorems apply to dual range spaces as well.

When we have a distribution $\mu:\mathbb{R}^d\to\mathbb{R}^+$, such that $\int_{x\in\mathbb{R}}\mu(x)\,dx=1$, we can think of this as the set P of all points in $\mathbb{R}^d$, where the weight w of a point $p\in\mathbb{R}^d$ is $\mu(p)$. Hence, if a point is randomly selected from P proportional to its weight w, then it is equivalent to selecting a point at random from the distribution μ. To simplify notation, we write $(\mu,\mathcal{A})$ as a range space where the ground set is this set $P=\mathbb{R}^d$ weighted by the distribution μ.

Let $g : \mathbb{R} \to \mathbb{R}^+$ be a function where $\int_{x=-\infty}^{\infty} g(x)\ dx = 1$. We can create an ε-sample Q_g of $(g, \mathcal{I}_+)$, where $\mathcal{I}_+$ describes the set of all one-sided intervals of the form $(-\infty, t)$, so that

$$\max_t \left| \int_{x=-\infty}^{t} g(x)\ dx - \frac{1}{|Q_g|} \sum_{q \in Q_g} 1(q < t) \right| \leq \varepsilon.$$

We can construct Q_g of size $O(\frac{1}{\varepsilon})$ by choosing a set of points in Q_g so that the integral between two consecutive points is always ε. But we do not need to be so precise. Consider the set of $\lceil 2/\varepsilon \rceil$ points $\{q'_1, q'_2, \ldots, q'_{\lceil 2/\varepsilon \rceil}\}$ such that $\int_{x=-\infty}^{q'_i} = i\varepsilon/2$. Any set of $\lceil 2/\varepsilon \rceil$ points $Q_g = \{q_1, q_2, \ldots, q_{\lceil 2/\varepsilon \rceil}\}$ such that $q'_i \leq q_i \leq q'_{i+1}$ is an ε-sample.

α-Kernels. Given a point set $P \in \mathbb{R}^d$ of size n and a direction $u \in \mathbb{S}^{d-1}$, let $P[u] = \arg\max_{p \in P} \langle p, u \rangle$, where $\langle \cdot, \cdot \rangle$ is the inner product operator. Let $\omega(P, u) = \langle P[u] - P[-u], u \rangle$ describe the width of P in direction u. We say that $K \subseteq P$ is an *α-kernel* of P if for all $u \in \mathbb{S}^{d-1}$

$$\omega(P, u) - \omega(K, u) \leq \alpha \cdot \omega(P, u).$$

α-kernels of size $O(1/\alpha^{(d-1)/2})$ [3] can be calculated in time $O(n + 1/\alpha^{d-3/2})$ [9, 42]. Computing many extent related problems such as diameter and smallest enclosing ball on the α-kernel approximates the function on the original set [3, 2, 9, 10].

2 Randomized Algorithm for ε-Quantizations

We start with a general algorithm (Algorithm 2.1) which will be made specific in several places in the paper. We only assume that we can draw a random point from μ_p for each $p \in P$ in constant time; if the time depends on some other parameters, the time complexity of the algorithms can be easily adjusted.

Algorithm 2.1 Approximate μ_P with regard to a family of shapes $\mathcal{S}$ or function $f_{\mathcal{S}}$

1: **for** $i = 1$ **to** $m = O((1/\varepsilon^2)(\nu + \log(1/\delta)))$ **do**
2: **for all** $p_j \in P$ **do**
3: Sample q_j from μ_{p_j}.
4: Set $V_i = f_{\mathcal{S}}(\{q_1, q_2, \ldots, q_n\})$.
5: Reduce or Simplify the set $\mathcal{V} = \{V_i\}_{i=1}^m$.

2.1 Algorithm for ε-Quantizations

For a function f on a point set P of size n, it takes $T_f(n)$ time to evaluate $f(P)$. We now construct an approximation to F_{μ_P} by adapting Algorithm 2.1 as follows. First draw a sample point q_j from each μ_{p_j} for $p_j \in P$, then evaluate $V_i = f(\{q_1, \ldots, q_n\})$. The fraction of trials of this process that produces a value dominated by v is the estimate of $F_{\mu_P}(v)$. In the univariate case we can reduce the size of $\mathcal{V}$ by returning $2/\varepsilon$ evenly spaced points according to the sorted order.

Theorem 2.1. *Let $T_f(n)$ be the time it takes to compute $f(Q)$ for any point set Q of size n. For a distribution μ_P of n points, with success probability at least $1 - \delta$, there exists an ε-quantization of size $O(1/\varepsilon)$ for F_{μ_P}, and it can be constructed in $O(T_f(n)(1/\varepsilon^2)\log(1/\delta))$ time.*

Proof. Because $F_{\mu_P} : \mathbb{R} \to [0,1]$ is an isotonic function, there exists another function $g : \mathbb{R} \to \mathbb{R}^+$ such that $F_{\mu_P}(t) = \int_{x=-\infty}^{t} g(x)\ dx$ where $\int_{x\in\mathbb{R}} g(x)\ dx = 1$. Thus g is a probability distribution of the values of f given inputs drawn from μ_P. This implies that an ε-sample of $(g, \mathcal{I}_+)$ is an ε-quantization of F_{μ_P}, since both estimate within ε the fraction of points in any range of the form $(-\infty, x)$.

By drawing a random sample q_i from each μ_{p_i} for $p_i \in P$, we are drawing a random point set Q from μ_P. Thus $f(Q)$ is a random sample from g. Hence, using the standard randomized construction for ε-samples, $O((1/\varepsilon^2)\log(1/\delta))$ such samples will generate an $(\varepsilon/2)$-sample for g, and hence an $(\varepsilon/2)$-quantization for F_{μ_P}, with probability at least $1-\delta$.

Since in an $(\varepsilon/2)$-quantization R every value $h_R(v)$ is different from $F_{\mu_P}(v)$ by at most $\varepsilon/2$, then we can take an $(\varepsilon/2)$-quantization of the function described by $h_R(\cdot)$ and still have an ε-quantization of F_{μ_P}. Thus, we can reduce this to an ε-quantization of size $O(1/\varepsilon)$ by taking a subset of $2/\varepsilon$ points spaced evenly according to their sorted order. □

We can construct k-variate ε-quantizations using the same basic procedure as in Algorithm 2.1. The output V_i of $f_{\mathbb{S}}$ is k-variate and thus results in a k-dimensional point.

Theorem 2.2. *Let $T_f(n)$ be the time it takes to compute $f(Q)$ for any point set Q of size n. Given a distribution μ_P of n points, with success probability at least $1-\delta$, we can construct a k-variate ε-quantization for F_{μ_P} of size $O((k/\varepsilon^2)(k+\log(1/\delta)))$ and in time $O(T_f(n)(1/\varepsilon^2)(k+\log(1/\delta)))$.*

Proof. Let $\mathcal{R}_+$ describe the family of ranges where a range $A_p = \{q \in \mathbb{R}^k \mid q \preceq p\}$. In the k-variate case there exists a function $g : \mathbb{R}^k \to \mathbb{R}^+$ such that $F_{\mu_P}(v) = \int_{x\preceq v} g(x)\ dx$ where $\int_{x\in\mathbb{R}^k} g(x)\ dx = 1$. Thus g describes the probability distribution of the values of f, given inputs drawn randomly from μ_P. Hence a random point set Q from μ_P, evaluated as $f(Q)$, is still a random sample from the k-variate distribution described by g. Thus, with probability at least $1-\delta$, a set of $O((1/\varepsilon^2)(k+\log(1/\delta)))$ such samples is an ε-sample of $(g, \mathcal{R}_+)$, which has VC-dimension k, and the samples are also a k-variate ε-quantization of F_{μ_P}. □

We can then reduce the size of the ε-quantization R to $O((k^2/\varepsilon)\log^{2k}(1/\varepsilon))$ in $O(|R|(k/\varepsilon^3)\log^{6k}(1/\varepsilon))$ time [34] or to $O((k^2/\varepsilon^2)\log(1/\varepsilon))$ in $O(|R|(k^{3k}/\varepsilon^{2k})\cdot\log^k(k/\varepsilon))$ time [11], since the VC-dimension is k and each data point requires $O(k)$ storage. However, we do not investigate the empirical performance of these deterministic algorithms in this paper. See [6] for an empirical study of alternatives to [11].

2.2 $(\varepsilon, \delta, \alpha)$-Kernels

The above construction works for a fixed family of summarizing shapes. In this section, we show how to build a single data structure, an $(\varepsilon, \delta, \alpha)$-*kernel*, for a distribution μ_P in $\mathbb{R}^d$ that can be used to construct (ε, α)-quantizations for several families of summarizing shapes. In particular, an $(\varepsilon, \delta, \alpha)$-kernel of μ_P is a data structure such that in any query direction $u \in \mathbb{S}^{d-1}$, with probability at least $1-\delta$, we can create an (ε, α)-quantization for the cumulative density function of $\omega(\cdot, u)$, the width in direction u. This data structure introduces a parameter α, which deals with relative geometric error, in addition to the error parameter ε, which deals with relative counting error and error parameter δ which accounts for potential error due to randomization.

We follow the randomized framework described above as follows. The desired $(\varepsilon, \delta, \alpha)$-kernel $\mathcal{K}$ consists of a set of $m = O((1/\varepsilon^2)\log(1/\delta))$ $(\alpha/2)$-kernels, $\{K_1, K_2, \ldots, K_m\}$, where each K_j is an $(\alpha/2)$-kernel of a point set Q_j drawn randomly from μ_P. Given $\mathcal{K}$, with probability at least $1-\delta$ we can then create an (ε, α)-quantization for the cumulative density function of width over μ_P in any direction $u \in \mathbb{S}^{d-1}$. Specifically, let $M = \{\omega(K_j, u)\}_{j=1}^m$.

Lemma 2.1. *With probability at least $1-\delta$, M is an (ε,α)-quantization for the cumulative density function of the width of μ_P in direction u.*

Proof. The width $\omega(Q_j,u)$ of a random point set Q_j drawn from μ_P is a random sample from the distribution over widths of μ_P in direction u. Thus, with probability at least $1-\delta$, m such random samples would create an ε-quantization. Using the width of the α-kernels K_j instead of Q_j induces an error on each random sample of at most $2\alpha\cdot\omega(Q_j,u)$. Then for a query width w, say there are γm point sets Q_j that have width at most w and $\gamma' m$ α-kernels K_j with width at most w; see Figure 3. Note that $\gamma'>\gamma$. Let $\hat{w}=w-2\alpha w$. For each point set Q_j that has width greater than w it follows that K_j has width greater than $\hat{w}$. Thus the number of α-kernels K_j that have width at most $\hat{w}$ is at most γm, and thus there is a width w' between w and $\hat{w}$ such that the number of α-kernels at most w' is exactly γm. □

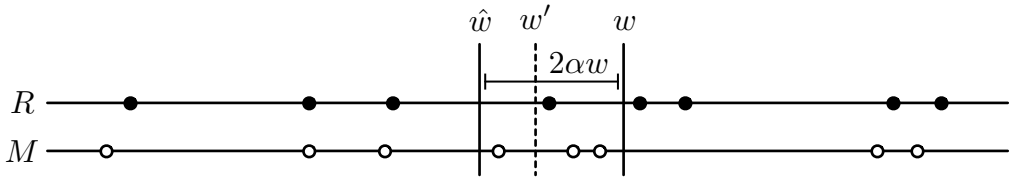


Figure 3: (ε,α)-quantization M (white circles) and ε-quantization R (black circles) given a query width w.

Since each K_j can be computed in $O(n+1/\alpha^{d-3/2})$ time, we obtain the following.

Theorem 2.3. *We can construct an $(\varepsilon,\delta,\alpha)$-kernel for μ_P on n points in $\mathbb{R}^d$ of size $O((1/\alpha^{(d-1)/2})(1/\varepsilon^2)\cdot\log(1/\delta))$ and in time $O((n+1/\alpha^{d-3/2})\cdot(1/\varepsilon^2)\log(1/\delta))$.*

k-Dependent $(\varepsilon,\delta,\alpha)$-Kernels. The definition of (ε,α)-quantizations can be extended to a *k-variate (ε,α)-quantizations* data structure with the following properties. A k-variate ε-quantization M is a set of points in $\mathbb{R}^k$ which induces a function $h_M:\mathbb{R}^k\to[0,1]$ where a query $h_M(x)=|M_x|/|M|$ returns the fraction of points in M which are dominated by or equal to x. Let $x^{(i)}$ represent the ith coordinate of a point $x\in\mathbb{R}^k$. For a query $x\in\mathbb{R}^k$, there exists a point $x'\in\mathbb{R}^k$ such that (1) for all integers $i\in[1,k]$ $|x^{(i)}-(x')^{(i)}|\le\alpha x^{(i)}$ and (2) $|M(x)-F_{\mu_P}(x')|\le\varepsilon$.

In addition, $(\varepsilon,\delta,\alpha)$-kernels can be generalized to approximate cumulative density functions of other functions $f:\mathbb{R}^{dn}\to\mathbb{R}^k$, specified as follows. We say a point $p'\in\mathbb{R}^k$ is a *relative θ-approximation* of $p\in\mathbb{R}^k$ if for each coordinate i we have $|p^{(i)}-p'^{(i)}|\le\theta p^{(i)}$. For a parameter $a\in[0,1]$, we say that f is *relative $\theta(\alpha)$-approximable* if for all $Q\in\mathbb{R}^{dn}$ and for any α-kernel K of Q, $f(K)$ is a relative $\theta(\alpha)$-approximation of $f(Q)$.

By setting $m=O((1/\varepsilon^2)(k+\log(1/\delta)))$ in the above algorithm, we can build a *k-dependent $(\varepsilon,\delta,\alpha)$-kernel* data structure $\mathcal{K}$ with the following properties. It has size $O((1/\alpha^{(d-1)/2})(1/\varepsilon^2)(k+\log(1/\delta)))$ and can be built in time $O((n+1/\alpha^{d-3/2})(1/\varepsilon^2)\cdot(k+\log(1/\delta)))$. To create a k-variate (ε,α)-quantization for a function f (with probability at least $1-\delta$), create a k-dimensional point $p_j=f(K_j)$ for each α-kernel K_j in $\mathcal{K}$. The set of m k-dimensional points forms the k-variate (ε,α)-quantization M.

Theorem 2.4. *Given a distribution μ_P of n points in $\mathbb{R}^d$, for $m=O((1/\varepsilon^2)(k+\log(1/\delta)))$, we can create a k-dependent $(\varepsilon,\delta,\alpha)$-kernel $\mathcal{K}$ of size $O((1/\alpha^{(d-1)/2})m)$ and in time $O((n+1/\alpha^{d-3/2})m)$. Let f be any relative $\theta(\alpha)$-approximable function that takes $T_f(N)$ time to evaluate on a set of N points. From $\mathcal{K}$, we can create a k-variate $(\varepsilon,\theta(\alpha))$-quantization of F_{μ_P} of size $O((k/\varepsilon^2)\log(1/\delta))$ and in time $O(T_f(1/\alpha^{(d-1)/2})m)$.*

Proof. Let $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ be the m points sets drawn randomly from μ_P and for the set $\mathcal{K} = \{K_1, \ldots, K_m\}$ let K_j be the α-kernel of Q_j. Consider the probability distribution g describing the values of $f(Q)$ where Q is drawn randomly from μ_P. The set of m k-dimensional points $\{w_1 = f(Q_1), \ldots, w_m = f(Q_m)\}$ describes an ε-sample of $(g, \mathcal{R}_+)$ and hence also an ε-quantization of F_{μ_P}. We claim the set $\{w'_1 = f(K_1), \ldots, w'_m = f(K_m)\}$ forms an (ε, α)-quantization of F_{μ_P}.

For a query point $w \in \mathbb{R}^k$, let γm point sets from $\mathcal{Q}$ produce a value $w_j = f(Q_j)$ such that $w_j \preceq w$, and let $\gamma' m$ point sets from $\mathcal{K}$ produce a value $w'_j = f(K_j)$ such that $w'_j \preceq w$. Note that $\gamma' > \gamma$. Let $\hat{w} = w - \theta(\alpha)w$; more specifically, for each coordinate $w^{(i)}$ of w, $\hat{w}^{(i)} = w^{(i)} - \theta(\alpha)w^{(i)}$. Because f is relative $\theta(\alpha)$-approximable, for each point set $Q_j \in \mathcal{Q}$ such that $w_j \not\preceq w$, then $w'_j \not\preceq \hat{w}$. Thus, the number of point sets such that $f(K_j) \preceq \hat{w}$ is at most γm, and hence there is a point w' between w and $\hat{w}$ such that the fraction of sampled point sets such that $f(K_j) \preceq w'$ is exactly γ, and hence is within ε of the true fraction of point sets sampled from μ_P with probability at least $1 - \delta$. □

To name a few examples, the width and diameter are relative 2α-approximable functions, thus the results apply directly with $k = 1$. The radius of the minimum enclosing ball is relative 4α-approximable with $k = 1$. The d directional widths of the minimum perimeter or minimum volume axis-aligned rectangle is relative 2α-approximable with $k = d$.

Remark 2.1. *If an $(\varepsilon, \delta, \alpha)$-kernel is used for one query, it is correct with probability at least $1 - \delta$, and if it is used for another query, it is also correct with probability at least $1 - \delta$. Although there is probably some dependence between these two quantities, it is not easy to prove in general, hence we only claim the probability they are both correct is at least $(1 - \delta)^2$. We can increase this back to $1 - \delta$ for k queries by setting $m = O((1/\varepsilon^2)(k + \log(1/\delta)))$, but we need to specify k in advance. If we had a deterministic construction to create an $(\varepsilon, 0, \alpha)$-kernel this would not be a problem, and we could, say, guarantee an (ε, α)-quantization for width in all directions simultaneously. However, this appears to be a much more difficult problem.*

Other coresets. In a similar fashion, coresets of a point set distribution μ_P can be formed using other coresets for other problems on discrete point sets. For instance, sample $m = O((1/\varepsilon^2)\log(1/\delta))$ points sets $\{P_1, \ldots, P_m\}$ each from μ_P and then store α-samples $\{Q_1 \subseteq P_1, \ldots, Q_m \subseteq P_m\}$ of each. (If we use random sampling in the second set, then not all distributions μ_{p_i} need to be sampled for each P_j in the first round.) This results in an $(\varepsilon, \delta, \alpha)$-sample of μ_P, and can, for example, be used to construct (with probability $1 - \delta$) an (ε, α)-quantization for the fraction of points expected to fall in a query disk. Similar constructions can be done for other coresets, such as ε-nets [22], k-center [4, 21], or smallest enclosing ball [8].

2.3 Shape Inclusion Probabilities

We can also use a variation of Algorithm 2.1 to construct ε-shape inclusion probability functions. For a point set $Q \subset \mathbb{R}^d$, let the summarizing shape $S_Q = \mathcal{S}(Q)$ be from some geometric family $\mathcal{S}$ so $(\mathbb{R}^d, \mathcal{S})$ has bounded VC-dimension ν. We randomly sample m point sets $\mathcal{Q} = \{Q_1, \ldots, Q_m\}$ each from μ_P and then find the summarizing shape $S_{Q_j} = \mathcal{S}(Q_j)$ (e.g. minimum enclosing ball) of each Q_j. Let this set of shapes be $S^{\mathcal{Q}}$. If there are multiple shapes from $\mathcal{S}$ which are equally optimal (as can happen degenerately[3] with, for example, minimum width slabs), choose one of these shapes at random. For a set of shapes $S' \subseteq \mathcal{S}$, let $S'_p \subseteq S'$ be the subset of shapes that contain $p \in \mathbb{R}^d$. We store $S^{\mathcal{Q}}$ and evaluate a query point $p \in \mathbb{R}^d$ by counting what fraction of the shapes the point is contained in, specifically returning $|S^{\mathcal{Q}}_p|/|S^{\mathcal{Q}}|$ in $O(\nu|S^{\mathcal{Q}}|) = O(\nu m)$ time. In some cases, this evaluation can be sped up with point location data structures.

[3] In cases such as the smallest enclosing ball under the ℓ_1 distance, there may be multiple possible optimal shapes, non-degenerately. We can either choose one at random, or redefine the summarizing shape as the union of all such shapes.

Theorem 2.5. *Consider a family of summarizing shapes $\mathcal{S}$ where $(\mathbb{R}^d, \mathcal{S})$ has VC-dimension ν and where it takes $T_{\mathcal{S}}(n)$ time to determine the summarizing shape $\mathcal{S}(Q)$ for any point set $Q \subset \mathbb{R}^d$ of size n. For a distribution μ_P of a point set of size n, with probability at least $1-\delta$, we can construct an ε-sip function of size $O((\nu/\varepsilon^2)(2^{\nu+1} + \log(1/\delta)))$ and in time $O(T_{\mathcal{S}}(n)(1/\varepsilon^2)\log(1/\delta))$.*

Proof. If $(\mathbb{R}^d, \mathcal{S})$ has VC-dimension ν, then the dual range space $(\mathcal{S}, P^*)$ has VC-dimension $\nu' \leq 2^{\nu+1}$, where P^* is all subsets $\mathcal{S}_p \subseteq \mathcal{S}$, for any $p \in \mathbb{R}^d$, such that $\mathcal{S}_p = \{S \in \mathcal{S} \mid p \in S\}$. Using the above algorithm, sample $m = O((1/\varepsilon^2)(\nu' + \log(1/\delta)))$ point sets Q from μ_P and generate the m summarizing shapes S_Q. Each shape is a random sample from $\mathcal{S}$ according to μ_P, and thus $S^{\mathcal{Q}}$ is an ε-sample of $(\mathcal{S}, P^*)$.

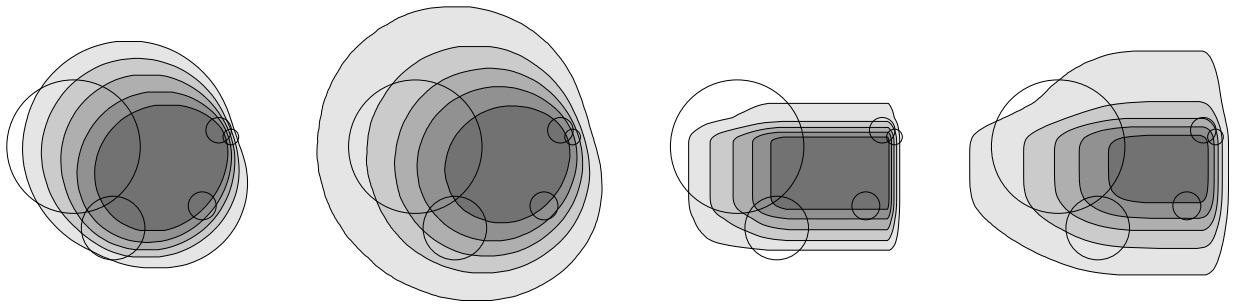
Let $w_{\mu_P}(S)$, for $S \in \mathcal{S}$, be the probability that S is the summarizing shape of a point set Q drawn randomly from μ_P. For any $\mathcal{S}' \subseteq P^*$, let $W_{\mu_P}(\mathcal{S}') = \int_{S \in \mathcal{S}'} w_{\mu_P}(S)$ be the probability that some shape from the subset $\mathcal{S}'$ is the summarizing shape of Q drawn from μ_P.

We approximate the sip function at $p \in \mathbb{R}^d$ by returning the fraction $|S^{\mathcal{Q}}_p|/m$. The true answer to the sip function at $p \in \mathbb{R}^d$ is $W_{\mu_P}(\mathcal{S}_p)$. Since $S^{\mathcal{Q}}$ is an ε-sample of $(\mathcal{S}, P^*)$, then with probability at least $1-\delta$

$$\left| \frac{|S^{\mathcal{Q}}_p|}{m} - \frac{W_{\mu_P}(\mathcal{S}_p)}{1} \right| = \left| \frac{|S^{\mathcal{Q}}_p|}{|S^{\mathcal{Q}}|} - \frac{W_{\mu_P}(\mathcal{S}_p)}{W_{\mu_P}(P^*)} \right| \leq \varepsilon.$$

Since for the family of summarizing shapes $\mathcal{S}$ the range space $(\mathbb{R}^d, \mathcal{S})$ has VC-dimension ν, each can be stored using that much space. □

Using deterministic techniques [11] the size can then be reduced to $O(2^{\nu+1}(\nu/\varepsilon^2) \cdot \log(1/\varepsilon))$ in time $O((2^{3(\nu+1)} \cdot (\nu/\varepsilon^2)\log(1/\varepsilon))^{2^{\nu+1}} \cdot 2^{3(\nu+1)}(\nu/\varepsilon^2)\log(1/\delta))$.



(a) (b) (c) (d)

Figure 4: The shape inclusion probability for the smallest enclosing ball (a,b) or smallest enclosing axis-aligned rectangle (c,d), for points uniformly distributed inside the circles (a,c) or normally distributed around circle centers with standard deviation given by radii (b,d).

Representing ε-sip functions by isolines. Shape inclusion probability functions are density functions. One convenient way of visually representing a density function in $\mathbb{R}^2$ is by drawing the isolines. A *γ-isoline* is a collection of closed curves bounding a region of the plane where the density function is greater than γ.

In each part of Figure 4 a set of 5 circles correspond to points with a probability distribution. In part (a,c) the probability distribution is uniform over the inside of the circles. In part (b,d) it is drawn from a multivariate Gaussian distribution, where the standard deviation is given by the radius or the circle. We generate ε-sip functions for the smallest enclosing ball in Figure 4(a,b) and for the smallest axis-aligned bounding box in Figure 4(c,d).

In all figures we draw approximations of $\{.9, .7, .5, .3, .1\}$-isolines. These drawing are generated by randomly selecting $m = 5000$ (Figure 4(a,b)) or $m = 25000$ (Figure 4(c,d)) shapes, counting the number of inclusions at different points in the plane and interpolating to get the isolines. The innermost and darkest region has probability $> 90\%$, the next one probability $> 70\%$, etc., the outermost region has probability $< 10\%$.

When μ_P describes the distribution for n points and n is large, then isolines are generally connected for convex summarizing shapes. In fact, in $O(n)$ time we can create a point which is contained in the convex hull of a point set sampled from μ_P with high probability. Specifics are discussed in Appendix A.

3 Measuring the Error

We have established asymptotic bounds of $O((1/\varepsilon^2)(\nu + \log(1/\delta))$ random samples for constructing ε-quantizations and ε-sip functions. In this section we empirically demonstrate that the constant hidden by the big-O notation is approximately 0.5, indicating that these algorithms are indeed quite practical. Additionally, we show that we can reduce the size of ε-quantizations to $2/\varepsilon$ without sacrificing accuracy and with only a factor 4 increase in the runtime. We also briefly compare the (ε, α)-quantizations produced with $(\varepsilon, \delta, \alpha)$-kernels to ε-quantizations. We show that the $(\varepsilon, \delta, \alpha)$-kernels become useful when the number of uncertain points becomes large, i.e. exceeding 1000.

Univariate ε-quantizatons. We consider a set of $n = 50$ sample points in $\mathbb{R}^3$ chosen randomly from the boundary of a cylinder piece of length 10 and radius 1. We let each point represent the center of 3-variate Gaussian distribution with standard deviation 2 to represent the probability distribution of an uncertain point. This set of distributions describes an uncertain point set $\mu_P : \mathbb{R}^{3n} \to \mathbb{R}^+$.

We want to estimate three statistics on μ_P: diam, the diameter of the point set; dwid, the width of the points set in a direction that makes an angle of 75° with the cylinder axis; and seb_2, the radius of the smallest enclosing ball (using code from Bernd Gärtner [17]). We can create ε-quantizations using our randomized algorithm with m samples from μ_P, where the value of m is from the set $\{16, 64, 256, 1024, 4096\}$.

We would like to evaluate the ε-quantizations versus the ground truth function F_{μ_P}; however, it is not clear how to evaluate F_{μ_P}. Instead, we create another ε-quantization Q with $\eta = 100000$ samples from μ_P, and treat this as if it were the ground truth. To evaluate each sample ε-quantization R versus Q we find the maximum deviation (i.e. $d_\infty(R, Q) = \max_{q \in \mathbb{R}} |h_R(q) - h_Q(q)|$) with h defined respect to diam, dwid, or seb_2. This can be done by for each value $r \in R$ evaluating $|h_R(r) - h_Q(r)|$ and $|(h_R(r) - 1/|R|) - h_Q(r)|$ and returning the maximum of both values over all $r \in R$. Since, for two consecutive points $q_i, q_{i+1} \in R$, the value of h_Q must increase monotonically between these values, so the maximum deviation must occur at the boundary of some such interval between consecutive points. This maximum error can be calculated in $O(\eta + m)$ time by scanning the two data structures in parallel and maintaining running sums (or in $O(m \log \eta)$ time using a binary tree on Q).

Given a fixed "ground truth" quantization Q we repeat this process for $\tau = 500$ trials of R, each returning a $d_\infty(R, Q)$ value. The set of these τ maximum deviations values results in another quantization S for each of diam, dwid, and seb_2. Intuitively, the maximum deviation quantization S describes the sample probability that $d_\infty(R, Q)$ will be less than some query value. These are plotted in Figure 5 for each value of m.

Note that the maximum deviation quantizations S are similar for all three statistics, and thus we can use these plots to estimate $1 - \delta$, the sample probability that $d_\infty(R, Q) \leq \varepsilon$, given a value m. We can fit this function as approximately $1 - \delta = 1 - \exp(-m\varepsilon^2/C + \nu)$ with $C = 0.5$ and $\nu = 1.0$. Thus solving for m in terms of ε, ν, and δ reveals: $m = C(1/\varepsilon^2)(\nu + \log(1/\delta))$. [4] This indicates that the big-O notation for

[4] Actually the function $1 - \delta = 1 - \exp(\varepsilon(\sqrt{m/C} - \nu)^2)$ and $m = C(1/\varepsilon^2)(\nu + \sqrt{\log(1/\delta)})^2$ with $C = 0.3$ and $\nu = 0.75$ fits the data much better but does not match the asymptotic bound as directly.

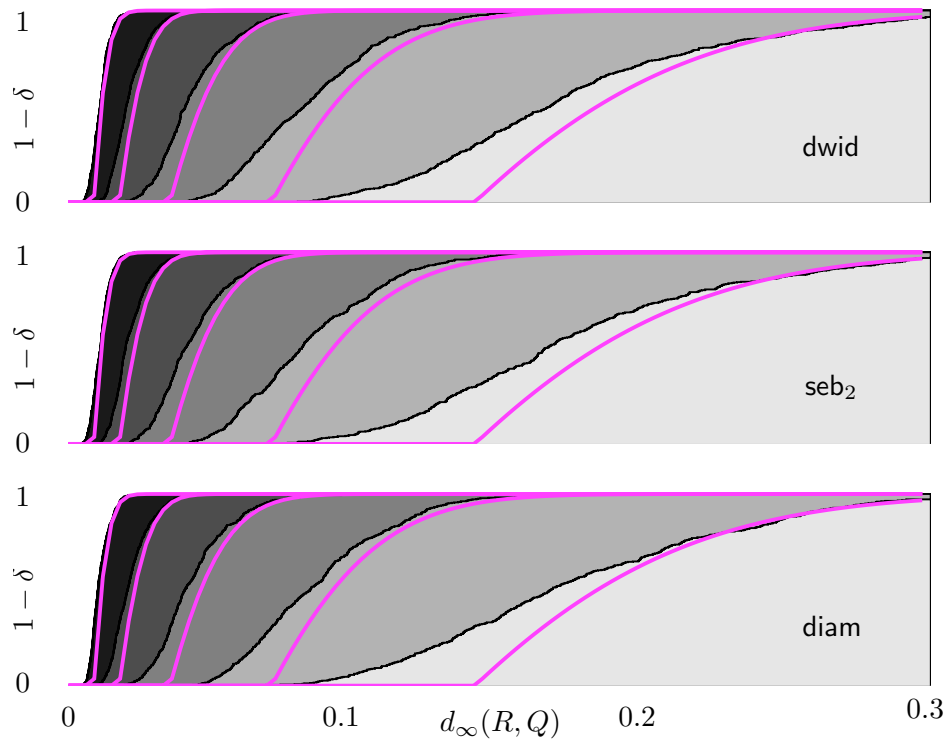


Figure 5: Shows quantizations of $\tau = 500$ trials of quantizations for $d_\infty(R, Q)$ where Q and R measure diam, dwid, and seb_2. The size of each R is $m = \{16, 64, 256, 1024, 4096\}$ (from right to left) and the "ground truth" quantization Q has size $\eta = 100000$. Smooth, thick curves are $1 - \delta = 1 - \exp(-2m\varepsilon^2 + 1)$ where $\varepsilon = d_\infty(R, Q)$.

the asymptotic bound of $O((1/\varepsilon^2)(\nu + \log(1/\delta))$ established in [26] for ε-samples only hides a constant of approximately 0.5.

Maximum error in sip functions. We can perform a similar analysis on sip functions. We use the same input data as is used to generate Figure 4(b) and create sip functions R for the smallest enclosing ball using $m = \{16, 36, 81, 182, 410\}$ samples from μ_P. We compare this to a "ground truth" sip function Q formed using $\eta = 5000$ sampled points. The maximum deviation between R and Q in this context is defined $d_\infty(R, Q) = \max_{q \in \mathbb{R}^2} |R(q) - Q(q)|$ and can be found by calculating $|R(q) - Q(q)|$ for all points $q \in \mathbb{R}^2$ at the intersection of boundaries of discs from R or Q.

We repeat this process for $\tau = 100$ trials, for each value of m. This creates a quantization S (for each value of m) measuring the maximum deviation for the sip functions. These maximum deviation quantizations are plotted in Figure 6. We fit these curves with a function $1 - \delta = 1 - \exp(-m\varepsilon^2/C + \nu)$ with $C = 0.5$ and $\nu = 2.0$, so $m = C(1/\varepsilon^2)(\nu + \log 1/\delta)$. Note that the dual range space $(\mathcal{B}, \mathbb{R}^{2*})$, defined by disks $\mathcal{B}$ has VC-dimension 2, so this is exactly what we would expect.

Maximum error in k-variate quantizations. We can extend these experiments to k-variate quantizations by considering the width in k different directions (we choose orthogonal directions, one along the cylinder axis). We use $k = \{1, 2, 3, 4\}$ directions for the same data set as above and construct a "ground truth" quantization Q with $\eta = 50000$ sampled point sets from μ_P. Then for $\tau = 500$ trials we construct

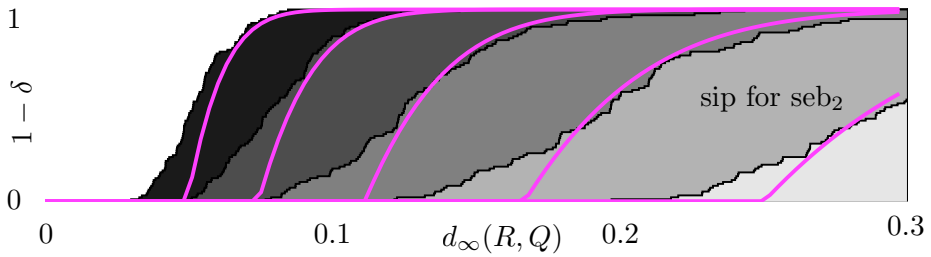


Figure 6: Shows ε-quantization of $\tau = 100$ trials of maximum deviation between sip functions for smallest enclosing disc with $m = \{16, 36, 81, 182, 410\}$ (from right to left) sample shapes versus a "ground truth" sip function with $\eta = 5000$ sample shapes.

quantizations R with $m = \{16, 36, 81, 182, 410\}$ samples from μ_P.

For each sample quantization R we compute the maximum error with respect to Q by comparing h_R at the critical point at intersection of regions $\{q \in \mathbb{R}^k | q \succeq r\}$ for up to k of the m samples $r \in R$ versus the value of h_Q. Using a similar argument as with univariate ε-quantizations, because h_R and h_Q are monotone, we can argue that the maximum difference must occur at one of these locations. The quantizations formed by the τ trials are shown in Figure 7.

The curves describing the quantizations are fit with a function $1 - \delta = 1 - \exp(-m\varepsilon^2/C + k)$ with $C = 0.5$, so $m = C(1/\varepsilon^2)(k + \log 1/\delta)$. This works okay for $k = \{1, 2\}$ but is too conservative for $k = \{3, 4\}$. In fact, the algorithm has better results for $k = \{3, 4\}$ than for $k = 2$, most noticeably when $m = 16$. We believe this is because the errors in different coordinates are not completely correlated, and we are seeing a regression to the mean.

3.1 Compressing ε-Quantizations

Theorem 2.1 describes how to compress the size of a univariate ε-quantization to $O(1/\varepsilon)$. We first create an $(\varepsilon/2)$-quantization of size m, then sort the values V_i, and finally take every $(m\varepsilon/2)$th value according to the sorted order. This returns an ε-quantization of size $2/\varepsilon$ and requires creating an initial ε-quantization with 4 times as many samples as we would have without this compression. The results, shown in Figure 8 using the same setup as in Figure 5, confirms that this compression scheme works better than the worst case claims. We only show the plot for diam, but the results for dwid and seb$_2$ are nearly identical. In particular, the error is smaller than the results in Figure 5, but it takes 4 times as long to compute.

3.2 $(\varepsilon, \delta, \alpha)$-Kernels versus ε-Quantizations

We also implemented the randomized algorithms for $(\varepsilon, \delta, \alpha)$-kernels to compare them with ε-quantizations for diam, dwid, and seb$_2$. We used existing code from Hai Yu [42] for α-kernels. For the input set μ_P we generated $n = 5000$ points $P \subset \mathbb{R}^3$ on the surface of a cylinder piece with radius 1 and axis length 10. Each point $p \in P$ represented the center of a Gaussian with standard deviation 3. We set $\varepsilon = 0.2$ and $\delta = 0.1$, resulting in $m = 40$ point sets sampled from μ_P. We also generated α-kernels of size at most 40 (the existing code did not allow the user to specify a parameter α, only the maximum size). The $(\varepsilon, \delta, \alpha)$-kernel has a total of 1338 points. We calculated ε-quantizations and (ε, α)-quantizations for diam, dwid, and seb$_2$, each compressed to a size 10 shown in Figure 9.

This method starts becoming useful in compressing μ_P when $n \gg 1000$, otherwise the total size of the $(\varepsilon, \delta, \alpha)$-kernel may be larger than μ_P. It may improve the efficiency for smaller values of n if the function

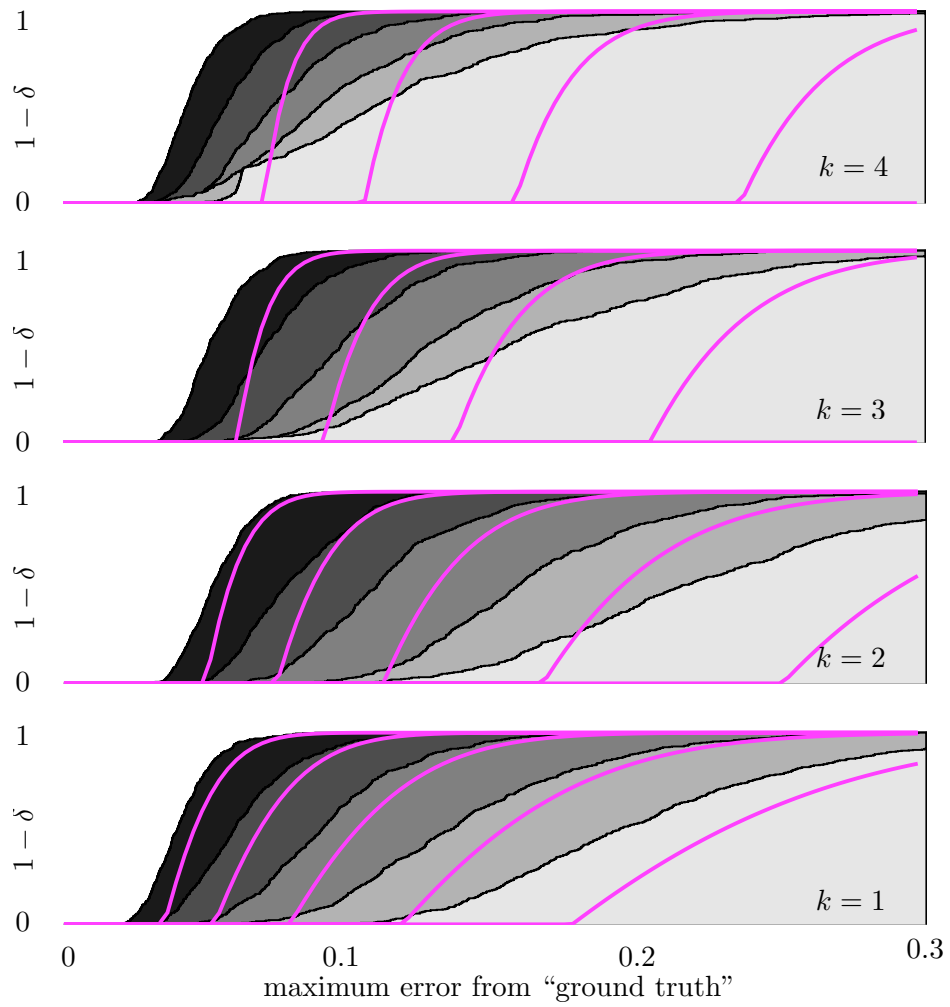


Figure 7: Shows quantization of $\tau = 500$ trials of k-variate quantizations defined by width in k-directions with $k = \{1, 2, 3, 4\}$ (from bottom to top) each of size $m = \{16, 36, 81, 182, 410\}$ (from right to left) versus a "ground truth" quantization with $\eta = 50000$.

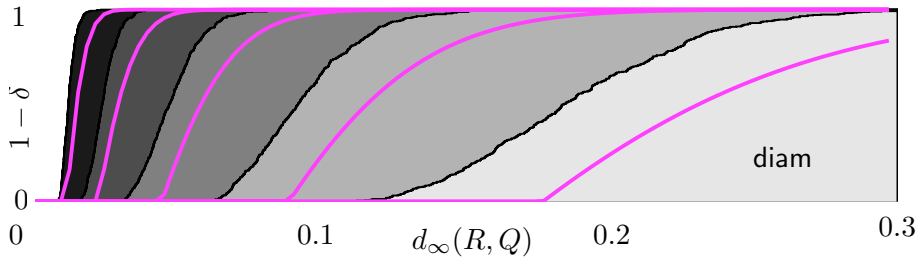


Figure 8: Shows quantization of $\tau = 500$ trials of quantizations for $d_\infty(R, Q)$ where Q and R measure diam. The size of each R is $m = \{64, 256, 1024, 4096, 16384\}$, then compressed to size $\{8, 16, 32, 64, 128\}$ (respectively, from right to left) and the "ground truth" quantization Q has size $\eta = 100000$.

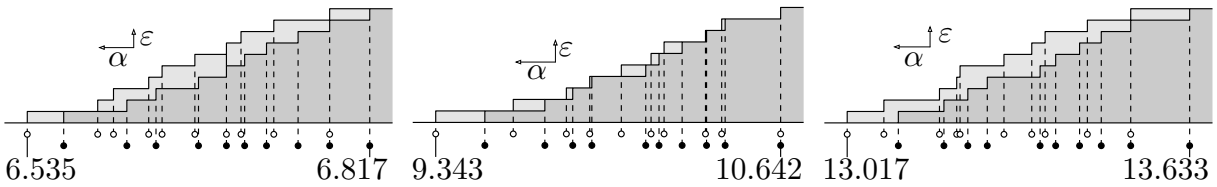


Figure 9: (ε, α)-quantization (white points) and ε-quantization (black points) for (a) seb_2, (b) dwid, and (c) diam.

$f : \mu_P \to \mathbb{R}$ that the quantization is approximating is expensive to compute (e.g. it takes $O(n^\rho)$ time for $\rho > 1$). We point the curious reader to [42] to validate the practicality of α-kernels.

Acknowledgements

We would like to thank Pankaj K. Agarwal for many helpful discussions. This research was partially supported by the Netherlands Organisation for Scientific Research (NWO) through the project GOGO.

References

[1] Charu C. Agarwal and Philip S. Yu, editors. *Privacy Preserving Data Mining: Models and Algorithms*. Springer, 2008.

[2] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi Varadarajan. Geometric approximations via coresets. *Current Trends in Combinatorial and Computational Geometry (E. Welzl, ed.)*, 2007.

[3] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measure of points. *Journal of ACM*, 51(4):2004, 2004.

[4] Pankaj K. Agarwal, Cecilia M. Procopiuc, and Kasturi R. Varadarajan. Approximation algorithms for k-line center. In *Proceedings 10th Annual European Symposium on Algorithms*, pages 54–63, 2002.

[5] Rakesh Agarwal and Ramakrishnan Srikant. Privacy-preserving data mining. *ACM SIGMOD Record*, 29:439–450, 2000.

[6] Hüseyin Akcan, Alex Astashyn, and Hervé Brönnimann. Deterministic algorithms for sampling count data. *Data & Knowledge Engineering*, 64(2):405–418, February 2008.

[7] Deepak Bandyopadhyay and Jack Snoeyink. Almost-Delaunay simplices: Nearest neighbor relations for imprecise points. In *ACM-SIAM Symp on Discrete Algorithms*, pages 403–412, 2004.

[8] Mihai Bădoiu and Ken Clarkson. Smaller core-sets for balls. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[9] Timothy Chan. Faster core-set constructions and data-stream algorithms in fixed dimensions. *Computational Geometry: Theory and Applications*, 35:20–35, 2006.

[10] Timothy Chan. Dynamic coresets. In *Proceedings of the 24th ACM Symposium on Computational Geometry*, pages 1–9, 2008.

[11] Bernard Chazelle and Jiri Matousek. On linear-time deterministic algorithms for optimization problems in fixed dimensions. *Journal of Algorithms*, 21:579–597, 1996.

[12] Reynold Cheng, Jichuan Chen, Mohamed Mokbel, and Chi-Yin Chow. Probability verifiers: Evaluating constrainted nearest-neighbor queries over uncertain data. In *Proceedings Interantional Conference on Data Engineering*, 2008.

[13] Reynold Cheng, Dmitri V. Kalashnikov, and Sunil Prabhakar. Evaluating probabilitic queries over imprecise data. In *Proceedings 2003 ACM SIGMOD International Conference on Management of Data*, 2003.

[14] Kenneth L. Clarkson, David Eppstein, Gary L. Miller, Carl Sturtivant, and Shang-Hua Teng. Approximating center points with iterative Radon points. *International Journal of Computational Geometry and Applications*, 6:357–377, 1996.

[15] Amol Deshpande, Carlos Guestrin, Samuel R. Madden, Joseph M. Hellerstein, and Wei Hong. Model-driven data acquisition in sensor networks. In *Proceedings 13th International Conference on Very Large Data Bases*, 2004.

[16] Austin Eliazar and Ronald Parr. Dp-slam 2.0. In *Proceedings 2004 IEEE International Conference on Robotics and Automation*, 2004.

[17] Bernd Gärtner. Fast and robust smallest enclosing balls. In *Proceedings 7th Annual European Symposium on Algorithms*, volume LNCS 1643, pages 325–338, 1999.

[18] Leonidas J. Guibas, D. Salesin, and J. Stolfi. Epsilon geometry: building robust algorithms from imprecise computations. In *Proc. 5th Annu. ACM Sympos. Comput. Geom.*, pages 208–217, 1989.

[19] Leonidas J. Guibas, D. Salesin, and J. Stolfi. Constructing strongly convex approximate hulls with inaccurate primitives. *Algorithmica*, 9:534–560, 1993.

[20] R. H. Güting and M. Schneider. *Moving Object Databases*. Morgan Kaufmann, San Francisco, 2005.

[21] Sariel Har-Peled. No coreset, no cry. In *Proceedings 24th Conference on Foundations of Software Technology and Theoretical Computer Science*, 2004.

[22] David Haussler and Emo Welzl. epsilon-nets and simplex range queries. *Discrete and Computational Geometry*, 2:127–151, 1987.

[23] Martin Held and Joseph S. B. Mitchell. Triangulating input-constrained planar point sets. *Information Processing Letters*, 109:54–56, 2008.

[24] Dmitri V. Kalashnikov, Yiming Ma, Sharad Mehrotra, and Ramaswamy Hariharan. Index for fast retreival of uncertain spatial point data. In *Proceedings 16th ACM SIGSPATIAL Interanational Conference on Advances in Geographic Information Systems*, 2008.

[25] Heinrich Kruger. Basic measures for imprecise point sets in $\mathbb{R}^d$. Master's thesis, Utrecht University, 2008.

[26] Yi Li, Philip M. Long, and Aravind Srinivasan. Improved bounds on the samples complexity of learning. *Journal of Computer ans System Science*, 62:516–527, 2001.

[27] T. M. Lillesand, R. W. Kiefer, and J. W. Chipman. *Remote Sensing and Image Interpretaion*. John Wiley & Sons, 2004.

[28] Maarten Löffler and Jack Snoeyink. Delaunay triangulations of imprecise points in linear time after preprocessing. In *Proc. 24th Sympoium on Computational Geometry*, pages 298–304, 2008.

[29] Jiri Matousek. Approximations and optimal geometric divide-and-conquer. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*, pages 505–511, 1991.

[30] Jiri Matousek. *Geometric Discrepancy; An Illustrated Guide*, volume 18 of *Algorithms and Combinatorics*. Springer, 1999.

[31] Jiri Matousek, Emo Welzl, and Lorenz Wernisch. Discrepancy and approximations for bounded vc-dimension. *Combinatorica*, 13(4):455–466, 1993.

[32] T. Nagai and N. Tokura. Tight error bounds of geometric problems on convex objects with imprecise coordinates. In *Jap. Conf. on Discrete and Comput. Geom.*, LNCS 2098, pages 252–263, 2000.

[33] Y. Ostrovsky-Berman and L. Joskowicz. Uncertainty envelopes. In *Abstracts 21st European Workshop on Comput. Geom.*, pages 175–178, 2005.

[34] Jeff M. Phillips. Algorithms for ε-approximations of terrains. In *Proceedings 35th International Colloquium on Automata, Languages, and Programming*, 2008. arXiV 0801.2793.

[35] Shobha Potluri, Anthony K. Yan, James J. Chou, Bruce R. Donald, and Chris Baily-Kellogg. Structure determination of symmetric homo-oligomers by complete search of symmetry configuration space, using nmr restraints and van der Waals packing. *Proteins*, 65:203–219, 2006.

[36] R. Rado. A theorem on general measure. *Journal of the London Mathematical Society*, 21:291–300, 1947.

[37] S. Shekhar and S. Chawla. *Spatial Databases: A Tour*. Pearsons, 2001.

[38] Yufei Tao, Reynold Cheng, Xiaokui Xiao, Wang Kay Ngai, Ben Kao, and Sunil Prabhakar. Indexing mutli-dimensional uncertain data with arbitrary probability density functions. In *Proceedings 31st Very Large Data Bases Conference*, 2005.

[39] Sebastian Thrun. Robotic mapping: A survey. *Exploring Artificial Intelligence in the New Millenium*, 2002.

[40] Marc van Kreveld and Maarten Löffler. Largest bounding box, smallest diameter, and related problems on imprecise points. *accepted for publication in Computational Geometry: Theory and Applications.*

[41] Vladimir Vapnik and Alexey Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16:264–280, 1971.

[42] Hai Yu, Pankaj K. Agarwal, Raghunath Poreddy, and Kasturi R. Varadarajan. Practical methods for shape fitting and kinetic data structures using coresets. In *Proceedings 20th Annual Symposium on Computational Geometry*, 2004.

A A center point for μ_P.

We can create a point $\bar{q} \in \mathbb{R}^d$ that is in the convex hull of a sampled point set Q from μ_P with high probability. This implies that for any summarizing shape that contains the convex hull, $\bar{q}$ is also contained in that summarizing shape. For a point set $P \subset \mathbb{R}^d$, a *β-center point* is a point $q \in \mathbb{R}^d$, such that any closed

halfspace that contains q also contains at least $1/\beta$ fraction points of all points in P. It is known that for any discrete point set a $(d+1)$-center point always exists [36]. Let $\mathcal{H}$ be the family of subsets defined by halfspaces. For a point set P of size n, a $(2d+2)$-center point can be created in $O(d^{5d+3}\log^d d)$ time [14] by first creating an $(1/(2d+2))$-sample of $(P,\mathcal{H})$, and then running a brute force algorithm. Because the first step is creating an ε-sample, this can be extended to Lebesgue-measureable sets such as probability distributions as well.

We use the following algorithm:

1. Create a $(2d+2)$-center point $\bar{p}_i$ for each μ_{p_i}. Let the set be $\bar{P}$.
2. Create $(2d+2)$-center point $\bar{q}$ of $\bar{P}$.

For d constant, the algorithm runs in $O(n)$ time because we can create $(2d+2)$-center points a total of $n+1$ times, and each takes $O(1)$ time.

Lemma A.1. *Given a distribution of a point set μ_P (such that each point distribution is polygonally approximable) of n points in $\mathbb{R}^d$, there is an $O(n)$ time algorithm to create a point $\bar{q}$ that will be in the convex hull of a point set drawn from μ_P with probability at least $1-(e^{1/(2d+2)^2})^n$.*

Proof. Because $\bar{p}_i$ is a $(2d+2)$-center point of μ_{p_i}, any halfspace that contains $\bar{p}_i$ on its boundary (and does not contain $\bar{q}$) has probability at least $1/(2d+2)$ of containing a point randomly drawn from μ_{p_i}. Also, because $\bar{q}$ is a $(2d+2)$-center point of $\bar{P}$, for any direction $u\in\mathbb{S}^{d-1}$ there are at least $n/(2d+2)$ points $\bar{p}_i$ from $\bar{P}$ for which $\langle q,u\rangle \leq \langle \bar{p}_i,u\rangle$. Thus, if a point q_i is drawn from μ_{p_i} such that $\langle q,u\rangle \leq \langle \bar{p}_i,u\rangle$ then the probability that $\langle \bar{q},u\rangle \leq \langle q_i,u\rangle$ is at least $1/(2d+2)$. Hence, the probability that there is a separating halfspace between $\bar{q}$ and the convex hull of Q (where the halfspace is orthogonal to some direction u) is at most

$$(1-1/(2d+2))^{n/(2d+2)} = ((1-1/(2d+2))^{1/(2d+2)})^n \leq (e^{1/(2d+2)^2})^n.$$

□

Theorem A.1. *For a set of $m<n$ point sets drawn i.i.d. from μ_P, it follows that $\bar{q}$ is in each of the m convex hulls for each point sets with high probability (specifically with probability $\geq 1-m(e^{1/(2d+2)^2})^n$).*

Proof. Let $\beta = e^{1/(2d+2)^2}$. For any one point set the probability that $\bar{q}$ is contained in the convex hull is at least $1-\beta^n$. By the union bound, the probability that it is contained in all m convex hulls is at least $(1-\beta^n)^m = 1-m\beta^n+\binom{m}{2}\beta^{2n}-\binom{m}{3}\beta^{3n}+\ldots$. Since $n>m$, the sum of all terms after the first two in the expansion increase the probability. □

We say a family of shapes $\mathcal{S}$ is *convex* if $S(P)\in\mathcal{S}$ contains the convex hull of P and $S(P)$ is always a convex set. When $\mathcal{S}$ is convex, then for any point q, the line segment $\overline{q\bar{q}}$ is completely contained in $S(P)$ if and only if $q\in S(P)$. Thus, given a set of m summarizing shapes, for every boundary of a summarizing shape $\overline{q\bar{q}}$ crosses, q is outside that summarizing shape. This implies the following corollary.

Corollary A.1. *Consider a distribution μ_P of point sets of size n, a convex family of shapes $\mathcal{S}$ inducing a* sip *function $s_{\mathcal{S}}$ on μ_P, and a positive integer $m<n$. For $\gamma\leq 1-1/m$ the subset of $\mathbb{R}^d$ inside of the γ-isoline of $s_{\mathcal{S}}$, exists, is connected, and is star-shaped with high probability, specifically with probability at least $1-m(e^{1/(2d+2)^2})^n$.*