Geometric Approximation via Coresets*

Pankaj K. Agarwal† Sariel Har-Peled‡ Kasturi R. Varadarajan§

February 22, 2005

Abstract

The paradigm of coresets has recently emerged as a powerful tool for efficiently approximating various extent measures of a point set P. Using this paradigm, one quickly computes a small subset Q of P, called a *coreset*, that approximates the original set P and and then solves the problem on Q using a relatively inefficient algorithm. The solution for Q is then translated to an approximate solution to the original point set P. This paper describes the ways in which this paradigm has been successfully applied to various optimization and extent measure problems.

1 Introduction

One of the classical techniques in developing approximation algorithms is the extraction of "small" amount of "most relevant" information from the given data, and performing the computation on this extracted data. Examples of the use of this technique in a geometric context include random sampling [Cha01, Mul94], convex approximation [Dud74, BI76], surface simplification [HG97], feature extraction and shape descriptors [DM98, dFM01]. For geometric problems where the input is a set of points, the question reduces to finding a small subset (i.e., *coreset*) of the points, such that one can perform the desired computation on the coreset.

As a concrete example, consider the problem of computing the diameter of a point set. Here it is clear that, in the worst case, classical sampling techniques like ε-approximation and ε-net would fail to compute a subset of points that contain a good approximation to the diameter [VC71, HW87]. While in this problem it is clear that convex approximation (i.e., an approximation of the convex hull of the point set) is helpful and provides us with the desired coreset, convex approximation of the point set is not useful for computing the narrowest annulus containing a point set in the plane.

In this paper, we describe several recent results which employ the idea of coresets to develop efficient approximation algorithms for various geometric problems. In particular, motivated by a variety applications, considerable work has been done on measuring various descriptors of the extent of a set P of n points in $\mathbb{R}^d$. We refer to such measures as *extent measures* of P. Roughly

*Research by the first author is supported by NSF under grants CCR-00-86013, EIA-98-70724, EIA-01-31905, and CCR-02-04118, and by a grant from the U.S.–Israel Binational Science Foundation. Research by the second author is supported by NSF CAREER award CCR-0132901. Research by the third author is supported by NSF CAREER award CCR-0237431

†Department of Computer Science, Box 90129, Duke University, Durham NC 27708-0129; pankaj@cs.duke.edu; http://www.cs.duke.edu/~pankaj/

‡Department of Computer Science, DCL 2111; University of Illinois; 1304 West Springfield Ave., Urbana, IL 61801; sariel@uiuc.edu; http://www.uiuc.edu/~sariel/

§Department of Computer Science, The University of Iowa, Iowa City, IA 52242-1419; kvaradar@cs.uiowa.edu; http://www.cs.uiowa.edu/~kvaradar/

speaking, an extent measure of P either computes certain statistics of P itself or of a (possibly nonconvex) geometric shape (e.g. sphere, box, cylinder, etc.) enclosing P. Examples of the former include computing the kth largest distance between pairs of points in P, and the examples of the latter include computing the smallest radius of a sphere (or cylinder), the minimum volume (or surface area) of a box, and the smallest width of a slab (or a spherical or cylindrical shell) that contain P. There has also been some recent work on maintaining extent measures of a set of moving points [AGHV01].

Shape fitting, a fundamental problem in computational geometry, computer vision, machine learning, data mining, and many other areas, is closely related to computing extent measures. A widely used shape-fitting problem asks for finding a shape that best fits P under some "fitting" criterion. A typical criterion for measuring how well a shape γ fits P, denoted as $\mu(P, \gamma)$, is the maximum distance between a point of P and its nearest point on γ, i.e., $\mu(P, \gamma) = \max_{p \in P} \min_{q \in \gamma} \|p - q\|$. Then one can define the extent measure of P to be $\mu(P) = \min_\gamma \mu(P, \gamma)$, where the minimum is taken over a family of shapes (such as points, lines, hyperplanes, spheres, etc.). For example, the problem of finding the minimum radius sphere (resp. cylinder) enclosing P is the same as finding the point (resp. line) that fits P best, and the problem of finding the smallest width slab (resp. spherical shell, cylindrical shell)[1] is the same as finding the hyperplane (resp. sphere, cylinder) that fits P best.

The exact algorithms for computing extent measures are generally expensive, e.g., the best known algorithms for computing the smallest volume bounding box containing P in $\mathbb{R}^3$ run in $O(n^3)$ time. Consequently, attention has shifted to developing approximation algorithms [BH01]. The goal is to compute an ε-approximation, for some $0 < \varepsilon < 1$, of the extent measure in roughly $O(nf(\varepsilon))$ or even $O(n + f(\varepsilon))$ time, that is, in time near-linear or linear in n. The framework of coresets has recently emerged as a general approach to achieve this goal. For any extent measure μ and an input point set P for which we wish to compute the extent measure, the general idea is to argue that there exists an easily computable subset $Q \subseteq P$, called a *coreset*, of size $1/\varepsilon^{O(1)}$, so that solving the underlying problem on Q gives an approximate solution to the original problem. For example, if $\mu(Q) \geq (1 - \varepsilon)\mu(P)$, then this approach gives an approximation to the extent measure of P. In the context of shape fitting, an appropriate property for Q is that for any shape γ from the underlying family, $\mu(Q, \gamma) \geq (1 - \varepsilon)\mu(P, \gamma)$. With this property, the approach returns a shape γ^* that is an approximate best fit to P.

Following earlier work [BH01, Cha02, ZS02] that hinted at the generality of this approach, Agarwal *et al.* [AHV04] provided a formal framework by introducing the notion of ε-kernel and showing that it yields a coreset for many optimization problems. They also showed that this technique yields approximation algorithms for a wide range of problems. Since the appearance of preliminary versions of their work, many subsequent papers have used a coreset based approach for other geometric optimization problems, including clustering and other extent-measure problems [APV02, BC03b, BHI02, HW04, KMY03, KY04].

In this paper, we have attempted to review coreset based algorithms for approximating extent measure and other optimization problems. Our aim is to communicate the flavor of the techniques involved and a sense of the power of this paradigm by discussing a number of its applications. We begin in Section 2 by describing ε-kernels of point sets and algorithms for constructing them. Section 3 defines the notion of ε-kernel for functions and describes a few of its applications. We then describe in Section 4 a simple incremental algorithm for shape fitting. Section 5 discusses the computation of of ε-kernels in the streaming model. Although ε-kernels provide coresets for a

[1] A *slab* is a region lying between two parallel hyperplanes; a *spherical shell* is the region lying between two concentric spheres; a *cylindrical shell* is the region lying between two coaxial cylinders.

variety of extent measures, they do not give coresets for many other problems, including clustering. Section 6 surveys the known results on coresets for clustering. The size of the coresets discussed in these sections increases exponentially with the dimension, so we conclude in Section 7 by discussing coresets for points in very high dimensions whose size depends polynomially on the dimension, or is independent of the dimension altogether.

2 Kernels for Point Sets

Let μ be a measure function (e.g., the width of a point set) from subsets of $\mathbb{R}^d$ to the non-negative reals $\mathbb{R}^+ \cup \{0\}$ that is monotone, i.e., for $P_1 \subseteq P_2$, $\mu(P_1) \leq \mu(P_2)$. Given a parameter $\varepsilon > 0$, we call a subset $Q \subseteq P$ an *ε-coreset* of P (with respect to μ) if

$$(1 - \varepsilon)\mu(P) \leq \mu(Q).$$

Agarwal *et al.* [AHV04] introduced the notion of *ε-kernels* and showed that it is an $f(\varepsilon)$-coreset for numerous minimization problems. We begin by defining ε-kernels and related concepts.

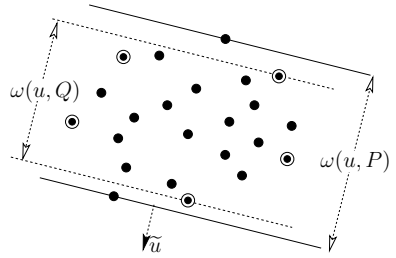


Figure 1. Directional width and ε-kernel.

ε-kernel. Let $\mathbb{S}^{d-1}$ denote the unit sphere centered at the origin in $\mathbb{R}^d$. For any set P of points in $\mathbb{R}^d$ and any direction $u \in \mathbb{S}^{d-1}$, we define the *directional width* of P in direction u, denoted by $\omega(u, P)$, to be

$$\omega(u, P) = \max_{p \in P} \langle u, p \rangle - \min_{p \in P} \langle u, p \rangle ,$$

where $\langle \cdot, \cdot \rangle$ is the standard inner product. Let $\varepsilon > 0$ be a parameter. A subset $Q \subseteq P$ is called an *ε-kernel* of P if for each $u \in \mathbb{S}^{d-1}$,

$$(1 - \varepsilon)\omega(u, P) \leq \omega(u, Q).$$

Clearly, $\omega(u, Q) \leq \omega(u, P)$. Agarwal *et al.* [AHV04] call a measure function μ *faithful* if there exists a constant c, depending on μ, so that for any $P \subseteq \mathbb{R}^d$ and for any ε, an ε-kernel of P is a $c\varepsilon$-coreset for P with respect to μ. Examples of faithful measures include diameter, width, radius of the smallest enclosing ball, and volume of the smallest enclosing box [AHV04]. A common property of these measures is that $\mu(P) = \mu(\text{conv}(P))$. We can thus compute an ε-coreset of P with respect to several measures by simply computing an (ε/c)-kernel of P.

Algorithms for computing kernels. An ε-kernel of P is a subset whose convex hull approximates, in a certain sense, the convex hull of P. Other notions of convex hull approximation have been studied and methods have been developed to compute them, see [BFP82, BI76, Dud74] for a sample. For example, Bentley, Faust, and Preparata [BFP82] show that for any point set $P \subseteq \mathbb{R}^2$

and $\varepsilon > 0$, a subset Q of P whose size is $O(1/\varepsilon)$ can be computed in $O(|P|+1/\varepsilon)$ time such that for any $p \in P$, the distance of p to $\mathrm{conv}(Q)$ is at most $\varepsilon\mathrm{diam}(Q)$. Note however that such a guarantee is not enough if we want Q to be a coreset of P with respect to faithful measures. For instance, the width of Q could be arbitrarily small compared to the width of P. The width of an ε-kernel of P, on the other hand, is easily seen to be a good approximation to the width of P. To the best of our knowledge, the first efficient method for computing a small ε-kernel of an arbitrary point set is implicit in the work of Barequet and Har-Peled [BH01].

We call P *α-fat*, for $\alpha \leq 1$, if there exists a point $p \in \mathbb{R}^d$ and a hypercube $\overline{\mathbb{C}}$ centered at the origin so that

$$p + \alpha\overline{\mathbb{C}} \subset \mathrm{conv}(P) \subset p + \overline{\mathbb{C}}.$$

A stronger version of the following lemma, which is very useful for constructing an ε-kernel, was proved in [AHV04] by adapting a scheme of [BH01]. Their scheme can be thought of as one that quickly computes an approximation to the Löwner-John Ellipsoid [Joh48].

Lemma 2.1 *Let P be a set of n points in $\mathbb{R}^d$ such that the volume of $\mathrm{conv}(P)$ is non-zero, and let $\mathbb{C} = [-1,1]^d$. One can compute in $O(n)$ time an affine transform τ so that $\tau(P)$ is an α-fat point set satisfying $\alpha\mathbb{C} \subset \mathrm{conv}(\tau(P)) \subset \mathbb{C}$, where α is a positive constant depending on d, and so that a subset $Q \subseteq P$ is an ε-kernel of P if and only if $\tau(Q)$ is an ε-kernel of $\tau(P)$.*

The importance of Lemma 2.1 is that it allows us to adapt some classical approaches for convex hull approximation [BFP82, BI76, Dud74] which in fact do compute an ε-kernel when applied to fat point sets.

We now describe algorithms for computing ε-kernels. By Lemma 2.1, we can assume that $P \subseteq [-1,+1]^d$ that is α-fat. We begin with a very simple algorithm.

Let δ be the largest value such that $\delta \leq (\varepsilon/\sqrt{d})\alpha$ and $1/\delta$ is an integer. We consider the d-dimensional grid $\mathbb{Z}$ of size δ. That is, $\mathbb{Z} = \{(\delta i_1, \ldots, \delta i_d) \mid i_1, \ldots, i_d \in \mathbb{Z}\}$. For each column along the x_d-axis in $\mathbb{Z}$, we choose one point from the highest nonempty cell of the column and one point from the lowest cell of the column; see Figure 2 (i). Let Q be the set of chosen points. Since $P \subseteq [-1,+1]^d$, $|Q| = O(1/(\alpha\varepsilon)^{d-1})$. Moreover Q can be constructed in time $O(n + 1/(\alpha\varepsilon)^{d-1})$ provided that the ceiling operation can be performed in constant time. Agarwal *et al.* [AHV04] showed that Q is an ε-kernel of P. Hence, we can compute an ε-kernel of P of size $O(1/\varepsilon^{d-1})$ in time $O(n + 1/\varepsilon^{d-1})$. This approach resembles the algorithm of Bentley, Faust, and Preparata [BFP82].

Next we describe an improved construction, observed independently by Chan [Cha04] and Yu *et al.* [YAPV04], which is a simplification of an algorithm by Agarwal *et al.* [AHV04], which in turn is an adaptation of a method of Dudley [Dud74]. Let $\mathcal{S}$ be the sphere of radius $\sqrt{d}+1$ centered at the origin. Set $\delta = \sqrt{\varepsilon\alpha} \leq 1/2$. One can construct a set $\mathcal{I}$ of $O(1/\delta^{d-1}) = O(1/\varepsilon^{(d-1)/2})$ points on the sphere $\mathcal{S}$ so that for any point x on $\mathcal{S}$, there exists a point $y \in \mathcal{I}$ such that $\|x - y\| \leq \delta$. We process P into a data structure that can answer ε-approximate nearest-neighbor queries [AMN+98]. For a query point q, let $\varphi(q)$ be the point of P returned by this data structure. For each point $y \in \mathcal{I}$, we compute $\varphi(y)$ using this data structure. We return the set $Q = \{\varphi(y) \mid y \in \mathcal{I}\}$; see Figure 2 (ii).

We now briefly sketch, following the argument in [YAPV04], why Q is is an ε-kernel of P. For simplicity, we prove the claim under the assumption that $\varphi(y)$ is the *exact* nearest-neighbor of y in P. Fix a direction $u \in \mathbb{S}^{d-1}$. Let $\sigma \in P$ be the point that maximizes $\langle u, p\rangle$ over all $p \in P$. Suppose the ray emanating from σ in direction u hits $\mathcal{S}$ at a point x. We know that there exists a point $y \in \mathcal{I}$ such that $\|x - y\| \leq \delta$. If $\varphi(y) = \sigma$, then $\sigma \in Q$ and

$$\max_{p\in P}\langle u, p\rangle - \max_{q\in Q}\langle u, q\rangle = 0.$$

Now suppose $\varphi(y) \neq \sigma$. Let B be the d-dimensional ball of radius $||y - \sigma||$ centered at y. Since $\|y - \varphi(y)\| \leq \|y - \sigma\|$, $\varphi(y) \in B$. Let us denote by z the point on the sphere ∂B that is hit by the ray emanating from y in direction $-u$. Let w be the point on zy such that $zy \perp \sigma w$ and h the point on σx such that $yh \perp \sigma x$; see Figure 2 (iii).

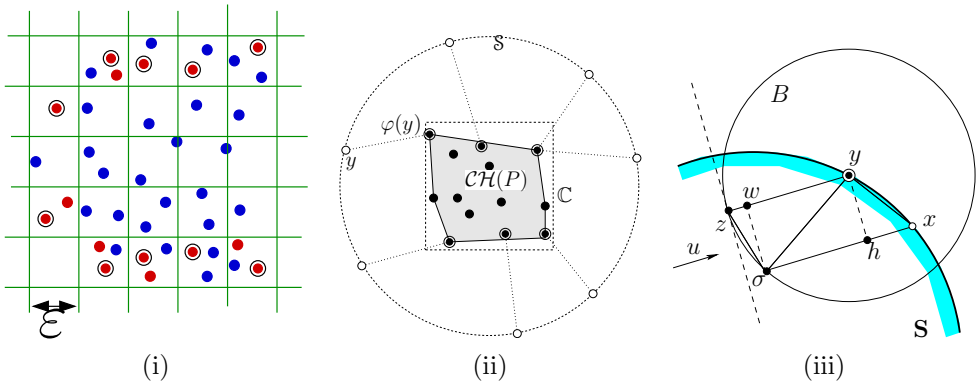


Figure 2. (i) A grid based algorithm for constructing an ε-kernel. (ii) An improved algorithm. (iii) Correctness of the improved algorithm.

The hyperplane normal to u and passing through z is tangent to B. Since $\varphi(y)$ lies inside B, $\langle u, \varphi(y)\rangle \geq \langle u, z\rangle$. Moreover, it can be shown that $\langle u, \sigma\rangle - \langle u, \varphi(y)\rangle \leq \alpha\varepsilon$. Thus, we can write

$$\max_{p \in P} \langle u, p\rangle - \max_{q \in Q} \langle u, q\rangle \leq \langle u, \sigma\rangle - \langle u, \varphi(y)\rangle \leq \alpha\varepsilon.$$

Similarly, we have $\min_{p \in P} \langle u, p\rangle - \min_{q \in Q} \langle u, q\rangle \geq -\alpha\varepsilon$.

The above two inequalities together imply that $\omega(u, Q) \geq \omega(u, P) - 2\alpha\varepsilon$. Since $\alpha\mathbb{C} \subset \text{conv}(P)$, $\omega(u, P) \geq 2\alpha$. Hence $\omega(u, Q) \geq (1 - \varepsilon)\omega(u, P)$, for any $u \in \mathbb{S}^{d-1}$, thereby implying that Q is an ε-kernel of P.

A straightforward implementation of the above algorithm, i.e., the one that answers a nearest-neighbor query by comparing the distances to all the points, runs in $O(n/\varepsilon^{(d-1)/2})$ time. However, we can first compute an $(\varepsilon/2)$-kernel Q' of P of size $O(1/\varepsilon^{d-1})$ using the simple algorithm and then compute an $(\varepsilon/4)$-kernel using the improved algorithm. Chan [Cha04] introduced the notion of discrete Voronoi diagrams, which can be used for computing the nearest neighbors of a set of grid points among the sites that are also a subset of a grid. Using this structure Chan showed that $\varphi(y)$, for all $y \in \mathcal{I}$, can be computed in a total time of $O(n + 1/\varepsilon^{d-1})$ time. Putting everything together, one obtains an algorithm that runs in $O(n + 1/\varepsilon^{d-1})$ time. Chan in fact gives a slightly improved result:

Theorem 2.2 ([Cha04]) *Given a set P of n points in $\mathbb{R}^d$ and a parameter $\varepsilon > 0$, one can compute an ε-kernel of P of size $O(1/\varepsilon^{(d-1)/2})$ in time $O(n + 1/\varepsilon^{d-(3/2)})$.*

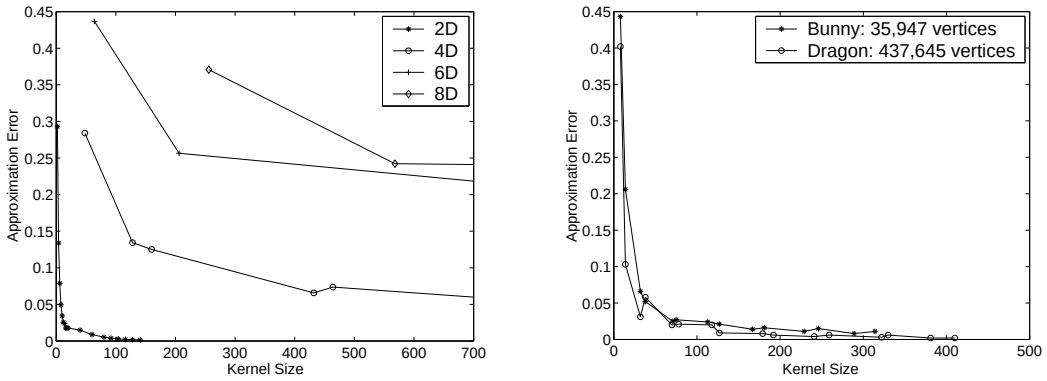
Experimental results. Yu *et al.* [YAPV04] implemented their ε-kernel algorithm and tested its performance on a variety of inputs. They measure the quality of an ε-kernel Q of P as the maximum relative error in the directional width of P and Q. Since it is hard to compute the maximum error over all directions, they sampled a set Δ of 1000 directions in $\mathbb{S}^{d-1}$ and computed the maximum relative error with respect to these directions, i.e.,

$$\text{err}(Q, P) = \max_{u \in \Delta} \frac{\omega(u, P) - \omega(u, Q)}{\omega(u, P)}. \tag{1}$$

Input Type	*Input Size*	$d = 2$		$d = 4$		$d = 6$		$d = 8$	
		Prepr	*Query*	*Prepr*	*Query*	*Prepr*	*Query*	*Prepr*	*Query*
sphere	10,000	0.03	0.01	0.06	0.05	0.10	9.40	0.15	52.80
	100,000	0.54	0.01	0.90	0.50	1.38	67.22	1.97	1393.88
	1,000,000	9.25	0.01	13.08	1.35	19.26	227.20	26.77	5944.89
cylinder	10,000	0.03	0.01	0.06	0.03	0.10	2.46	0.16	17.29
	100,000	0.60	0.01	0.91	0.34	1.39	30.03	1.94	1383.27
	1,000,000	9.93	0.01	13.09	0.31	18.94	87.29	26.12	5221.13
clustered	10,000	0.03	0.01	0.06	0.01	0.10	0.08	0.15	2.99
	100,000	0.31	0.01	0.63	0.02	1.07	1.34	1.64	18.39
	1,000,000	5.41	0.01	8.76	0.02	14.75	1.08	22.51	54.12

Table 1. Running time for computing ε-kernels of various synthetic data sets, $\varepsilon < 0.05$. *Prepr* denotes the preprocessing time, including converting P into a fat set and building ANN data structures. *Query* denotes the time for performing approximate nearest-neighbor queries. Running time is measured in seconds. The experiments were conducted on a Dell PowerEdge 650 server with a 3.06GHz Pentium IV processor and 3GB memory, running Linux 2.4.20.

They implemented the constant-factor approximation algorithm by Barequet and Har-Peled [BH01] for computing the minimum-volume bounding box to convert P into an α-fat set, and they used the ANN library [AM98] for answering approximate nearest-neighbor queries. Table 1 shows the running time of their algorithm for a variety of synthetic inputs: (i) points uniformly distributed on a sphere, (ii) points distributed on a cylinder, and (iii) clustered point sets, consisting of 20 equal sized clusters. The running time is decomposed into two components: (i) preprocessing time that includes the time spent in converting P into a fat set and in preprocessing P for approximate nearest-neighbor queries, and (ii) query time that includes the time spent in computing $\varphi(x)$ for $x \in \mathbb{J}$. Figure 3 shows how the error $\mathrm{err}(Q, P)$ changes as the function of kernel. These experiments show that their algorithm works extremely well in low dimensions (≤ 4) both in terms of size and running time. See [YAPV04] for more detailed experiments.



(a) (b)

Figure 3. Approximation errors under different sizes of computed ε-kernels. (a) *sphere*, (b) various geometric models. All synthetic inputs had 100,000 points.

Applications. Theorem 2.2 can be used to compute coresets for faithful measures, defined in Section 2. In particular, if we have a faithful measure μ that can be computed in $O(n^\alpha)$ time, then by Theorem 2.2, we can compute a value $\overline{\mu}$, $(1-\varepsilon)\mu(P) \leq \overline{\mu} \leq \mu(P)$ by first computing an (ε/c)-kernel Q of P and then using an exact algorithm for computing $\mu(Q)$. The total running time of the algorithm is $O(n + 1/\varepsilon^{d-(3/2)} + 1/\varepsilon^{\alpha(d-1)/2})$. For example, a $(1+\varepsilon)$-approximation of

the diameter of a point set can be computed in time $O(n + 1/\varepsilon^{d-1})$ since the exact diameter can be computed in quadratic time. By being a little more careful, the running time of the diameter algorithm can be improved to $O(n+1/\varepsilon^{d-(3/2)})$ [Cha04]. Table 2 gives running times for computing an $(1 + \varepsilon)$-approximation of a few faithful measures.

Extent	Time complexity
Diameter	$n + 1/\varepsilon^{d-(3/2)}$
Width	$(n + 1/\varepsilon^{d-2}) \log(1/\varepsilon)$
Minimum enclosing cylinder	$n + 1/\varepsilon^{d-1}$
Minimum enclosing box(3D)	$n + 1/\varepsilon^3$

Table 2. Time complexity of computing $(1 + \varepsilon)$-approximations for certain faithful measures.

We note that ε-kernels in fact guarantee a stronger property for several faithful measures. For instance, if Q is an ε-kernel of P, and C is some cylinder containing Q, then a "concentric" scaling of C by a factor of $(1 + c\varepsilon)$, for some constant c, contains P. Thus we can compute not only an approximation to the minimum radius r^* of a cylinder containing P, but also a cylinder of radius at most $(1 + \varepsilon)r^*$ that contains P.

The approach described in this section for approximating faithful measures had been used for geometric approximation algorithms before the framework of ε-kernels was introduced; see e.g. [AP02, BH01, Cha02, ZS02]. The framework of ε-kernels, however, provides a unified approach and turns out to be crucial for the approach developed in the next section for approximating measures that are not faithful.

3 Kernels for Sets of Functions

The crucial notion used to derive coresets and efficient approximation algorithms for measures that are not faithful is that of a kernel of a set of functions.

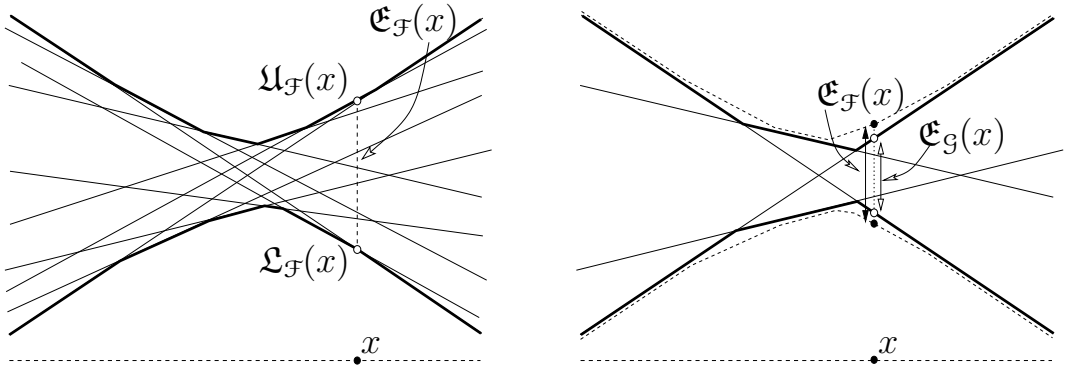


Figure 4. Envelopes, extent, and ε-kernel.

Envelopes and extent. Let $\mathcal{F} = \{f_1, \ldots, f_n\}$ be a set of n d-variate real-valued functions defined over $x = (x_1, \ldots, x_{d-1}, x_d) \in \mathbb{R}^d$. The *lower envelope* of $\mathcal{F}$ is the graph of the function $\mathfrak{L}_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}$ defined as $\mathfrak{L}_{\mathcal{F}}(x) = \min_{f \in \mathcal{F}} f(x)$. Similarly, the *upper envelope* of $\mathcal{F}$ is the graph of the function

$\mathfrak{U}_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}$ defined as $\mathfrak{U}_{\mathcal{F}}(x) = \max_{f \in \mathcal{F}} f(x)$. The *extent* $\mathfrak{E}_{\mathcal{F}} : \mathbb{R}^d \to \mathbb{R}$ of $\mathcal{F}$ is defined as

$$\mathfrak{E}_{\mathcal{F}}(x) = \mathfrak{U}_{\mathcal{F}}(x) - \mathfrak{L}_{\mathcal{F}}(x).$$

Let $\varepsilon > 0$ be a parameter. We say that a subset $\mathcal{G} \subseteq \mathcal{F}$ is an *ε-kernel* of $\mathcal{F}$ if

$$(1 - \varepsilon)\mathfrak{E}_{\mathcal{F}}(x) \leq \mathfrak{E}_{\mathcal{G}}(x) \qquad \forall x \in \mathbb{R}^d.$$

Obviously, $\mathfrak{E}_{\mathcal{G}}(x) \leq \mathfrak{E}_{\mathcal{F}}(x)$, as $\mathcal{G} \subseteq \mathcal{F}$.

Let $\mathcal{H} = \{h_1, \ldots, h_n\}$ be a family of d-variate linear functions and $\varepsilon > 0$ a parameter. We define a *duality* transformation that maps the d-variate function (or a hyperplane in $\mathbb{R}^{d+1}$) $h : x_{d+1} = a_1 x_1 + a_2 x_2 + \cdots + a_d x_d + a_{d+1}$ to the point $h^\star = (a_1, a_2, \ldots, a_d, a_{d+1})$ in $\mathbb{R}^{d+1}$. Let $\mathcal{H}^\star = \{h^\star \mid h \in \mathcal{H}\}$. It can be proved [AHV04] that $\mathcal{K} \subseteq \mathcal{H}$ is an ε-kernel of $\mathcal{H}$ if and only if $\mathcal{K}^*$ is an ε-kernel of $\mathcal{H}^*$. Hence, by computing an ε-kernel of $\mathcal{H}^*$ we can also compute an ε-kernel of $\mathcal{H}$. The following is therefore a corollary of Theorem 2.2.

Corollary 3.1 ([AHV04, Cha04]) *Given a set $\mathcal{F}$ of n d-variate linear functions and a parameter $\varepsilon > 0$, one can compute an ε-kernel of $\mathcal{F}$ of size $O(1/\varepsilon^{d/2})$ in time $O(n + 1/\varepsilon^{d-(1/2)})$.*

We can compute ε-kernels of a set of polynomial functions by using the notion of linearization.

Linearization. Let $f(x, a)$ be a $(d+p)$-variate polynomial, $x \in \mathbb{R}^d$ and $a \in \mathbb{R}^p$. Let $a^1, \ldots, a^n \in \mathbb{R}^p$, and set $\mathcal{F} = \{f_i(x) \equiv f(x, a^i) \mid 1 \leq i \leq n\}$. Suppose we can express $f(x, a)$ in the form

$$f(x, a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \cdots + \psi_k(a)\varphi_k(x), \tag{2}$$

where $\psi_0, \ldots, \psi_k$ are p-variate polynomials and $\varphi_1, \ldots, \varphi_k$ are d-variate polynomials. We define the map $\varphi : \mathbb{R}^d \to \mathbb{R}^k$

$$\varphi(x) = (\varphi_1(x), \ldots, \varphi_k(x)).$$

Then the image $\Gamma = \{\varphi(x) \mid x \in \mathbb{R}^d\}$ of $\mathbb{R}^d$ is a d-dimensional surface in $\mathbb{R}^k$ (if $k \geq d$), and for any $a \in \mathbb{R}^p$, $f(x, a)$ maps to a k-variate linear function

$$h_a(y_1, \ldots, y_k) = \psi_0(a) + \psi_1(a)y_1 + \cdots + \psi_k(a)y_k$$

in the sense that for any $x \in \mathbb{R}^d$, $f(x, a) = h_a(\varphi(x))$. We refer to k as the *dimension* of the *linearization* φ, and say that $\mathcal{F}$ admits a linearization of dimension k. The most popular example of linearization is perhaps the so-called lifting transform that maps $\mathbb{R}^d$ to a unit paraboloid in $\mathbb{R}^{d+1}$. For example, let $f(x_1, x_2, a_1, a_2, a_3)$ be the function whose absolute value is some measure of the "distance" between a point $(x_1, x_2) \in \mathbb{R}^2$ and a circle with center (a_1, a_2) and radius a_3, which is the 5-variate polynomial

$$f(x_1, x_2, a_1, a_2, a_3) = a_3^2 - (x_1 - a_1)^2 - (x_2 - a_2)^2 .$$

We can rewrite f in the form

$$f(x_1, x_2, a_1, a_2, a_3) = [a_3^2 - a_1^2 - a_2^2] + [2a_1x_1] + [2a_2x_2] - [x_1^2 + x_2^2] , \tag{3}$$

thus, setting

$$\begin{array}{llll} \psi_0(a) = a_3^2 - a_1^2 - a_2^2, & \psi_1(a) = 2a_1, & \psi_2(a) = 2a_2, & \psi_3(a) = -1, \\ & \varphi_1(x) = x_1, & \varphi_2(x) = x_2, & \varphi_3(x) = x_1^2 + x_2^2, \end{array}$$

we get a linearization of dimension 3. Agarwal and Matoušek [AM94] describe an algorithm that computes a linearization of the smallest dimension under certain mild assumptions.

Returning to the set $\mathcal{F}$, let $\mathcal{H} = \{h_{a^i} \mid 1 \leq i \leq n\}$. It can be verified [AHV04] that a subset $\mathcal{K} \subseteq \mathcal{H}$ is an ε-kernel if and only if the set $\mathcal{G} = \{f_i \mid h_{a^i} \in \mathcal{K}\}$ is an ε-kernel of $\mathcal{F}$.

Combining the linearization technique with Corollary 3.1, one obtains the following result [AHV04].

Theorem 3.2 *Let $\mathcal{F} = \{f_1(x), \ldots, f_n(x)\}$ be a family of d-variate polynomials, where $f_i(x) \equiv f(x, a^i)$ and $a^i \in \mathbb{R}^p$ for each $1 \leq i \leq n$, and $f(x, a)$ is a $(d+p)$-variate polynomial. Suppose that $\mathcal{F}$ admits a linearization of dimension k, and let $\varepsilon > 0$ be a parameter. We can compute an ε-kernel of $\mathcal{F}$ of size $O(1/\varepsilon^{\sigma})$ in time $O(n + 1/\varepsilon^{k-1/2})$, where $\sigma = \min\{d, k/2\}$.*

Let $\mathcal{F} = \left\{(f_1)^{1/r}, \ldots, (f_n)^{1/r}\right\}$, where $r \geq 1$ is an integer and each f_i is a polynomial of some bounded degree. Agarwal *et al.* [AHV04] showed that if $\mathcal{G}$ is an $(\varepsilon/2(r-1))^r$-kernel of $\{f_1, \ldots, f_n\}$, then $\left\{(f_i)^{1/r} \mid f_i \in \mathcal{G}\right\}$ is an ε-kernel of $\mathcal{F}$. Hence, we obtain the following.

Theorem 3.3 *Let $\mathcal{F} = \left\{(f_1)^{1/r}, \ldots, (f_n)^{1/r}\right\}$ be a family of d-variate functions as in Theorem 3.2, each f_i is a polynomial that is non-negative for every $x \in \mathbb{R}^d$, and $r \geq 2$ is an integer constant. Let $\varepsilon > 0$ be a parameter. Suppose that $\mathcal{F}$ admits a linearization of dimension k. We can compute in $O(n + 1/\varepsilon^{r(k-1/2)})$ time an ε-kernel of size $O(1/\varepsilon^{r\sigma})$ where $\sigma = \min\{d, k/2\}$.*

Applications to shape fitting problems. Agarwal *et al.* [AHV04] showed that Theorem 3.3 can be used to compute coresets for a number of unfaithful measures as well. We illustrate the idea by sketching their $(1+\varepsilon)$-approximation algorithm for computing a minimum-width spherical shell that contains $P = \{p_1, \ldots, p_n\}$. A spherical shell is (the closure of) the region bounded by two concentric spheres: the width of the shell is the difference of their radii. Let $f_i(x) = \|x - p_i\|$. Set $\mathcal{F} = \{f_1, \ldots, f_n\}$. Let $w(x, S)$ denote the width of the thinnest spherical shell centered at x that contains a point set S, and let $w^* = w^*(S) = \min_{x \in \mathbb{R}^d} w(x, S)$ be the width of the thinnest spherical shell containing S. Then

$$w(x, P) = \max_{p \in P} \|x - p\| - \min_{p \in P} \|x - p\| = \max_{f_p \in \mathcal{F}} f_p(x) - \min_{f_p \in \mathcal{F}} f_p(x) = \mathfrak{E}_{\mathcal{F}}(x).$$

Let $\mathcal{G}$ be an ε-kernel of $\mathcal{F}$, and suppose $Q \subseteq P$ is the set of points corresponding to $\mathcal{G}$. Then for any $x \in \mathbb{R}^d$, we have $w(x, Q) \geq (1 - \varepsilon) w(x, P)$. So if we first compute $\mathcal{G}$ (and therefore Q) using Theorem 3.3, compute the minimum-width spherical shell A^* containing Q, and take the smallest spherical shell containing P centered at the center of A^*, we get a $(1 + O(\varepsilon))$-approximation to the minimum-width spherical shell containing P. The running time of such an approach is $O(n + f(\varepsilon))$. It is a simple and instructive exercise to translate this approach to the problem of computing a $(1 + \varepsilon)$-approximation of the minimum-width cylindrical shell enclosing a set of points.

Using the kernel framework, Har-Peled and Wang [HW04] have shown that shape fitting problems can be approximated efficiently even in the presence of a few outliers. Let us consider the following problem: Given a set P of n points in $\mathbb{R}^d$, and an integer $1 \leq k \leq n$, find the minimum-width slab that contains $n - k$ points from P. They present an ε-approximation algorithm for this problem whose running time is near-linear in n. They obtain similar results for problems like minimum-width spherical/cylindrical shell and indeed all the shape fitting problems to which the kernel framework applies. Their algorithm works well if the number of outliers k is small. Erickson et al [EHM04] show that for large values of k, say roughly $n/2$, the problem is as hard as the $(d-1)$-dimensional affine degeneracy problem: Given a set of n points (with integer co-ordinates)

in $\mathbb{R}^{d-1}$, do any d of them lie on a common hyperplane? It is widely believed that the affine degeneracy problem requires $\Omega(n^{d-1})$ time.

Points in motion. Theorems 3.2 and 3.3 can be used to maintain various extent measures of a set of moving points. Let $P = \{p_1, \ldots, p_n\}$ be a set of n points in $\mathbb{R}^d$, each moving independently. Let $p_i(t) = (p_{i1}(t), \ldots, p_{id}(t))$ denote the position of point p_i at time t. Set $P(t) = \{p_i(t) \mid 1 \leq i \leq n\}$. If each p_{ij} is a polynomial of degree at most r, we say that the motion of P has *degree* r. We call the motion of P *linear* if $r = 1$ and *algebraic* if r is bounded by a constant.

Given a parameter $\varepsilon > 0$, we call a subset $Q \subseteq P$ an *ε-kernel* of P if for any direction $u \in \mathbb{S}^{d-1}$ and for all $t \in \mathbb{R}$,

$$(1 - \varepsilon)\omega(u, P(t)) \leq \omega(u, Q(t)),$$

where $\omega()$ is the directional width. Assume that the motion of P is linear, i.e., $p_i(t) = a_i + b_i t$, for $1 \leq i \leq n$, where $a_i, b_i \in \mathbb{R}^d$. For a direction $u = (u_1, \ldots, u_d) \in \mathbb{S}^{d-1}$, we define a polynomial

$$\begin{aligned} f_i(u,t) &= \langle p_i(t), u\rangle = \langle a_i + b_i t, u\rangle \\ &= \sum_{j=1}^{d} a_{ij}u_j + \sum_{j=1}^{d} b_{ij} \cdot (tu_j). \end{aligned}$$

Set $\mathcal{F} = \{f_1, \ldots, f_n\}$. Then

$$\omega(u, P(t)) = \max_i \langle p_i(t), u\rangle - \min_i \langle p_i(t), u\rangle = \max_i f_i(u,t) - \min_i f_i(u,t) = \mathfrak{E}_{\mathcal{F}}(u,t).$$

Evidently, $\mathcal{F}$ is a family of $(d+1)$-variate polynomials that admits a linearization of dimension $2d$ (there are $2d$ monomials). Exploiting the fact that $u \in \mathbb{S}^{d-1}$, Agarwal *et al.* [AHV04] show that $\mathcal{F}$ is actually a family of d-variate polynomials that admits a linearization of dimension $2d-1$. Using Theorem 3.2, we can therefore compute an ε-kernel of P of size $O(1/\varepsilon^{d-(1/2)})$ in time $O(n + 1/\varepsilon^{2d-(3/2)})$. The above argument can be extended to higher degree motions in a straightforward manner. The following theorem summarizes the main result.

Theorem 3.4 *Given a set P of n moving points in $\mathbb{R}^d$ whose motion has degree $r > 1$ and a parameter $\varepsilon > 0$, we can compute an ε-kernel Q of P of size $O(1/\varepsilon^d)$ in $O(n + 1/\varepsilon^{(r+1)d-(3/2)})$ time.*

The theorem implies that at any time t, $Q(t)$ is a coreset for $P(t)$ with respect to all faithful measures. Using the same technique, a similar result can be obtained for unfaithful measures such as the minimum-width spherical shell.

Yu *et al.* [YAPV04] have performed experiments with kinetic data structures that maintain the axes-parallel bounding box and convex hull of a set of points P with algebraic motion. They compare the performance of the kinetic data structure for the entire point set P with that of the data structure for a kernel Q computed by methods similar to Theorem 3.4. The experiments indicate that the number of events that the data structure for Q needs to process is significantly lower than for P even when Q is a very good approximation to P.

4 An Incremental Algorithm for Shape Fitting

Let P be a set of n points in $\mathbb{R}^d$. Bădoiu *et al.* [BHI02] gave a simple incremental algorithm for computing an ε-approximation to the minimum-enclosing ball of P. They showed, rather surprisingly, that the number of iterations of their algorithm depends only on ε and is independent of

both d and n. The bound was improved by Bădoiu and Clarkson [BC03b, BC03a] and by Kumar *et al.* [KMY03]. Kumar and Yıldırım [KY04] analyzed a similar algorithm for the minimum-volume enclosing ellipsoid and gave a bound on the number of iterations that is independent of d. The minimum-enclosing ball and minimum-enclosing ellipsoid are convex optimization problems, and it is somewhat surprising that a variant of this iterative algorithm works for non-convex optimization problems, e.g., the minimum-width cylinder, slab, spherical shell, and cylindrical shell containing P. As shown by Yu *et al.* [YAPV04], the number of iterations of the incremental algorithm is independent of the number n of points in P for all of these problems.

We describe here the version of the algorithm for computing the minimum-width slab containing P. The algorithm and its proof of convergence are readily translated to the other problems mentioned. Let Q be any affinely independent subset of $d+1$ points in P.

1. Let S be the minimum-width slab containing Q, computed by some brute-force method. If a $(1+\varepsilon)$-expansion of S contains P, we return this $(1+\varepsilon)$-expansion.
2. Otherwise, let $p \in P$ be the point farthest from S.
3. Set $Q = Q \cup \{p\}$ and go to Step 1.

It is clear that when the algorithm terminates, it does so with an ε-approximation to the minimum-width slab containing P. Also, the running time of the algorithm is $O(k(n + f(O(k))))$, where k is the number of iterations of the algorithm, and $f(t)$ is the running time of the brute-force algorithm for computing a minimum-enclosing slab of t points. Following an argument similar to the one used for proving the correctness of the algorithm for constructing ε-kernels, Yu *et al.* [YAPV04] proved that the above algorithm converges within $O(1/\varepsilon^{(d-1)/2})$ iterations. They also do an experimental analysis of this algorithm and conclude that its typical performance is quite good in comparison with even the coreset based algorithms. This is because the number of iterations for typical point sets is quite small, as might be expected. See the original paper for details.

We conclude this section with an interesting open problem: Does the incremental algorithm for the minimum-enclosing cylinder problem terminate in $O(f(d) \cdot g(d, \varepsilon))$ iterations, where $f(d)$ is a function of d only, and $g(d, \varepsilon)$ is a function that depends only polynomially on d? Note that the algorithm for the minimum-enclosing ball terminates in $O(1/\varepsilon)$ iterations, while the algorithm for the minimum-enclosing slab can be shown to require $\Omega(1/\varepsilon^{(d-1)/2})$ iterations.

5 Coresets in a Streaming Setting

Algorithms for computing an ε-kernel for a given set of points in $\mathbb{R}^d$ can be adapted for efficiently maintaining an ε-kernel of a set of points under insertions and deletions [AHV04]. Here we describe the algorithm of Agarwal *et al.* [AHV04] for maintaining ε-kernels in the streaming setting. Suppose we are receiving a stream of points $p_1, p_2, \ldots$ in $\mathbb{R}^d$. Given a parameter $\varepsilon > 0$, we wish to maintain an ε-kernel of the n points received so far. The resource that we are interested in minimizing is the space used by the data structure. Note that our analysis is in terms of n, the number of points inserted into the data structure. However, n does not need to be specified in advance. We assume the existence of an algorithm $\mathbb{A}$ that can compute a δ-kernel of a subset $S \subseteq P$ of size $O(1/\delta^k)$ in time $O(|S| + T_{\mathbb{A}}(\delta))$; obviously $T_{\mathbb{A}}(\delta) = \Omega(1/\delta^k)$. We will use $\mathbb{A}$ to maintain an ε-kernel dynamically. Besides such an algorithm, our scheme only uses abstract properties of kernels such as the following:

(1) If P_2 is an ε-kernel of P_1, and P_3 is a δ-kernel of P_2, then P_3 is a $(\delta + \varepsilon)$-kernel of P_1;

(2) If P_2 is an ε-kernel of P_1, and Q_2 is an ε-kernel of Q_1, then $P_2 \cup Q_2$ is an ε-kernel of $P_1 \cup Q_1$.[2]

Thus the scheme applies more generally, for instance, to some notions of coresets defined in the clustering context.

We assume without loss of generality that $1/\varepsilon$ is an integer. We use the dynamization technique of Bentley and Saxe [BS80], as follows: Let $P = \langle p_1, \ldots, p_n \rangle$ be the sequence of points that we have received so far. For integers $i \geq 1$, let $\rho_i = \varepsilon/ci^2$, where $c > 0$ is a constant, and set $\delta_i = \prod_{l=1}^{i}(1+\rho_l) - 1$. We partition P into subsets $P_0, P_1, \ldots, P_u$, where $u = \lfloor \log_2 \varepsilon^k n \rfloor + 1$, as follows. $|P_0| = n \mod 1/\varepsilon^k$, and for $1 \leq i \leq u$, $|P_i| = 2^{i-1}/\varepsilon^k$ if the ith rightmost bit in the binary expansion of $\lfloor \varepsilon^k n \rfloor$ is 1, otherwise $|P_i| = 0$. Furthermore, if $0 \leq i < j \leq u$, the points in P_j arrived before any point in P_i. These conditions uniquely specify $P_0, \ldots, P_u$. We refer to i as the *rank* of P_i. Note that for $i \geq 1$, there is at most one non-empty subset of rank i.

Unlike the standard Bentley-Saxe technique, we do not maintain each P_i explicitly. Instead, for each non-empty subset P_i, we maintain a δ_i-kernel Q_i of P_i; if $P_i = \emptyset$, we set $Q_i = \emptyset$ as well. We also let $Q_0 = P_0$. Since

$$1 + \delta_i = \prod_{l=1}^{i}\Big(1 + \frac{\varepsilon}{cl^2}\Big) \leq \exp\Big(\sum_{l=1}^{i}\frac{\varepsilon}{cl^2}\Big) = \exp\Big(\frac{\varepsilon}{c}\sum_{l=1}^{i}\frac{1}{l^2}\Big) \leq \exp\Big(\frac{\pi^2\varepsilon}{6c}\Big) \leq 1 + \frac{\varepsilon}{3}, \tag{4}$$

provided c is chosen sufficiently large, Q_i is an $(\varepsilon/3)$-kernel of P_i. Therefore, $\bigcup_{i=0}^{u} Q_i$ is an $(\varepsilon/3)$-kernel of P. We define the *rank* of a set Q_i to be i. For $i \geq 1$, if P_i is non-empty, $|Q_i|$ will be $O(1/\rho_i^k)$ because $\rho_i \leq \delta_i$; note that $|Q_0| = |P_0| < 1/\varepsilon^k$.

For each $i \geq 0$, we also maintain an $\varepsilon/3$-kernel K_i of $\bigcup_{j \geq i} Q_j$, as follows. Let $u = \lfloor \log_2(\varepsilon^k n) \rfloor + 1$ be the largest value of i for which P_i is non-empty. We have $K_u = Q_u$, and for $1 \leq i < u$, K_i is a ρ_i-kernel of $K_{i+1} \cup Q_i$. Finally, $K_0 = Q_0 \cup K_1$. The argument in (4), by the coreset properties (1) and (2), implies that K_i is an $(\varepsilon/3)$-kernel of $\bigcup_{j \geq i} Q_j$, and thus K_0 is the required ε-kernel of P. The size of the entire data structure is

$$\begin{aligned}\sum_{i=0}^{u}(|Q_i| + |K_i|) &\leq |Q_0| + |K_0| + \sum_{i=1}^{u} O(1/\rho_i^k) \\ &= O(1/\varepsilon^k) + \sum_{i=1}^{\lfloor \log_2 \varepsilon^k n \rfloor + 1} O\Big(\frac{i^{2k}}{\varepsilon^k}\Big) = O\Big(\frac{\log^{2k+1} n}{\varepsilon^k}\Big).\end{aligned}$$

At the arrival of the next point p_{n+1}, the data structure is updated as follows. We add p_{n+1} to Q_0 (and conceptually to P_0). If $|Q_0| < 1/\varepsilon^k$ then we are done. Otherwise, we promote Q_0 to have rank 1. Next, if there are two δ_j-kernels Q_x, Q_y of rank j, for some $j \leq \lfloor \log_2 \varepsilon^k(n+1) \rfloor + 1$, we compute a ρ_{j+1}-kernel Q_z of $Q_x \cup Q_y$ using algorithm $\mathbb{A}$, set the rank of Q_z to $j+1$, and discard the sets Q_x and Q_y. By construction, Q_z is a δ_{j+1}-kernel of $P_z = P_x \cup P_y$ of size $O(1/\rho_{j+1}^k)$ and $|P_z| = 2^j/\varepsilon^k$. We repeat this step until the ranks of all Q_i's are distinct. Suppose ξ is the maximum rank of a Q_i that was reconstructed, then we recompute $K_\xi, \ldots, K_0$ in that order. That is, for $\xi \geq i \geq 1$, we compute a ρ_i-kernel of $K_{i+1} \cup Q_i$ and set this to be K_i; finally, we set $K_0 = K_1 \cup Q_0$.

[2]This property is, strictly speaking, not true for kernels. However, if we slightly modify the definition to say that $Q \subseteq P$ is an ε-kernel of P if the $1/(1-\varepsilon)$-expansion of any slab that contains Q also contains P, both properties are seen to hold. Since the modified definition is intimately connected with the definition we use, we feel justified in pretending that the second property also holds for kernels.

For any fixed $i \geq 1$, Q_i and K_i are constructed after every $2^{i-1}/\varepsilon^k$ insertions, therefore the amortized time spent in updating Q after inserting a point is

$$\sum_{i=1}^{\lfloor \log_2 \varepsilon^k n \rfloor + 1} \frac{\varepsilon^k}{2^{i-1}} O\Big(\frac{i^{2k}}{\varepsilon^k} + T_{\mathbb{A}}\Big(\frac{\varepsilon}{ci^2}\Big)\Big) = O\Big(\sum_{i=1}^{\lfloor \log_2 \varepsilon^k n \rfloor + 1} \frac{\varepsilon^k}{2^{i-1}} T_{\mathbb{A}}\Big(\frac{\varepsilon}{ci^2}\Big)\Big).$$

If $T_{\mathbb{A}}(x)$ is bounded by a polynomial in $1/x$, then the above expression is bounded by $O(\varepsilon^k T_{\mathbb{A}}(\varepsilon))$.

Theorem 5.1 ([AHV04]) *Let P be a stream of points in $\mathbb{R}^d$, and let $\varepsilon > 0$ be a parameter. Suppose that for any subset $S \subseteq P$, we can compute an ε-kernel of S of size $O(1/\varepsilon^k)$ in $O(|S| + T_{\mathbb{A}}(\varepsilon))$ time, where $T_{\mathbb{A}}(\varepsilon) \geq 1/\varepsilon^k$ is bounded by a polynomial in $1/\varepsilon$. Then we can maintain an ε-kernel of P of size $O(1/\varepsilon^k)$ using a data structure of size $O(\log^{2k+1}(n)/\varepsilon^k)$. The amortized time to insert a point is $O(\varepsilon^k T_{\mathbb{A}}(\varepsilon))$, and the running time in the worst case is $O\big((\log^{2k+1} n)/\varepsilon^k + T_{\mathbb{A}}(\varepsilon/\log^2 n)\log n\big)$.*

Combined with Theorem 2.2, we get a data-structure using $(\log n/\varepsilon)^{O(d)}$ space to maintain an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ using $(1/\varepsilon)^{O(d)}$ amortized time for each insertion.

Improvements. The previous scheme raises the question of whether there is a data structure that uses space independent of the size of the point set to maintain an ε-kernel. Chan [Cha04] shows that the answer is "yes" by presenting a scheme that uses only $(1/\varepsilon)^{O(d)}$ storage. This result implies a similar result for maintaining coresets for all the extent measures that can be handled by the framework of kernels. His scheme is somewhat involved, but the main ideas and difficulties are illustrated by a simple scheme, reproduced below, that he describes that uses constant storage for maintaining a constant-factor approximation to the radius of the smallest enclosing cylinder containing the point set. We emphasize that the question is that of maintaining an approximation to the radius: it is not hard to maintain the axis of an approximately optimal cylinder.

A very simple constant-factor offline algorithm for approximating the minimum-width cylinder enclosing a set P of points was proposed by Agarwal *et al.* [AAS01]. The algorithm picks an arbitrary input point, say o, finds the farthest point v from o, and returns the farthest point from the line $\overline{ov}$.

Let $\mathrm{rad}(P)$ denote the minimum radius of all cylinders enclosing P, and let $d(p, \ell)$ denote the distance between point p and line ℓ. The following observation immediately implies an upper bound of 4 on the approximation factor of the above algorithm.

Observation 5.2 $d(p, \overline{ov}) \leq 2\Big(\dfrac{\|o-p\|}{\|o-v\|} + 1\Big)\,\mathrm{rad}(\{o, v, p\})$.

Unfortunately, the above algorithm requires two passes, one to find v and one to find the radius, and thus does not fit in the streaming framework. Nevertheless, a simple variant of the algorithm, which maintains an approximate candidate for v on-line, works, albeit with a larger constant:

Theorem 5.3 ([Cha04]) *Given a stream of points in $\mathbb{R}^d$ (where d is not necessarily constant), we can maintain a factor-18 approximation of the minimum radius over all enclosing cylinders with $O(d)$ space and update time.*

Proof: Initially, say o and v are the first two points, and set $w = 0$. We may assume that o is the origin. A new point is inserted as follows:

`insert`(p):

1. $w := \max\{w, \mathrm{rad}(\{o, v, p\})\}$
2. if $\|p\| > 2\|v\|$ then $v := p$
3. Return w

Note that after each point is inserted, the algorithm returns a quantity that is shown below to be an approximation to the radius of the smallest enclosing cylinder of all the points inserted thus far.

In the following analysis, w_f and v_f refer to the final values of w and v, and v_i refers to the value of v after its i-th change. Note that $\|v_i\| > 2\|v_{i-1}\|$ for all i. Also, we have $w_f \geq \mathrm{rad}(\{o, v_{i-1}, v_i\})$ since $\mathrm{rad}(\{o, v_{i-1}, v_i\})$ was one of the "candidates" for w. From Observation 5.2, it follows that

$$d(v_{i-1}, \overline{ov_i}) \leq 2\left(\frac{\|v_{i-1}\|}{\|v_i\|} + 1\right)\mathrm{rad}(\{o, v_{i-1}, v_i\}) \leq 3\mathrm{rad}(\{o, v_{i-1}, v_i\}) \leq 3w_f.$$

Fix a point $q \in P$, where P denotes the entire input point set. Suppose that $v = v_j$ just after q is inserted. Since $\|q\| \leq 2\|v_j\|$, Observation 5.2 implies that $d(q, \overline{ov_j}) \leq 6w_f$.

For $i > j$, we have $d(q, \overline{ov_i}) \leq d(q, \overline{ov_{i-1}}) + d(\hat{q}, \overline{ov_i})$, where $\hat{q}$ is the orthogonal projection of q to $\overline{ov_{i-1}}$. By similarity of triangles,

$$d(\hat{q}, \overline{ov_i}) = (\|\hat{q}\| / \|v_{i-1}\|)d(v_{i-1}, \overline{ov_i}) \leq (\|q\| / \|v_{i-1}\|)3w_f.$$

Therefore,

$$d(q, \overline{ov_i}) \leq \begin{cases} 6w_f & \text{if } i = j, \\ d(q, \overline{ov_{i-1}}) + \dfrac{\|q\|}{\|v_{i-1}\|}3w_f & \text{if } i > j. \end{cases}$$

Expanding the recurrence, one can obtain that $d(q, \overline{ov_f}) \leq 18w_f$. So, $w_f \leq \mathrm{rad}(P) \leq 18w_f$. ∎

6 Coresets for Clustering

Given a set P of n points in $\mathbb{R}^d$ and an integer $k > 0$, a typical clustering problem asks for partitioning P into k subsets (called *clusters*), $P_1, \ldots, P_k$, so that certain *objective* function is minimzed. Given a function μ that measures the extent of a cluster, we consider two types of clustering objective functions: *centered* clustering in which the objective function is $\max_{1 \leq i \leq k} \mu(P_i)$, and the *summed* clustering in which the objective function is $\sum_{i=1}^{k} \mu(P_i)$; k-center and k-line-center are two examples of the first type, and k-median and k-means are two examples of the second type.

It is natural to ask whether coresets can be used to compute clusterings efficiently. In the previous sections we showed that an ε-kernel of a point set provides a coreset for several extent measures of P. However, the notion of ε-kernel is too weak to provide a coreset for a clustering problem because it approximates the extent of the entire P while for clustering problems we need a subset that approximates the extent of "relevant" subsets of P as well. Nevertheless, coresets exist for many clustering problems, though the precise definition of coreset depends on the type of clustering problem we are considering. We review some of these results in this section.

6.1 k-center and its variants

We begin by defining generalized k-clustering: we call a *cluster* to be a pair (f, S), where f is a q-dimensional subspace for some $q \leq d$ and $S \subseteq P$. Define $\mu(f, S) = \max_{p \in S} d(p, f)$. We define $\mathcal{B}(f, r)$ to be the Minkowski sum of f and the ball of radius r centered at the origin; $\mathcal{B}(f, r)$ is a ball

(resp. cylinder) of radius r if f is a point (resp. line), and a slab of width $2r$ if f is a hyperplane. Obviously, $S \subseteq \mathcal{B}(f, \mu(f, S))$. We call $\mathcal{C} = \{(f_1, P_1), \ldots, (f_k, P_k)\}$ a *k-clustering* (of dimension q) if each f_i is a q-dimensional subspace and $P = \bigcup_{i=1}^{k} P_i$. We define $\mu(\mathcal{C}) = \max_{1 \le i \le k} \mu(f_i, P_i)$, and set $r_{\text{opt}}(P, k, q) = \min_{\mathcal{C}} \mu(\mathcal{C})$, where the minimum is taken over all k-clusterings (of dimension q) of P. We use $C_{\text{opt}}(P, k, q)$ to denote an optimal k-clustering (of dimension q) of P. For $q = 0, 1, d-1$, the above clustering problems are called k-center, k-line-center, and k-hyperplane-center problems, respectively; they are equivalent to covering P by k balls, cylinders, and slabs of minimum radius, respectively.

We call $Q \subseteq P$ an *additive ε-coreset* of P if for every k-clustering $\mathcal{C} = \{(f_1, Q_1), \ldots, (f_k, Q_k)\}$ of Q, with $r_i = \mu(f_i, Q_i)$,

$$P \subseteq \bigcup_{i=1}^{k} \mathcal{B}(f_i, r_i + \varepsilon \mu(\mathcal{C})),$$

i.e., union of the expansion of each $\mathcal{B}(f_i, r_i)$ by $\varepsilon\mu(\mathcal{C})$ covers P. If the following stronger property is also true for all k-clusterings $\mathcal{C}$

$$P \subseteq \bigcup_{i=1}^{k} \mathcal{B}(f_i, (1 + \varepsilon) r_i),$$

then we call Q a *multiplicative ε-coreset*.

We review the known results on additive and multiplicative coreset for k-center, k-line-center, and k-hyperplane-center.

k-center. The existence of an additive coreset for k-center follows from the following simple observation. Let $r^* = r_{\text{opt}}(P, k, 0)$, and let $\mathcal{B} = \{B_1, \ldots, B_k\}$ be a family of k balls of radius r^* that cover P. Draw a d-dimensional Cartesian grid of side length $\varepsilon r^*/2d$; $O(k/\varepsilon^d)$ of these grid cells intersect the balls in $\mathcal{B}$. For each such cell τ that also contains a point of P, we arbitrarily choose a point from $P \cap \tau$. The resulting set $\mathcal{S}$ of $O(k/\varepsilon^d)$ points is an additive ε-coreset of P, as proved by Agarwal and Procopiuc [AP02]. In order to construct $\mathcal{S}$ efficiently, we use Gonzalez's greedy algorithm [Gon85] to compute a factor-2 approximation of k-center, which returns a value $\tilde{r} \le 2r^*$. We then draw the grid of side length $\varepsilon\tilde{r}/4d$ and proceed as above. Using a fast implementation of Gonzalez's algorithm proposed in [FG88, Har04a], one can compute an additive ε-coreset of size $O(k/\varepsilon^d)$ in time $O(n + k/\varepsilon^d)$.

Agarwal *et al.* [APV02] proved the existence of a small multiplicative ε-coreset for k-center in $\mathbb{R}^1$. It was subsequently extended to higher dimensions by Har-Peled [Har04b]. We sketch their construction.

Theorem 6.1 ([APV02, Har04b]) *Let P be a set of n points in $\mathbb{R}^d$, and $0 < \varepsilon < 1/2$ a parameter. There exists a multiplicative ε-coreset of size $O\big(k!/\varepsilon^{dk}\big)$ of P for k-center.*

Proof: For $k = 1$, by definition, an additive ε-coreset of P is also a multiplicative ε-coreset of P. For $k > 1$, let $r^* = r_{\text{opt}}(P, k, 0)$, the smallest r for which k balls of radius r cover P. We draw a d-dimensional grid of side length $\varepsilon r_{\text{opt}}/(5d)$, and let $\mathbb{C}$ be the set of (hyper-)cubes of this grid that contain points of P. Clearly, $|\mathbb{C}| = O(k/\varepsilon^d)$. Let Q' be an additive $(\varepsilon/2)$-coreset of P. For every cell Δ in $\mathbb{C}$, we inductively compute an ε-multiplicative coreset of $P \cap \Delta$ with respect to $(k-1)$-center. Let Q_Δ be this set, and let $\mathcal{Q} = \bigcup_{\Delta \in \mathbb{C}} Q_\Delta \cup Q'$. We argue below that the set $\mathcal{Q}$ is the required multiplicative coreset. The bound on its size follows by a simple calculation.

Let $\mathcal{B}$ be any family of k balls that covers Q. Consider any hypercube Δ of $\mathbb{C}$. Suppose Δ intersects all the k balls of $\mathcal{B}$. Since Q' is an additive $(\varepsilon/2)$-coreset of P, one of the balls in $\mathcal{B}$ must

be of radius at least $r^*/(1+\varepsilon/2) \geq r^*(1-\varepsilon/2)$. Clearly, if we expand such a ball by a factor of $(1+\varepsilon)$, it completely covers Δ, and therefore also covers all the points of $\Delta \cap P$.

We now consider the case when Δ intersects at most $k-1$ balls of $\mathcal{B}$. By induction, $Q_\Delta \subseteq \mathcal{Q}$ is an ε-multiplicative coreset of $P \cap \Delta$ for $(k-1)$-center. Therefore, if we expand each ball in $\mathcal{B}$ that intersects Δ by a factor of $(1+\varepsilon)$, the resulting set of balls will cover $P \cap \Delta$. ∎

Surprisingly, additive coresets for k-center exist even for a set of moving points in $\mathbb{R}^d$. More precisely, let P be a set of n points in $\mathbb{R}^d$ with algebraic motion of degree at most Δ, and let $0 < \varepsilon \leq 1/2$ be a parameter. Har-Peled [Har04a] showed that there exists a subset $Q \subseteq P$ of size $O((k/\varepsilon^d)^{\Delta+1})$ so that for all $t \in \mathbb{R}$, $Q(t)$ is an additive ε-coreset of $P(t)$. For $k = O(n^{1/4}\varepsilon^d)$, Q can be computed in time $O(nk/\varepsilon^d)$.

k-line-center. The existence of an additive coreset for k-line-center, i.e., for the problem of covering P by k congruent cylinders of the minimum radius, was first proved by Agarwal *et al.* [APV02].

Theorem 6.2 ([APV02]) *Given a set P of finite points in $\mathbb{R}^d$ and a parameter $0 < \varepsilon < 1/2$, there exists an additive ε-coreset of size $O((k+1)!/\varepsilon^{d-1+k})$ of P for the k-line-center problem.*

Proof: Let $C_{\text{opt}} = \{(\ell_1, P_1), \ldots, (\ell_k, P_k)\}$ be an optimal k-clustering (of dimension 1) of P, and let $r^* = \mu(P, k, 1)$, i.e., the cylinders of radius r^* with axes $\ell_1, \ldots, \ell_k$ cover P and $P_i \subset \mathcal{B}(\ell_i, r^*)$. For each $1 \leq i \leq k$, draw a family L_i of $O(1/\varepsilon^{d-1})$ lines parallel to ℓ_i so that for any point in P_i there is a line in L_i within distance $\varepsilon r^*/2$. Set $L = \bigcup_i L_i$. We project each point $p \in P_i$ to the line in L_i that is nearest to p. Let $\bar{p}$ be the resulting projection of p, and let $\bar{P}_\ell$ be the set of points that project onto $\ell \in L$. Set $\bar{P} = \bigcup_\ell \bar{P}_\ell$. It can be argued that a multiplicative $(\varepsilon/3)$-coreset of $\bar{P}$ is an additive ε-coreset of P. Since the points in $\bar{P}_\ell$ lie on a line, by Theorem 6.1, a multiplicative $(\varepsilon/3)$-coreset $\bar{Q}_\ell$ of $\bar{P}_\ell$ of size $O(k!/\varepsilon^k)$ exists. Observing that $\bar{Q} = \bigcup_{\ell \in L} \bar{Q}_\ell$ is a multiplicative $(\varepsilon/3)$-coreset of $\bar{P}$, and thus $Q = \{p \mid \bar{p} \in \bar{Q}\}$ is an additive ε-coreset of P of size $O((k+1)!/\varepsilon^{d-1+k})$. ∎

Although Theorem 6.2 proves the existence of an additive coreset for k-line-center, the proof is non-constructive. However, Agarwal *et al.* [APV02] have shown that the iterated reweighting technique of Clarkson [Cla93] can be used in conjunction with Theorem 6.2 to compute an ε-approximate solution to the k-line-center problem in $O(n \log n)$ expected time, with constants depending on k, ε, and d.

When coresets do not exist. We now present two negative results on coresets for centered clustering problems. Surprisingly, there are no multiplicative coresets for k-line-center even in $\mathbb{R}^2$.

Theorem 6.3 ([Har04b]) *For any $n \geq 3$, there exists a point set $P = \{p_1, \ldots, p_n\}$ in $\mathbb{R}^2$, such that the size of any multiplicative $(1/2)$-coreset of P with for 2-line-center is at least $|P| - 2$.*

Proof: Let $p_i = (1/2^i, 2^i)$ and $P(i) = \{p_1, \ldots, p_i\}$. Let Q be a $(1/2)$-coreset of $P = P(n)$. Let $Q_i^- = Q \cap P(i)$ and $Q_i^+ = Q \setminus Q_i^-$.

If the set Q does not contain the point $p_i = \left(1/2^i, 2^i\right)$, for some $2 \leq i \leq n-1$, then Q_i^- can be covered by a horizontal strip h^- of width $\leq 2^{i-1}$ that has the x-axis as its lower boundary. Clearly, if we expand h^- by a factor of $3/2$, it still will not cover p_i. Similarly, we can cover Q_i^+ by a vertical strip h^+ of width $1/2^{i+1}$ that has the y-axis as its left boundary. Again, if we expand h^+ by a factor of $3/2$, it will still not cover p_i. We conclude, that any multiplicative $(1/2)$-coreset for P must include all the points $p_2, p_3, \ldots, p_{n-1}$. ∎

This construction can be embedded in $\mathbb{R}^3$, as described by Har-Peled [Har04b], to show that even an additive coreset does not exist for 2-plane-clustering in $\mathbb{R}^3$, i.e., the problem of covering the input point set of two slabs of the minimum width.

For the special case of 2-plane-center in $\mathbb{R}^3$, a near-linear-time approximation algorithm is known [Har04b]. The problem of approximating the best k-hyperplane-clustering for $k \geq 3$ in $\mathbb{R}^3$ and $k \geq 2$ in higher dimensions in near-linear time is still open.

k-median and k-means clustering. Next we focus our attention to coresets for the summed clustering problem. For simplicity, we consider the k-median clustering problem, which calls for computing k "facility" points so that the average distance between the points of C and their nearest facility is minimized. Since the objective function involves sum of distances, we need to assign weights to points in coresets to approximate the objective function of the clustering for the entire point set. We therefore define k-median clustering for a weighted point set.

Let P be a set of n points in $\mathbb{R}^d$, and let $w : P \to \mathbb{Z}^+$ be a weight function. For a point set $C \subseteq \mathbb{R}^d$, let $\mu(P, w, C) = \sum_{p \in P} w(p) d(p, C)$, where $d(p, C) = \min_{q \in C} d(p, q)$. Given C, we partition P into k clusters by assigning each point in P to its nearest neighbor in C. Define

$$\mu(P, w, k) = \min_{C \subset \mathbb{R}^d, |C|=k} \mu(P, w, C).$$

For $k = 1$, it is the so-called Fermat-Weber problem [Wes93]. A subset $Q \subseteq P$ with a weight function $\chi : P \to \mathbb{Z}^+$ is called an *ε-coreset for k-median* if for any set C of k points in $\mathbb{R}^d$,

$$(1 - \varepsilon)\mu(P, w, C) \leq \mu(Q, \chi, C) \leq (1 + \varepsilon)\mu(P, w, C).$$

Here we sketch the proof by Har-Peled and Mazumdar [HM04] for the existence of a small coreset for the k-median problem. There are two main ingredients in their construction. First suppose we have at our disposal a set $A = \{a_1, \ldots, a_m\}$ of "support" points in $\mathbb{R}^d$ so that $\mu(P, w, A) \leq c\mu(P, w, k)$ for a constant $c \geq 1$, i.e., A is a good approximation of the "centers" of an optimal k-median clustering. We construct an ε-coreset $\mathcal{S}$ of size $O((|A| \log n)/\varepsilon^d)$ using A, as follows.

Let $P_i \subseteq P$, for $1 \leq i \leq m$, be the set of points for which a_i is the nearest neighbor in A. We draw an exponential grid around a_i and choose a subset of $O((\log n)/\varepsilon^d)$ points of P_i, with appropriate weights, for $\mathcal{S}$. Set $\rho = \mu(P, w, A)/cn$, which is a lower bound on the average radius $\mu(P, w, k)/n$ of the optimal k-median clustering. Let $\mathbb{C}_j$ be the axis-parallel hypercube with side length $\rho 2^j$ centered at a_i, for $0 \leq i \leq \lceil 2 \log(cn) \rceil$. Set $V_0 = \mathbb{C}_0$ and $V_i = \mathbb{C}_i \setminus \mathbb{C}_{i-1}$ for $i \geq 1$. We partition each V_i into a grid of side length $\varepsilon \rho 2^j / \alpha$, where $\alpha \geq 1$ is a constant. For each grid cell τ in the resulting exponential grid that contains at least one point of P_i, we choose an arbitrary point in $P_i \cap \tau$ and set its weight to $\sum_{p \in P_i \cap \tau} w(p)$. Let $\mathcal{S}_i$ be the resulting set of weighted points. We repeat this step for all points in A, and set $\mathcal{S} = \bigcup_{i=1}^m \mathcal{S}_i$. Har-Peled and Mazumdar showed that $\mathcal{S}$ is indeed an ε-coreset of P for the k-median problem, provided α is chosen appropriately.

The second ingredient of their construction is the existence of a small "support" set A. Initially, a random sample of P of $O(k \log n)$ points is chosen and the points of P that are "well-served" by this set of random centers is filtered out. The process is repeated for the remaining points of P until we get a set A' of $O(k \log^2 n)$ support points. Using the above procedure, we can construct an $(1/2)$-coreset $\mathcal{S}$ of size $O(k \log^3 n)$. Next, a simple polynomial-time local-search algorithm, described in [HM04], can be applied to this coreset and a support set A of size k can be constructed, which is a constant-factor approximation to the optimal k-median/means clustering. Plugging this A back into the above coreset construction yields an ε-coreset of size $O((k/\varepsilon^d) \log n)$.

Theorem 6.4 ([HM04]) *Given a set P of n points in $\mathbb{R}^d$, and parameters $\varepsilon > 0$ andk, one can compute a coreset of P for k-means and k-median clustering of size $O((k/\varepsilon^d) \log n)$. The running time of this algorithm is $O(n + \text{poly}(k, \log n, 1/\varepsilon))$, where $\text{poly}(\cdot)$ is a polynomial.*

Using a more involved construction, Har-Peled and Kushal [HK04] showed that for both k-median and k-means clustering, one can construct a coreset whose size is independent of the size of the input point set. In particular, they show that there is a coreset of size $O(k^2/\varepsilon^d)$ for k-median and $O(k^3/\varepsilon^{d+1})$ for k-means.

7 Coresets in High Dimensions

Most of the coreset constructions have exponential dependence on the dimensions. In this section, we do not consider d to be a fixed constant but assume that it can be as large as the number of input points. It is natural to ask whether the dependence on the dimension can be reduced or removed altogether. For example, consider a set P of n points in $\mathbb{R}^d$. A 2-approximate coreset for the minimum enclosing ball of P has size 2 (just pick a point in P, and its furthest neighbor in P). Thus, dimension-independent coresets do exist.

As another example, consider the question of whether a small coreset exists for the width measure of P (i.e., the width of the thinnest slab containing P). It is easy to verify that any ε-approximate coreset for the width needs to be of size at least $1/\varepsilon^{\Omega((d-1)/2)}$. Indeed, consider spherical cap on the unit hypersphere, with angular radius $c\sqrt{\varepsilon}$, for appropriate constant c. The height of this cap is $1 - \cos(c\sqrt{\varepsilon}) \leq 2\varepsilon$. Thus, a coreset of the hypersphere, for the measure of width, in high dimension, would require any such cap to contain at least one point of the coreset. As such, its size must be exponential, and we conclude that high-dimensional coresets (with size polynomial in the dimension) do not always exist.

7.1 Minimum enclosing ball

Given a set of points P, an approximation of the minimum radius ball enclosing P can be computed in polynomial time using the ellipsoid method since this is a quadratic convex programming problem [Gär95, GLS88]. However, the natural question is whether one can compute a small coreset, $Q \subseteq P$, such that the minimum enclosing ball for Q is a good approximation to the real minimum enclosing ball.

Bădoiu *et al.* [BHI02] presented an algorithm, which we have already mentioned in Section 4, that generates a coreset of size $O(1/\varepsilon^2)$. The algorithms starts with a set C_0 that contains a single (arbitrary) point of P. Next, in the ith iteration, the algorithm computes the smallest enclosing ball for C_{i-1}. If the $(1+\varepsilon)$-expansion of the ball contains P, then we are done, as we have computed the required coreset. Otherwise, take the point from P furthest from the center of the ball and add it to the coreset. Bădoiu *et al.* [BHI02] showed that this algorithm terminates within $O(1/\varepsilon^2)$ iterations. The bound was later improved to $O(1/\varepsilon)$ by Kumar *et al.* [KMY03] and Bădoiu and Clarkson [BC03b]. Bădoiu and Clarkson showed a matching lower bound and gave an elementary algorithm that uses the "hill climbing" technique. Using this algorithm instead of the ellipsoid method, we obtain a simple algorithm with running time $O(dn/\varepsilon + 1/\varepsilon^{O(1)})$ [BC03a].

It is important to note that this coreset Q is *weaker* than its low dimensional counterpart: it is not necessarily true that the $(1+\varepsilon)$-expansion of *any* ball containing Q contains P. What is true is that the smallest ball containing Q, when $(1+\varepsilon)$-expanded, contains P. In fact, it is easy to verify that the size of a coreset guaranteeing the stronger property is exponential in the dimension in the worst case.

Smallest enclosing ball with outliers. As an application of this coreset, one can compute approximately the smallest ball containing all but k of the points. Indeed, consider the smallest

such ball b_{opt}, and consider $P' = P \cap b_{\text{opt}}$. There is a coreset $Q \subseteq P'$ such that (1) $|Q| = O(1/\varepsilon)$ and (2) the smallest enclosing ball for Q, if ε-expanded, contains at least $n-k$ points of P. Thus, one can just enumerate all possible subsets of size $O(1/\varepsilon)$ as "candidates" for Q, and for each such subset, compute its smallest enclosing ball, expand the ball, and check how many points of P it contains. Finally, the smallest candidate ball that contains at least $n-k$ points of P is the required approximation. The running time of this algorithm is $dn^{O(1/\varepsilon)}$.

k-center. We execute k copies of the incremental algorithm for the min-enclosing ball together. Whenever getting a new point, we need to determine to which of the k clusters it belongs to. To this end, we ask an oracle to identify the cluster it belongs to. It is easy to verify that this algorithm generates an ε-approximate k-center clustering in k/ε iterations. The running time is $O(dkn/\varepsilon + dk/\varepsilon^{O(1)})$.

To remove the oracle, which generates $O(k/\varepsilon)$ integer numbers between 1 and k, we just generate all possible sequence answers that the oracle might give. Since there are $O(k^{O(k/\varepsilon)})$ sequences, we get that the running time of the new algorithm (which is oracle free) is $O(dnk^{O(k/\varepsilon)})$. One can even handle outliers; see [BC03a] for details.

7.2 Minimum enclosing cylinder

One natural problem is the computation of a cylinder of minimum radius containing the points of P. We saw in Section 5 that the line through any point in P and its furthest neighbor is the axis for a constant-factor approximation. In [HV02], Har-Peled and Varadarajan showed that there is a subset $Q \subseteq P$ of $(1/\varepsilon)^{O(1)}$ points such that the axis of an ε-approximate cylinder lies in the subspace spanned by Q. By enumerating all possible candidates for Q, and solving a "low-dimensional" problem for each of the resulting candidate subspaces, they obtain an algorithm that runs in $dn^{(1/\varepsilon)^{O(1)}}$ time. A slightly faster, but more involved algorithm, was described earlier by Bădoiu *et al.* [BHI02].

The algorithm of Har-Peled and Varadarajan extends immediately to the problem of computing a k-flat (i.e., an affine subspace of dimension k) that minimizes the maximum distance to a point in P. The resulting running time is $dn^{(k/\varepsilon)^{O(1)}}$. The approach also handles outliers and multiple flats.

Linear-time algorithm. A natural approach for improving the running time of the minimum enclosing cylinder, is to try and adapt the general approach underlying the algorithm of Bădoiu and Clarkson [BC03a] to the cylinder case. Here, the idea is that we start from a center line ℓ_0. At each iteration, we find the furthest point $p_i \in P$ from ℓ_{i-1}. We then generate a line ℓ_i which is "closer" to the optimal center line. This can be done by consulting with an oracle, that provides us with information about how to move the line. By careful implementation, and removing the oracle, the resulting algorithm takes $O(ndC_\varepsilon)$ time, where $C_\varepsilon = \exp\bigl(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\bigr)$. See [HV04] for more details.

This also implies a linear time algorithm for computing the minimum radius k-flat. The exact running time is $n \cdot d \cdot \exp\left(\frac{e^{O(k^2)}}{\varepsilon^{2k+1}}\log^2\frac{1}{\varepsilon}\right)$.

The constants involved were recently improved by Panigrahy [Pan04], who also simplified the analysis.

Handling multiple slabs in linear time is an open problem for further research. Furthermore, computing the best k-flat in the presence of outliers in near-linear time is also an open problem.

The L_2 measure. A natural problem is to compute the k-flat minimizing not the maximum distance, but rather the sum of squared distances; this is known as the L_2 measure, and it can be solved in $O(\min(dn^2, nd^2))$ time, using singular value decomposition [GvL96]. Recently, Rademacher *et al.* [RVW04] showed that there exists a coreset for this problem. Namely, there are $O(k^2/\varepsilon)$ points in P, such that their span contains a k-flat which is a $(1+\varepsilon)$-approximation to the best k-flat approximating the point set under the L_2 measure. Their proof is not constructive, and it would be nice to come up with a constructive and efficient algorithm for computing this coreset.

7.3 k-means and k-median clustering

Bădoiu *et al.* [BHI02] consider the problem of computing a k-median clustering of a set P of n points in $\mathbb{R}^d$. They show that for a random sample X from P of size $O(1/\varepsilon^3 \log 1/\varepsilon)$, the following two events happen with probability bounded below by a positive constant: (i) The flat $\mathrm{span}(X)$ contains a $(1+\varepsilon)$-approximate 1-median for P, and (ii) X contains a point close to the center of a 1-median of P. Thus, one can generate a small number of candidate points on $\mathrm{span}(X)$, such that one of those points is a median which is an $(1+\varepsilon)$-approximate 1-median for P.

To get k-median clustering, one needs to do this random sampling in each of the k clusters. It is unclear how to do this if those clusters are of completely different cardinality. Bădoiu *et al.* [BHI02] suggest an elaborate procedure to do so, by guessing the average radius and cardinality of the heaviest cluster, generating a candidate set for centers for this cluster using random sampling, and then recursing on the remaining points. The resulting running time is $2^{(k/\varepsilon)^{O(1)}} d^{O(1)} n \log^{O(k)} n$, and the results are correct with high-probability.

A similar procedure works for k-means, see de Vega *et al.* [dlVKKR03]. Those algorithms were recently improved to have running time with linear dependency on n, both for the case of k-median and k-means [KSS04].

7.4 Maximum margin classifier

Let P^+ and P^- be two sets of points, labeled as positive and negative, respectively. In support vector machines, one is looking for a hyperplane h such that P^+ and P^- are on different sides of h, and the minimum distance between h and the points of $P = P^+ \cup P^-$ is maximized. The distance between h and the closest point of P is known as the *margin* of h. In particular, the larger the margin is, the better generalization bounds one can prove on h. See [CS00] for more information about learning and support vector machines.

In the following, let $\Delta = \Delta(P)$ denote the diameter of P, and let ρ denote the width of the maximum width margin for P. Har-Peled and Zimak [HZ04] showed an iterative algorithm for computing a coreset for this problem. Specifically, by iteratively picking the point that has maximum violation of the current classifier to be in the coreset, they show that the algorithm terminates after $O((\Delta/\rho)^2/\varepsilon)$ iterations. Thus, there exist subsets $Q^- \subseteq P^-$ and $Q^+ \subseteq P^+$, such that the maximum margin linear classifier h for Q^+ and Q^- has a $\geq (1-\varepsilon)\rho$ margin for P. As in the case of computing the minimum enclosing ball, one calls a procedure for computing the best linear separator only on the growing coresets, which are small. Kowalczyk [Kow99] presented a similar iterative algorithm, but the size of the resulting coreset seems to be larger.

8 Conclusions

In this paper, we have surveyed several approximation algorithms for geometric problems that use the coreset paradigm. We have certainly not attempted to be comprehensive and our paper does

not reflect all the research work that can be viewed as employing this paradigm. For example, we do not touch upon the body of work on sublinear algorithms [CLM03] or on property testing in the geometric context [CS01]. Even among the results that we do cover, the choice of topics for detailed exposition is (necessarily) somewhat subjective.

Acknowledgements. We are grateful to the referees for their detailed, helpful comments.

References

[AAS01] P. K. Agarwal, B. Aronov, and M. Sharir. Exact and approximation algorithms for minimum-width cylindrical shells. *Discrete Comput. Geom.*, 26(3):307–320, 2001.

[AGHV01] P. K. Agarwal, L. J. Guibas, J. Hershberger, and E. Veach. Maintaining the extent of a moving point set. *Discrete Comput. Geom.*, 26(3):353–374, 2001.

[AHV04] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Approximating extent measures of points. *J. Assoc. Comput. Mach.*, 51(4):606–635, 2004.

[AM94] P. K. Agarwal and J. Matoušek. On range searching with semialgebraic sets. *Discrete Comput. Geom.*, 11:393–418, 1994.

[AM98] S. Arya and D. Mount. ANN: library for approximate nearest neighbor searching. `http://www.cs.umd.edu/~mount/ANN/`, 1998.

[AMN+98] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu. An optimal algorithm for approximate nearest neighbor searching in fixed dimensions. *J. Assoc. Comput. Mach.*, 45(6), 1998.

[AP02] P. K. Agarwal and C. M. Procopiuc. Exact and approximation algorithms for clustering. *Algorithmica*, 33(2):201–226, 2002.

[APV02] P. K. Agarwal, C. M. Procopiuc, and K. R. Varadarajan. Approximation algorithms for k-line center. In *Proc. 10th Annu. European Sympos. Algorithms*, pages 54–63, 2002.

[BC03a] M. Bădoiu and K. L. Clarkson. Optimal coresets for balls. `http://cm.bell-labs.com/who/clarkson/coresets2.pdf`, 2003.

[BC03b] M. Bădoiu and K. Clarkson. Smaller coresets for balls. In *Proc. 14th ACM-SIAM Sympos. Discrete Algorithms*, pages 801–802, 2003.

[BFP82] J. L. Bentley, G. M. Faust, and F. P. Preparata. Approximation algorithms for convex hulls. *Commun. ACM*, 25:64–68, 1982.

[BH01] G. Barequet and S. Har-Peled. Efficiently approximating the minimum-volume bounding box of a point set in three dimensions. *J. Algorithms*, 38:91–109, 2001.

[BHI02] M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via coresets. In *Proc. 34th Annu. ACM Sympos. Theory Comput.*, pages 250–257, 2002.

[BI76] E. M. Bronshteyn and L. D. Ivanov. The approximation of convex sets by polyhedra. *Siberian Math. J.*, 16:852–853, 1976.

[BS80] J. L. Bentley and J. B. Saxe. Decomposable searching problems i: Static-to-dynamic transformation. *J. Algorithms*, 1(4):301–358, 1980.

[Cha01] B. Chazelle. *The Discrepancy Method: Randomness and Complexity*. Cambridge University Press, New York, 2001.

[Cha02] T. M. Chan. Approximating the diameter, width, smallest enclosing cylinder and minimum-width annulus. *Internat. J. Comput. Geom. Appl.*, 12(2):67–85, 2002.

[Cha04] T. M. Chan. Faster coreset constructions and data stream algorithms in fixed dimensions. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 152–159, 2004.

[Cla93] K. L. Clarkson. Algorithms for polytope covering and approximation. In *Proc. 3th Workshop Algorithms Data Struct.*, volume 709 of *Lect. Notes in Comp. Sci.*, pages 246–252. Springer-Verlag, 1993.

[CLM03] B. Chazelle, D. Liu, and A. Magen. Sublinear geometric algorithms. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 531–540, 2003.

[CS00] N. Cristianini and J. Shaw-Taylor. *Support Vector Machines*. Cambridge Press, 2000.

[CS01] A. Czumaj and C. Sohler. Property testing with geometric queries. In *Proc. 9th Annu. European Symp. Algorithms*, pages 266–277, 2001.

[dFM01] L. da Fontana Costa and R. Marcondes Cesar, Jr. *Shape Analysis and Classification*. CRC Press, Boca Raton, 2001.

[dlVKKR03] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani. Approximation schemes for clustering problems. In *Proc. 35th Annu. ACM Sympos. Theory Comput.*, pages 50–58, 2003.

[DM98] I. Dryden and K. Mardia. *Statistical Shape Analysis*. John Wiley and Sons, San Diego, 1998.

[Dud74] R. M. Dudley. Metric entropy of some classes of sets with differentiable boundaries. *J. Approx. Theory*, 10(3):227–236, 1974.

[EHM04] J. Erickson, S. Har-Peled, and D. Mount. On the least median square problem. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 273–279, 2004.

[FG88] T. Feder and D. H. Greene. Optimal algorithms for approximate clustering. In *Proc. 20th Annu. ACM Sympos. Theory Comput.*, pages 434–444, 1988.

[Gär95] B. Gärtner. A subexponential algorithm for abstract optimization problems. *SIAM J. Comput.*, 24:1018–1035, 1995.

[GLS88] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 1988. 2nd edition 1994.

[Gon85] T. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[GvL96] G. H. Golub and C. F. van Loan. *Matrix computations*. Johns Hopkins University Press, Baltimore, MD, 3nd edition, 1996.

[Har04a] S. Har-Peled. Clustering motion. *Discrete Comput. Geom.*, 31(4):545–565, 2004.

[Har04b] S. Har-Peled. No coreset, no cry. In *Proc. 24th Conf. Found. Soft. Tech. Theoret. Comput. Sci.*, 2004. To appear.

[HG97] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. Technical report, CMU-CS, 1997. http://www.uiuc.edu/~garland/papers.html.

[HK04] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. http://www.uiuc.edu/~sariel/papers/04/small_coreset/, 2004.

[HM04] S. Har-Peled and S. Mazumdar. Coresets for k-means and k-median clustering and their applications. In *Proc. 36th Annu. ACM Sympos. Theory Comput.*, pages 291–300, 2004.

[HV02] S. Har-Peled and K. R. Varadarajan. Projective clustering in high dimensions using coresets. In *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, pages 312–318, 2002.

[HV04] S. Har-Peled and K. R. Varadarajan. High-dimensional shape fitting in linear time. *Discrete Comput. Geom.*, 32(2):269–288, 2004.

[HW87] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[HW04] S. Har-Peled and Y. Wang. Shape fitting with outliers. *SIAM J. Comput.*, 33(2):269–285, 2004.

[HZ04] S. Har-Peled and D. Zimak. Coresets for SVM. manuscript, 2004.

[Joh48] F. John. Extremum problems with inequalities as subsidary conditions. *Courant Anniversary*, pages 187–204, 1948.

[KMY03] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim. Approximate minimum enclosing balls in high dimensions using coresets. *J. Exp. Algorithmics*, 8:1.1, 2003.

[Kow99] A. Kowalczyk. Maximal margin perceptron. In A.J. Smola, P.L. Bartlett, B. Scholkopf, and D. Schuurmans, editors, *Advances in Large Margin Classiers*, pages 75–114. MIT Press, 1999.

[KSS04] A. Kumar, Y. Sabharwal, and S. Sen. A simple linear time $(1+\varepsilon)$-approximation algorithm for k-means clustering in any dimensions. In *Proc. 45th Annu. IEEE Sympos. Found. Comput. Sci.*, pages 454–462, 2004.

[KY04] P. Kumar and E.A. Yildirim. Approximating minimum volume enclosing ellipsoids using core sets. *J. Opt. Theo. Appl.*, 2004. to appear.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[Pan04] R. Panigrahy. Minimum enclosing polytope in high dimensions. menuscript, 2004.

[RVW04] L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via adaptive sampling. manuscript, 2004.

[VC71] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Probab. Appl.*, 16:264–280, 1971.

[Wes93] G. Wesolowsky. The Weber problem: History and perspective. *Location Science*, 1:5–23, 1993.

[YAPV04] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan. Practical methods for shape fitting and kinetic data structures using core sets. In *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, pages 263–272, 2004.

[ZS02] Y. Zhou and S. Suri. Algorithms for a minimum volume enclosing simplex in three dimensions. *SIAM J. Comput.*, 31(5):1339–1357, 2002.