

# Geometric Approximation via Coresets

PANKAJ K. AGARWAL, SARIEL HAR-PELED,  
AND KASTURI R. VARADARAJAN

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 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 [Chazelle 2000; Mulmuley 1993], convex approximation [Dudley 1974; Bronshteyn and Ivanov 1976], surface simplification [Heckbert and Garland 1997], feature extraction and shape descriptors [Dryden and Mardia 1998; Costa and César 2001]. For geometric problems where the input is a set of points, the question reduces to finding a small subset (a *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  $\varepsilon$ -approximation and  $\varepsilon$ -net would fail to compute a subset of points that contain a good approximation to the diameter [Vapnik and Chervonenkis 1971; Haussler and Welzl 1987]. While in this problem it is clear that convex approximation

---

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.

(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 of 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 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  $k$ -th 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 [Agarwal et al. 2001b].

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. The 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  $\gamma$  fits  $P$ , denoted as  $\mu(P, \gamma)$ , is the maximum distance between a point of  $P$  and its nearest point on  $\gamma$ , 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)<sup>1</sup> 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 [Barequet and Har-Peled 2001]. The goal is to compute an  $(1+\varepsilon)$ -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  $\mu$  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

---

<sup>1</sup>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.

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  $\gamma$  from the underlying family,  $\mu(Q, \gamma) \geq (1 - \varepsilon)\mu(P, \gamma)$ . With this property, the approach returns a shape  $\gamma^*$  that is an approximate best fit to  $P$ .

Following earlier work [Barequet and Har-Peled 2001; Chan 2002; Zhou and Suri 2002] that hinted at the generality of this approach, [Agarwal et al. 2004] provided a formal framework by introducing the notion of  $\varepsilon$ -kernel and showing that it yields a coresets 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 coresets based approach for other geometric optimization problems, including clustering and other extent-measure problems [Agarwal et al. 2002; Bădoiu and Clarkson 2003; Bădoiu et al. 2002; Har-Peled and Wang 2004; Kumar et al. 2003; Kumar and Yildirim 2005].

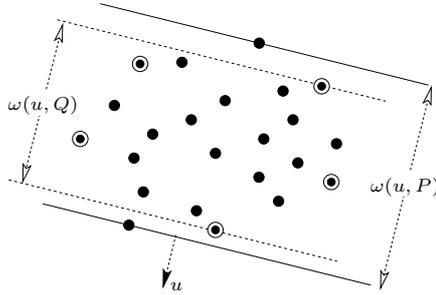
In this paper, we have attempted to review coresets based algorithms for approximating extent measure and other optimization problems. Our aim is to communicate the flavor of the techniques involved and a sense of the power of this paradigm by discussing a number of its applications. We begin in Section 2 by describing  $\varepsilon$ -kernels of point sets and algorithms for constructing them. Section 3 defines the notion of  $\varepsilon$ -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  $\varepsilon$ -kernels in the streaming model. Although  $\varepsilon$ -kernels provide coresets for a 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  $\mu$  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  $\varepsilon$ -coresets of  $P$  (with respect to  $\mu$ ) if

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

Agarwal et al. [2004] introduced the notion of  $\varepsilon$ -kernels and showed that it is an  $f(\varepsilon)$ -coresets for numerous minimization problems. We begin by defining  $\varepsilon$ -kernels and related concepts.



**Figure 1.** Directional width and  $\varepsilon$ -kernel.

**$\varepsilon$ -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  $\varepsilon$ -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. [2004] call a measure function  $\mu$  *faithful* if there exists a constant  $c$ , depending on  $\mu$ , so that for any  $P \subseteq \mathbb{R}^d$  and for any  $\varepsilon$ , an  $\varepsilon$ -kernel of  $P$  is a  $c\varepsilon$ -coreset for  $P$  with respect to  $\mu$ . Examples of faithful measures considered in that reference include diameter, width, radius of the smallest enclosing ball, and volume of the smallest enclosing box. A common property of these measures is that  $\mu(P) = \mu(\text{conv}(P))$ . We can thus compute an  $\varepsilon$ -coreset of  $P$  with respect to several measures by simply computing an  $(\varepsilon/c)$ -kernel of  $P$ .

**Algorithms for computing kernels.** An  $\varepsilon$ -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 [Bentley et al. 1982; Bronshteyn and Ivanov 1976; Dudley 1974] for a sample. For example, in the first of these articles Bentley, Faust, and Preparata 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  $\text{conv}(Q)$  is at most  $\varepsilon \text{diam}(Q)$ . 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  $\varepsilon$ -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  $\varepsilon$ -kernel of an arbitrary point set is implicit in [Barequet and Har-Peled 2001].

We call  $P$   $\alpha$ -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

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

A stronger version of the following lemma, which is very useful for constructing an  $\varepsilon$ -kernel, was proved in [Agarwal et al. 2004] by adapting a scheme from [Barequet and Har-Peled 2001]. Their scheme can be thought of as one that quickly computes an approximation to the Löwner–John Ellipsoid [John 1948].

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

The importance of Lemma 2.1 is that it allows us to adapt some classical approaches for convex hull approximation [Bentley et al. 1982; Bronshteyn and Ivanov 1976; Dudley 1974] which in fact do compute an  $\varepsilon$ -kernel when applied to fat point sets.

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

Let  $\delta$  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  $\delta$ . That is,

$$\mathbb{Z} = \{(\delta i_1, \dots, \delta i_d) \mid i_1, \dots, i_d \in \mathbb{Z}\}.$$

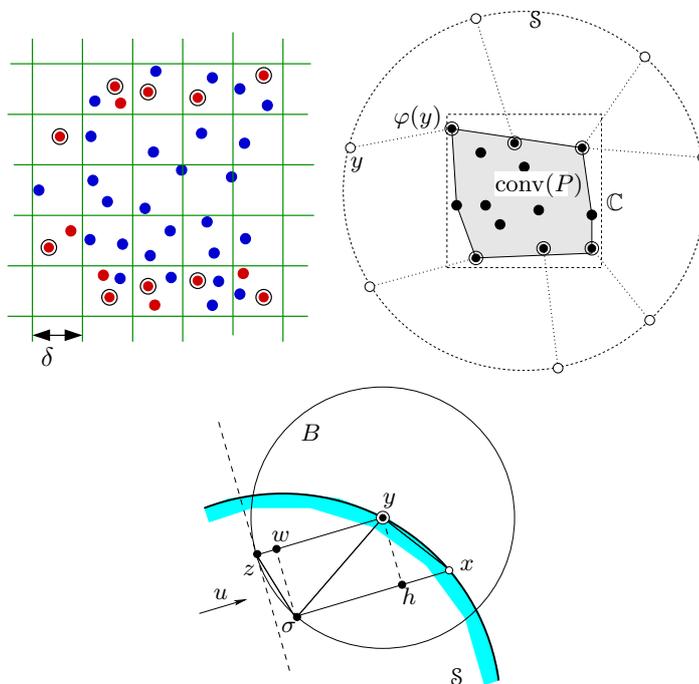
For each column along the  $x_d$ -axis in  $\mathbb{Z}$ , we choose one point from the highest nonempty cell of the column and one point from the lowest nonempty cell of the column; see Figure 2, top left. Let  $Q$  be the set of chosen points. Since  $P \subseteq [-1, +1]^d$ ,  $|Q| = O(1/(\alpha\varepsilon)^{d-1})$ . 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. [2004] showed that  $Q$  is an  $\varepsilon$ -kernel of  $P$ . Hence, we can compute an  $\varepsilon$ -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 et al. 1982].

Next we describe an improved construction, observed independently in [Chan 2004] and [Yu et al. 2004], which is a simplification of an algorithm of [Agarwal et al. 2004], which in turn is an adaptation of a method of Dudley [1974]. 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  $\mathcal{J}$  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  $y \in \mathcal{J}$  such that  $\|x - y\| \leq \delta$ . We process  $P$  into a data structure that can answer  $\varepsilon$ -approximate nearest-neighbor queries [Arya et al. 1998]. For a query point  $q$ , let  $\varphi(q)$  be the point of  $P$  returned by this data structure. For each point  $y \in \mathcal{J}$ , we compute  $\varphi(y)$  using this data structure. We return the set  $Q = \{\varphi(y) \mid y \in \mathcal{J}\}$ ; see Figure 2, top right.

We now briefly sketch, following the argument in [Yu et al. 2004], why  $Q$  is an  $\varepsilon$ -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  $\sigma$  in direction  $u$  hits  $\mathcal{S}$  at a point  $x$ . We know that there exists a point  $y \in \mathcal{J}$  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  $\partial B$  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  $\sigma x$  such that  $yh \perp \sigma x$ ; see Figure 2, bottom.



**Figure 2.** Top left: A grid based algorithm for constructing an  $\varepsilon$ -kernel. Top right: An improved algorithm. Bottom: 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  $\varepsilon$ -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 [2004] 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  $y \in \mathcal{J}$ , 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** [Chan 2004]. *Given a set  $P$  of  $n$  points in  $\mathbb{R}^d$  and a parameter  $\varepsilon > 0$ , one can compute an  $\varepsilon$ -kernel of  $P$  of size  $O(1/\varepsilon^{(d-1)/2})$  in time  $O(n + 1/\varepsilon^{d-(3/2)})$ .*

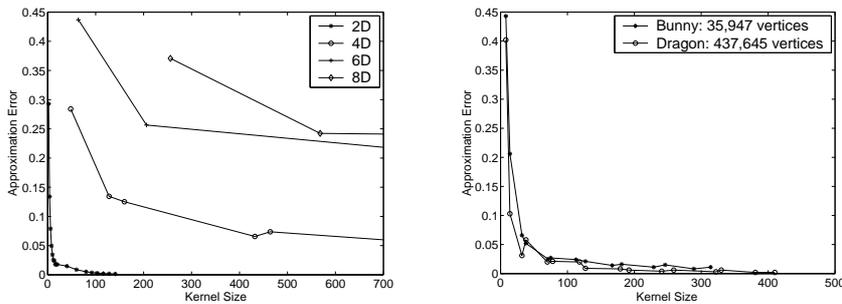
**Experimental results.** Yu et al. [2004] implemented their  $\varepsilon$ -kernel algorithm and tested its performance on a variety of inputs. They measure the quality of an  $\varepsilon$ -kernel  $Q$  of  $P$  as the maximum relative error in the directional width of  $P$  and  $Q$ . Since it is hard to compute the maximum error over all directions, they sampled a set  $\Delta$  of 1000 directions in  $\mathbb{S}^{d-1}$  and computed the maximum relative error with respect to these directions, i.e.,

$$\text{err}(Q, P) = \max_{u \in \Delta} \frac{\omega(u, P) - \omega(u, Q)}{\omega(u, P)}. \quad (2-1)$$

They implemented the constant-factor approximation algorithm of [Barequet and Har-Peled 2001] for computing the minimum-volume bounding box to convert  $P$  into an  $\alpha$ -fat set, and they used the ANN library [Arya and Mount 1998] 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  $x \in \mathcal{J}$ . Figure 3 shows how the error  $\text{err}(Q, P)$  changes as the function of kernel. These experiments show that their algorithm works extremely well in low dimensions ( $\leq 4$ ) both in terms of size and running time. See [Yu et al. 2004] for more detailed experiments.

Input Type	Input Size	$d = 2$		$d = 4$		$d = 6$		$d = 8$	
		Pre	Que	Pre	Que	Pre	Que	Pre	Que
sphere	$10^4$	0.03	0.01	0.06	0.05	0.10	9.40	0.15	52.80
	$10^5$	0.54	0.01	0.90	0.50	1.38	67.22	1.97	1393.88
	$10^6$	9.25	0.01	13.08	1.35	19.26	227.20	26.77	5944.89
cylinder	$10^4$	0.03	0.01	0.06	0.03	0.10	2.46	0.16	17.29
	$10^5$	0.60	0.01	0.91	0.34	1.39	30.03	1.94	1383.27
	$10^6$	9.93	0.01	13.09	0.31	18.94	87.29	26.12	5221.13
clustered	$10^4$	0.03	0.01	0.06	0.01	0.10	0.08	0.15	2.99
	$10^5$	0.31	0.01	0.63	0.02	1.07	1.34	1.64	18.39
	$10^6$	5.41	0.01	8.76	0.02	14.75	1.08	22.51	54.12

**Table 1.** Running time for computing  $\varepsilon$ -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.



**Figure 3.** Approximation errors under different sizes of computed  $\varepsilon$ -kernels. Left: *sphere*. Right: 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  $\mu$  that can be computed in  $O(n^\alpha)$  time, then by Theorem 2.2, we can compute a value  $\bar{\mu}$ ,  $(1-\varepsilon)\mu(P) \leq \bar{\mu} \leq \mu(P)$  by first computing an  $(\varepsilon/c)$ -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  $O(n + 1/\varepsilon^{d-(3/2)})$  [Chan 2004]. Table 2 gives running times for computing an  $(1+\varepsilon)$ -approximation of a few faithful measures.

We note that  $\varepsilon$ -kernels in fact guarantee a stronger property for several faithful measures. For instance, if  $Q$  is an  $\varepsilon$ -kernel of  $P$ , and  $C$  is some cylinder containing

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)	$n + 1/\varepsilon^3$

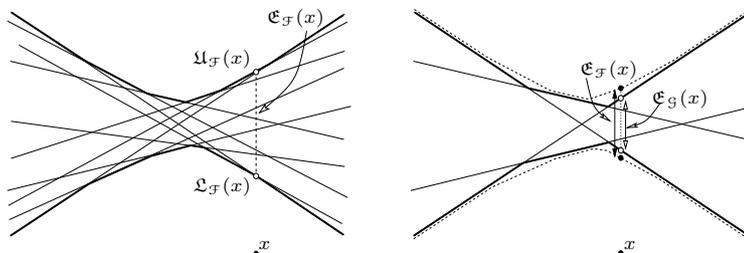
**Table 2.** Time complexity of computing  $(1 + \varepsilon)$ -approximations for certain faithful measures.

$Q$ , 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  $\varepsilon$ -kernels was introduced; see [Agarwal and Procopiuc 2002; Barequet and Har-Peled 2001; Chan 2002; Zhou and Suri 2002], for example. The framework of  $\varepsilon$ -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  $\varepsilon$ -kernel.

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

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

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

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

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

Let  $\mathcal{H} = \{h_1, \dots, h_n\}$  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  $\mathbb{R}^{d+1}$ )  $h : x_{d+1} = a_1x_1 + a_2x_2 + \dots + a_dx_d + a_{d+1}$  to the point  $h^* = (a_1, a_2, \dots, a_d, a_{d+1})$  in  $\mathbb{R}^{d+1}$ . Let  $\mathcal{H}^* = \{h^* \mid h \in \mathcal{H}\}$ . It can be proved [Agarwal et al. 2004] that  $\mathcal{K} \subseteq \mathcal{H}$  is an  $\varepsilon$ -kernel of  $\mathcal{H}$  if and only if  $\mathcal{K}^*$  is an  $\varepsilon$ -kernel of  $\mathcal{H}^*$ . Hence, by computing an  $\varepsilon$ -kernel of  $\mathcal{H}^*$  we can also compute an  $\varepsilon$ -kernel of  $\mathcal{H}$ . The following is therefore a corollary of Theorem 2.2.

**COROLLARY 3.1** [Agarwal et al. 2004; Chan 2004]. *Given a set  $\mathcal{F}$  of  $n$   $d$ -variate linear functions and a parameter  $\varepsilon > 0$ , one can compute an  $\varepsilon$ -kernel of  $\mathcal{F}$  of size  $O(1/\varepsilon^{d/2})$  in time  $O(n + 1/\varepsilon^{d-(1/2)})$ .*

We can compute  $\varepsilon$ -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  $a^1, \dots, a^n \in \mathbb{R}^p$ , 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

$$f(x, a) = \psi_0(a) + \psi_1(a)\varphi_1(x) + \dots + \psi_k(a)\varphi_k(x), \quad (3-1)$$

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

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

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

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

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*  $\varphi$ , 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

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

thus, setting

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

we get a linearization of dimension 3. Agarwal and Matoušek [1994] 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 [Agarwal et al. 2004] that a subset  $\mathcal{K} \subseteq \mathcal{H}$  is an  $\varepsilon$ -kernel if and only if the set  $\mathcal{G} = \{f_i \mid h_{a^i} \in \mathcal{K}\}$  is an  $\varepsilon$ -kernel of  $\mathcal{F}$ .

Combining the linearization technique with Corollary 3.1, one obtains:

**THEOREM 3.2** [Agarwal et al. 2004]. *Let  $\mathcal{F} = \{f_1(x), \dots, f_n(x)\}$  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  $\varepsilon$ -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  $\mathcal{F} = \{(f_1)^{1/r}, \dots, (f_n)^{1/r}\}$ , where  $r \geq 1$  is an integer and each  $f_i$  is a polynomial of some bounded degree. Agarwal et al. [2004] showed that if  $\mathcal{G}$  is an  $(\varepsilon/2(r-1))^r$ -kernel of  $\{f_1, \dots, f_n\}$ , then  $\{(f_i)^{1/r} \mid f_i \in \mathcal{G}\}$  is an  $\varepsilon$ -kernel of  $\mathcal{F}$ . Hence, we obtain the following.

**THEOREM 3.3.** *Let  $\mathcal{F} = \{(f_1)^{1/r}, \dots, (f_n)^{1/r}\}$  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  $\varepsilon$ -kernel of size  $O(1/\varepsilon^{r\sigma})$  where  $\sigma = \min\{d, k/2\}$ .*

**Applications to shape fitting problems.** Agarwal et al. [2004] 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  $P = \{p_1, \dots, p_n\}$ . 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  $\mathcal{F} = \{f_1, \dots, f_n\}$ . 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

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

Let  $\mathcal{G}$  be an  $\varepsilon$ -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 [2004] 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  $\varepsilon$ -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. [2004] 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  $P = \{p_1, \dots, p_n\}$  be a set of  $n$  points in  $\mathbb{R}^d$ , each moving independently. Let  $p_i(t) = (p_{i1}(t), \dots, p_{id}(t))$  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  $\varepsilon$ -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  $u = (u_1, \dots, u_d) \in \mathbb{S}^{d-1}$ , we define a polynomial

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

Set  $\mathcal{F} = \{f_1, \dots, f_n\}$ . 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. [2004] 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  $\varepsilon$ -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  $\varepsilon$ -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 coresets 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. [2004] 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  $\mathbb{R}^d$ . In [Bădoiu et al. 2002] a simple incremental algorithm is given for computing an  $\varepsilon$ -approximation to the minimum-enclosing ball of  $P$ . They showed, rather surprisingly, that the number of iterations of their algorithm depends only on  $\varepsilon$  and is independent of both  $d$  and  $n$ . The bound was improved by Bădoiu and Clarkson [2003; 2003] and by Kumar et al. [2003]. Kumar and Yıldırım [ $\geq$  2005] 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 in [Yu et al. 2004], 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  $P$ .

- (i) 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.
- (ii) Otherwise, let  $p \in P$  be the point farthest from  $S$ .
- (iii) Set  $Q = Q \cup \{p\}$  and go to Step 1.

It is clear that when the algorithm terminates, it does so with an  $\varepsilon$ -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  $\varepsilon$ -kernels, Yu et al. [2004] 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 coresets 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  $\varepsilon$ -kernel for a given set of points in  $\mathbb{R}^d$  can be adapted for efficiently maintaining an  $\varepsilon$ -kernel of a set of points under insertions and deletions. Here we describe the algorithm from [Agarwal et al. 2004] for maintaining  $\varepsilon$ -kernels in the streaming setting. Suppose we are receiving a stream of points  $p_1, p_2, \dots$  in  $\mathbb{R}^d$ . Given a parameter  $\varepsilon > 0$ , we wish to maintain an  $\varepsilon$ -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  $\delta$ -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  $\varepsilon$ -kernel dynamically. Besides such an algorithm, our scheme only uses abstract properties of kernels such as the following:

- (1) If  $P_2$  is an  $\varepsilon$ -kernel of  $P_1$ , and  $P_3$  is a  $\delta$ -kernel of  $P_2$ , then  $P_3$  is a  $(\delta + \varepsilon)$ -kernel of  $P_1$ ;
- (2) If  $P_2$  is an  $\varepsilon$ -kernel of  $P_1$ , and  $Q_2$  is an  $\varepsilon$ -kernel of  $Q_1$ , then  $P_2 \cup Q_2$  is an  $\varepsilon$ -kernel of  $P_1 \cup Q_1$ .<sup>2</sup>

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

---

<sup>2</sup>This property is, strictly speaking, not true for kernels. However, if we slightly modify the definition to say that  $Q \subseteq P$  is an  $\varepsilon$ -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.

We assume without loss of generality that  $1/\varepsilon$  is an integer. We use the dynamization technique of [Bentley and Saxe 1980], as follows: Let  $P = \langle p_1, \dots, p_n \rangle$  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  $P_0, P_1, \dots, P_u$ , where  $u = \lfloor \log_2 \varepsilon^k n \rfloor + 1$ , as follows.  $|P_0| = n \bmod 1/\varepsilon^k$ , and for  $1 \leq i \leq u$ ,  $|P_i| = 2^{i-1}/\varepsilon^k$  if the  $i$ -th 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  $P_0, \dots, P_u$ . 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  $\delta_i$ -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

$$\begin{aligned} 1 + \delta_i &= \prod_{l=1}^i \left(1 + \frac{\varepsilon}{cl^2}\right) \leq \exp\left(\sum_{l=1}^i \frac{\varepsilon}{cl^2}\right) \\ &= \exp\left(\frac{\varepsilon}{c} \sum_{l=1}^i \frac{1}{l^2}\right) \leq \exp\left(\frac{\pi^2 \varepsilon}{6c}\right) \leq 1 + \frac{\varepsilon}{3}, \end{aligned} \quad (5-1)$$

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  $\rho_i$ -kernel of  $K_{i+1} \cup Q_i$ . Finally,  $K_0 = Q_0 \cup K_1$ . The argument in (5-1), by the coresset 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  $\varepsilon$ -kernel of  $P$ . The size of the entire data structure is

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

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  $\delta_j$ -kernels  $Q_x, Q_y$  of rank  $j$ , for some  $j \leq \lfloor \log_2 \varepsilon^k(n+1) \rfloor + 1$ , we compute a  $\rho_{j+1}$ -kernel  $Q_z$  of  $Q_x \cup Q_y$  using algorithm **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  $\delta_{j+1}$ -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  $\xi$  is the maximum rank of a  $Q_i$  that was reconstructed, then

we recompute  $K_\xi, \dots, K_0$  in that order. That is, for  $\xi \geq i \geq 1$ , we compute a  $\rho_i$ -kernel of  $K_{i+1} \cup Q_i$  and set this to be  $K_i$ ; finally, we set  $K_0 = K_1 \cup Q_0$ .

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

$$\sum_{i=1}^{\lceil \log_2 \varepsilon^k n \rceil + 1} \frac{\varepsilon^k}{2^{i-1}} O\left(\frac{i^{2k}}{\varepsilon^k} + T_{\mathbb{A}}\left(\frac{\varepsilon}{ci^2}\right)\right) = O\left(\sum_{i=1}^{\lceil \log_2 \varepsilon^k n \rceil + 1} \frac{\varepsilon^k}{2^{i-1}} T_{\mathbb{A}}\left(\frac{\varepsilon}{ci^2}\right)\right).$$

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** [Agarwal et al. 2004]. *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  $\varepsilon$ -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  $\varepsilon$ -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  $O((\log^{2k+1} n)/\varepsilon^k + T_{\mathbb{A}}(\varepsilon/\log^2 n) \log n)$ .*

Combined with Theorem 2.2, we get a data-structure using  $(\log n/\varepsilon)^{O(d)}$  space to maintain an  $\varepsilon$ -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  $\varepsilon$ -kernel. Chan [2004] 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 in [Agarwal et al. 2001a]. 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  $\text{rad}(P)$  denote the minimum radius of all cylinders enclosing  $P$ , and let  $d(p, \ell)$  denote the distance between point  $p$  and line  $\ell$ . The following observation immediately implies an upper bound of 4 on the approximation factor of the above algorithm.

**OBSERVATION 5.2.**  $d(p, \overline{ov}) \leq 2\left(\frac{\|o-p\|}{\|o-v\|} + 1\right) \text{rad}(\{o, v, p\})$ .

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** [Chan 2004]. *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.  $w := \max\{w, \text{rad}(\{o, v, p\})\}$ .  
 2. if  $\|p\| > 2\|v\|$  then  $v := p$ .  
 3. Return  $w$ .

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  $w_f \geq \text{rad}(\{o, v_{i-1}, v_i\})$  since  $\text{rad}(\{o, v_{i-1}, v_i\})$  was one of the ‘‘candidates’’ for  $w$ . From Observation 5.2, it follows that

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

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,

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

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

## 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*),  $P_1, \dots, P_k$ , so that certain *objective* function is minimized. Given a function  $\mu$  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  $\varepsilon$ -kernel of a point set provides a coreset for several extent measures of  $P$ . However, the notion of  $\varepsilon$ -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 define 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  $\mathbb{C} = \{(f_1, P_1), \dots, (f_k, P_k)\}$  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  $\mu(\mathbb{C}) = \max_{1 \leq i \leq k} \mu(f_i, P_i)$ , and set  $r_{\text{opt}}(P, k, q) = \min_{\mathbb{C}} \mu(\mathbb{C})$ , 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  $\varepsilon$ -coreset* of  $P$  if for every  $k$ -clustering  $\mathbb{C} = \{(f_1, Q_1), \dots, (f_k, Q_k)\}$  of  $Q$ , with  $r_i = \mu(f_i, Q_i)$ ,

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

i.e., union of the expansion of each  $\mathcal{B}(f_i, r_i)$  by  $\varepsilon \mu(\mathbb{C})$  covers  $P$ . If for every  $k$ -clustering  $\mathbb{C}$  of  $Q$ , with  $r_i = \mu(f_i, Q_i)$ , we have the stronger property

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

then we call  $Q$  a *multiplicative  $\varepsilon$ -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  $\mathcal{B} = \{B_1, \dots, B_k\}$  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  $\tau$  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  $\varepsilon$ -coreset of  $P$ , as proved by Agarwal and Procopiuc [2002]. In order to construct  $\mathcal{S}$  efficiently, we use Gonzalez’s greedy algorithm [1985] to compute a factor-2 approximation of  $k$ -center, which returns a value  $\tilde{r} \leq 2r^*$ . We then draw the grid of side length  $\varepsilon \tilde{r}/4d$  and proceed as above. Using a fast implementation of Gonzalez’s algorithm as proposed in [Feder and Greene 1988; Har-Peled 2004a], one can compute an additive  $\varepsilon$ -coreset of size  $O(k/\varepsilon^d)$  in time  $O(n + k/\varepsilon^d)$ .

Agarwal et al. [2002] proved the existence of a small multiplicative  $\varepsilon$ -coreset for  $k$ -center in  $\mathbb{R}^1$ . It was subsequently extended to higher dimensions by Har-Peled [2004b]. We sketch their construction.

**THEOREM 6.1** [Agarwal et al. 2002; Har-Peled 2004b]. *Let  $P$  be a set of  $n$  points in  $\mathbb{R}^d$ , and  $0 < \varepsilon < 1/2$  a parameter. There exists a multiplicative  $\varepsilon$ -coreset of size  $O(k!/\varepsilon^{dk})$  of  $P$  for  $k$ -center.*

**PROOF.** For  $k = 1$ , by definition, an additive  $\varepsilon$ -coreset of  $P$  is also a multiplicative  $\varepsilon$ -coreset of  $P$ . For  $k > 1$ , let  $r^* = r_{\text{opt}}(P, k, 0)$  denote 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  $\mathcal{C}$  be the set of (hyper-)cubes of this grid that contain points of  $P$ . Clearly,  $|\mathcal{C}| = O(k/\varepsilon^d)$ . Let  $Q'$  be an additive  $(\varepsilon/2)$ -coreset of  $P$ . For every cell  $\Delta$  in  $\mathcal{C}$ , we inductively compute an  $\varepsilon$ -multiplicative coreset of  $P \cap \Delta$  with respect to  $(k - 1)$ -center. Let  $Q_\Delta$  be this set, and let  $\mathcal{Q} = \bigcup_{\Delta \in \mathcal{C}} Q_\Delta \cup Q'$ . 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  $\Delta$  of  $\mathcal{C}$ . Suppose  $\Delta$  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  $\Delta$ , and therefore also covers all the points of  $\Delta \cap P$ .

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

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  $\Delta$ , and let  $0 < \varepsilon \leq 1/2$  be a parameter. Har-Peled [2004a] 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  $\varepsilon$ -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 coresets for  $k$ -line-center, i.e., for the problem of covering  $P$  by  $k$  congruent cylinders of the minimum radius, was first proved in [Agarwal et al. 2002].

**THEOREM 6.2** [Agarwal et al. 2002]. *Given a set  $P$  of finite points in  $\mathbb{R}^d$  and a parameter  $0 < \varepsilon < 1/2$ , there exists an additive  $\varepsilon$ -coreset of size*

$$O((k+1)!/\varepsilon^{d-1+k})$$

*of  $P$  for the  $k$ -line-center problem.*

**PROOF.** Let  $C_{\text{opt}} = \{(\ell_1, P_1), \dots, (\ell_k, P_k)\}$  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  $\ell_1, \dots, \ell_k$  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  $\ell_i$  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  $\bar{P} = \bigcup_{\ell \in L} \bar{P}_\ell$ . It can be argued that a multiplicative  $(\varepsilon/3)$ -coreset of  $\bar{P}$  is an additive  $\varepsilon$ -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  $\varepsilon$ -coreset of  $P$  of size  $O((k+1)!/\varepsilon^{d-1+k})$ .  $\square$

Although Theorem 6.2 proves the existence of an additive coresets for  $k$ -line-center, the proof is non-constructive. However, Agarwal et al. [2002] have shown that the iterated reweighting technique of Clarkson [1993] can be used in conjunction with Theorem 6.2 to compute an  $\varepsilon$ -approximate solution to the  $k$ -line-center problem in  $O(n \log n)$  expected time, with constants depending on  $k$ ,  $\varepsilon$ , 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** [Har-Peled 2004b]. *For any  $n \geq 3$ , there exists a point set  $P = \{p_1, \dots, p_n\}$  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  $P(i) = \{p_1, \dots, p_i\}$ . 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  $p_i = (1/2^i, 2^i)$ , 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  $p_2, p_3, \dots, p_{n-1}$ .  $\square$

This construction can be embedded in  $\mathbb{R}^3$ , as described in [Har-Peled 2004b], 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 [Har-Peled 2004b]. 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.

**6.2.  $k$ -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  $w : P \rightarrow \mathbb{Z}^+$  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

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

For  $k = 1$ , this is the so-called Fermat–Weber problem [Wesolowsky 1993]. A subset  $Q \subseteq P$  with a weight function  $\chi : P \rightarrow \mathbb{Z}^+$  is called an  $\varepsilon$ -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 from [Har-Peled and Mazumdar 2004] 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  $A = \{a_1, \dots, a_m\}$  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  $\varepsilon$ -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  $\tau$  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  $\varepsilon$ -coreset of  $P$  for the  $k$ -median problem, provided  $\alpha$  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 are 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 [Har-Peled and Mazumdar 2004], 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  $\varepsilon$ -coreset of size  $O((k/\varepsilon^d) \log n)$ .

**THEOREM 6.4** [Har-Peled and Mazumdar 2004]. *Given a set  $P$  of  $n$  points in  $\mathbb{R}^d$ , and parameters  $\varepsilon > 0$  and  $k$ , 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 [2004] 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. Chen [2004] recently showed that for both  $k$ -median and  $k$ -means clustering, there are coresets whose size is  $O(dk\varepsilon^{-2} \log n)$ , which has linear dependence on  $d$ . In particular, this implies a streaming algorithm for  $k$ -means and  $k$ -median clustering using (roughly)  $O(dk\varepsilon^{-2} \log^3 n)$  space. The question of whether the dependence on  $n$  can be removed altogether is still open.

## 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  $\varepsilon$ -approximate coresets 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 coresets of the hypersphere, for the measure of width, in high dimension, would require any such cap to contain at least one point of the coresets. 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 [Gärtner 1995; Grötschel et al. 1988]. However, the natural question is whether one can compute a small coresets,  $Q \subseteq P$ , such that the minimum enclosing ball for  $Q$  is a good approximation to the real minimum enclosing ball.

Bădoiu et al. [2002] presented an algorithm, which we have already mentioned in Section 4, that generates a coresets of size  $O(1/\varepsilon^2)$ . The algorithm starts with a set  $C_0$  that contains a single (arbitrary) point of  $P$ . Next, in the  $i$ -th 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 coresets. Otherwise, take the point from  $P$  furthest from the center of the ball and add it to the coresets. The authors show that this algorithm terminates within  $O(1/\varepsilon^2)$  iterations. The bound was later improved to  $O(1/\varepsilon)$  in [Kumar et al. 2003; Bădoiu and Clarkson 2003]. 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  $O(dn/\varepsilon + 1/\varepsilon^{O(1)})$  [Bădoiu and Clarkson 2003].

It is important to note that this coresets  $Q$  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 coresets guaranteeing the stronger property is exponential in the dimension in the worst case.

**Smallest enclosing ball with outliers.** As an application of this coresets, one can compute approximately the smallest ball containing all but  $k$  of the points. Indeed, consider the smallest such ball  $b_{\text{opt}}$ , and consider  $P' = P \cap b_{\text{opt}}$ . There is a coresets  $Q \subseteq P'$  such that

- (1)  $|Q| = O(1/\varepsilon)$ , and
- (2) the smallest enclosing ball for  $Q$ , if  $\varepsilon$ -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 simultaneously  $k$  copies of the incremental algorithm for the min-enclosing ball. 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  $\varepsilon$ -approximate  $k$ -center clustering in  $k/\varepsilon$  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 [Bădoiu and Clarkson 2003] 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. Har-Peled and Varadarajan [2002] showed that there is a subset  $Q \subseteq P$  of  $(1/\varepsilon)^{O(1)}$  points such that the axis of an  $\varepsilon$ -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 in [Bădoiu et al. 2002].

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 (but constant number of) flats.

**Linear-time algorithm.** A natural approach for improving the running time of the minimum enclosing cylinder, is to adapt the general approach underlying the algorithm of [Bădoiu and Clarkson 2003] to the cylinder case. Here, the idea is that we start from a center line  $\ell_0$ . At each iteration, we find the furthest point  $p_i \in P$  from  $\ell_{i-1}$ . We then generate a line  $\ell_i$  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  $C_\varepsilon = \exp(\frac{1}{\varepsilon^3} \log^2 \frac{1}{\varepsilon})$ . See [Har-Peled and Varadarajan 2004] 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 [2004], 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 [Golub and Van Loan 1996]. Recently, Rademacher et al. [2004] showed that there exists a coresset 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 also yields a polynomial time algorithm to construct such a coresset. An interesting question is whether there is a significantly more efficient algorithm for computing a coresset. Rademacher et al. also show that their approach leads to a polynomial time approximation scheme for fitting multiple  $k$ -flats, when  $k$  and the number of flats are constants.

**7.3.  $k$ -means and  $k$ -median clustering.** Bădoiu et al. [2002] 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  $\text{span}(X)$  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  $\text{span}(X)$ , 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. [2002] 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 la Vega et al. 2003]. Those algorithms were recently improved to have running time with linear dependency on  $n$ , both for the case of  $k$ -median and  $k$ -means [Kumar et al. 2004].

**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 [Cristianini and Shaw-Taylor 2000] for more information about learning and support vector machines.

In the following, let  $\Delta = \Delta(P)$  denote the diameter of  $P$ , and let  $\rho$  denote the width of the maximum width margin for  $P$ . Har-Peled and Zimak [2004] showed an iterative algorithm for computing a coresets for this problem. Specifically, by iteratively picking the point that has maximum violation of the current classifier to be in the coresets, they show that the algorithm terminates after  $O((\Delta/\rho)^2/\epsilon)$  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  $\geq (1-\epsilon)\rho$  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 [2000] presented a similar iterative algorithm, but the size of the resulting coresets seems to be larger.

## 8. Conclusions

In this paper, we have surveyed several approximation algorithms for geometric problems that use the coresets 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 [Chazelle et al. 2003] or on property testing in the geometric context [Czumaj and Sohler 2001]. 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

- [Agarwal and Matoušek 1994] P. K. Agarwal and J. Matoušek, “On range searching with semialgebraic sets”, *Discrete Comput. Geom.* **11:4** (1994), 393–418.

- [Agarwal and Procopiuc 2002] P. K. Agarwal and C. M. Procopiuc, “Exact and approximation algorithms for clustering”, *Algorithmica* **33**:2 (2002), 201–226.
- [Agarwal et al. 2001a] P. K. Agarwal, B. Aronov, and M. Sharir, “Exact and approximation algorithms for minimum-width cylindrical shells”, *Discrete Comput. Geom.* **26**:3 (2001), 307–320.
- [Agarwal et al. 2001b] P. K. Agarwal, L. J. Guibas, J. Hershberger, and E. Veach, “Maintaining the extent of a moving point set”, *Discrete Comput. Geom.* **26**:3 (2001), 353–374.
- [Agarwal et al. 2002] P. K. Agarwal, C. M. Procopiuc, and K. R. Varadarajan, “Approximation algorithms for  $k$ -line center”, pp. 54–63 in *Algorithms—ESA 2002*, Lecture Notes in Comput. Sci. **2461**, Springer, Berlin, 2002.
- [Agarwal et al. 2004] P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan, “Approximating extent measures of points”, *J. Assoc. Comput. Mach.* **51** (2004), 606–635.
- [Arya and Mount 1998] S. Arya and D. Mount, “ANN: Library for approximate nearest neighbor searching”, 1998. Available at <http://www.cs.umd.edu/~mount/ANN/>.
- [Arya et al. 1998] 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. ACM* **45**:6 (1998), 891–923.
- [Bădoiu and Clarkson 2003] M. Bădoiu and K. L. Clarkson, “Smaller core-sets for balls”, pp. 801–802 in *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms (Baltimore, MD, 2003)*, ACM, New York, 2003.
- [Barequet and Har-Peled 2001] G. Barequet and S. Har-Peled, “Efficiently approximating the minimum-volume bounding box of a point set in three dimensions”, *J. Algorithms* **38**:1 (2001), 91–109.
- [Bentley and Saxe 1980] J. L. Bentley and J. B. Saxe, “Decomposable searching problems. I. Static-to-dynamic transformation”, *J. Algorithms* **1**:4 (1980), 301–358.
- [Bentley et al. 1982] J. L. Bentley, M. G. Faust, and F. P. Preparata, “Approximation algorithms for convex hulls”, *Comm. ACM* **25**:1 (1982), 64–68.
- [Bronshteyn and Ivanov 1976] E. M. Bronshteyn and L. D. Ivanov, “The approximation of convex sets by polyhedra”, *Siberian Math. J.* **16** (1976), 852–853.
- [Bădoiu and Clarkson 2003] M. Bădoiu and K. L. Clarkson, “Optimal core-sets for balls”, 2003. Available at <http://cm.bell-labs.com/who/clarkson/coresets2.pdf>.
- [Bădoiu et al. 2002] M. Bădoiu, S. Har-Peled, and P. Indyk, “Approximate Clustering via Core-Sets”, pp. 250–257 in *Proc. 34th Annu. ACM Sympos. Theory Comput.*, 2002. Available at <http://www.uiuc.edu/~sariel/research/papers/02/coreset/>.
- [Chan 2002] T. M. Chan, “Approximating the diameter, width, smallest enclosing cylinder, and minimum-width annulus”, *Internat. J. Comput. Geom. Appl.* **12** (2002), 67–85.
- [Chan 2004] T. M. Chan, “Faster core-set constructions and data stream algorithms in fixed dimensions”, pp. 152–159 in *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004.
- [Chazelle 2000] B. Chazelle, *The discrepancy method*, Cambridge University Press, Cambridge, 2000.

- [Chazelle et al. 2003] B. Chazelle, D. Liu, and A. Magen, “Sublinear geometric algorithms”, pp. 531–540 in *Proc. 35th ACM Symp. Theory of Comput.*, 2003.
- [Chen 2004] K. Chen, “Clustering algorithms using adaptive sampling”, 2004. Manuscript.
- [Clarkson 1993] K. L. Clarkson, “Algorithms for polytope covering and approximation”, pp. 246–252 in *Algorithms and data structures* (Montreal, PQ, 1993), Lecture Notes in Comput. Sci. **709**, Springer, Berlin, 1993.
- [Costa and César 2001] L. Costa and R. M. César, Jr., *Shape analysis and classification*, CRC Press, Boca Raton (FL), 2001.
- [Cristianini and Shaw-Taylor 2000] N. Cristianini and J. Shaw-Taylor, *Support vector machines*, Cambridge Univ. Press, New York, 2000.
- [Czumaj and Sohler 2001] A. Czumaj and C. Sohler, “Property testing with geometric queries (extended abstract)”, pp. 266–277 in *Algorithms—ESA* (Århus, 2001), Lecture Notes in Comput. Sci. **2161**, Springer, Berlin, 2001.
- [Dryden and Mardia 1998] I. L. Dryden and K. V. Mardia, *Statistical shape analysis*, Wiley Series in Probability and Statistics: Probability and Statistics, John Wiley & Sons Ltd., Chichester, 1998.
- [Dudley 1974] R. M. Dudley, “Metric entropy of some classes of sets with differentiable boundaries”, *J. Approximation Theory* **10** (1974), 227–236.
- [Erickson and Har-Peled 2004] J. Erickson and S. Har-Peled, “Optimally cutting a surface into a disk”, *Discrete Comput. Geom.* **31**:1 (2004), 37–59.
- [Feder and Greene 1988] T. Feder and D. H. Greene, “Optimal algorithms for approximate clustering”, pp. 434–444 in *Proc. 20th Annu. ACM Sympos. Theory Comput.*, 1988.
- [Gärtner 1995] B. Gärtner, “A subexponential algorithm for abstract optimization problems”, *SIAM J. Comput.* **24**:5 (1995), 1018–1035.
- [Golub and Van Loan 1996] G. H. Golub and C. F. Van Loan, *Matrix computations*, 3rd ed., Johns Hopkins University Press, Baltimore, MD, 1996.
- [Gonzalez 1985] T. F. Gonzalez, “Clustering to minimize the maximum intercluster distance”, *Theoret. Comput. Sci.* **38**:2-3 (1985), 293–306.
- [Grötschel et al. 1988] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*, Algorithms and Combinatorics **2**, Springer-Verlag, Berlin, 1988. Second edition, 1994.
- [Har-Peled 2004a] S. Har-Peled, “Clustering motion”, *Discrete Comput. Geom.* **31**:4 (2004), 545–565.
- [Har-Peled 2004b] S. Har-Peled, “No Coreset, No Cry”, in *Proc. 24th Conf. Found. Soft. Tech. Theoret. Comput. Sci.*, 2004. Available at <http://www.uiuc.edu/~sariel/papers/02/2slab/>. To appear.
- [Har-Peled and Kushal 2004] S. Har-Peled and A. Kushal, “Smaller coresets for  $k$ -median and  $k$ -means clustering”, 2004. Manuscript.
- [Har-Peled and Mazumdar 2004] S. Har-Peled and S. Mazumdar, “Coresets for  $k$ -Means and  $k$ -Median Clustering and their Applications”, pp. 291–300 in *Proc. 36th Annu. ACM Sympos. Theory Comput.*, 2004. Available at <http://www.uiuc.edu/~sariel/research/papers/03/kcoreset/>.

- [Har-Peled and Varadarajan 2002] S. Har-Peled and K. R. Varadarajan, “Projective clustering in high dimensions using core-sets”, pp. 312–318 in *Proc. 18th Annu. ACM Sympos. Comput. Geom.*, 2002. Available at <http://www.uiuc.edu/~sariel/research/papers/01/kflat/>.
- [Har-Peled and Varadarajan 2004] S. Har-Peled and K. R. Varadarajan, “High-dimensional shape fitting in linear time”, *Discrete Comput. Geom.* **32**:2 (2004), 269–288.
- [Har-Peled and Wang 2004] S. Har-Peled and Y. Wang, “Shape fitting with outliers”, *SIAM J. Comput.* **33**:2 (2004), 269–285.
- [Har-Peled and Zimak 2004] S. Har-Peled and D. Zimak, “Coresets for SVM”, 2004. Manuscript.
- [Haussler and Welzl 1987] D. Haussler and E. Welzl, “ $\epsilon$ -nets and simplex range queries”, *Discrete Comput. Geom.* **2**:2 (1987), 127–151.
- [Heckbert and Garland 1997] P. S. Heckbert and M. Garland, “Survey of polygonal surface simplification algorithms”, Technical report, CMU-CS, 1997. Available at <http://www.uiuc.edu/~garland/papers.html>.
- [John 1948] F. John, “Extremum problems with inequalities as subsidiary conditions”, pp. 187–204 in *Studies and essays presented to R. Courant on his 60th birthday, January 8, 1948*, Interscience Publishers, Inc., New York, N. Y., 1948.
- [Kowalczyk 2000] A. Kowalczyk, *Maximal margin perceptron*, edited by A. Smola et al., MIT Press, Cambridge (MA), 2000.
- [Kumar and Yildirim  $\geq$  2005] P. Kumar and E. Yildirim, “Approximating minimum volume enclosing ellipsoids using core sets”, *J. Opt. Theo. Appl.* To appear.
- [Kumar et al. 2003] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim, “Approximate minimum enclosing balls in high dimensions using core-sets”, *J. Exp. Algorithmics* **8** (2003), 1.1. Available at <http://www.compgeom.com/~piyush/meb/journal.pdf>.
- [Kumar et al. 2004] A. Kumar, Y. Sabharwal, and S. Sen, “A simple linear time  $(1 + \epsilon)$ -approximation algorithm for  $k$ -means clustering in any dimensions”, in *Proc. 45th Annu. IEEE Sympos. Found. Comput. Sci.*, 2004.
- [Mulmuley 1993] K. Mulmuley, *Computational geometry: an introduction through randomized algorithms*, Prentice Hall, Englewood Cliffs, NJ, 1993.
- [Panigrahy 2004] R. Panigrahy, “Minimum enclosing polytope in high dimensions”, 2004. Manuscript.
- [Rademacher et al. 2004] L. Rademacher, S. Vempala, and G. Wang, “Matrix approximation and projective clustering via iterative sampling”, 2004. Manuscript.
- [Vapnik and Chervonenkis 1971] V. N. Vapnik and A. Y. Chervonenkis, “On the uniform convergence of relative frequencies of events to their probabilities”, *Theory Probab. Appl.* **16** (1971), 264–280.
- [de la Vega et al. 2003] W. F. de la Vega, M. Karpinski, C. Kenyon, and Y. Rabani, “Approximation schemes for clustering problems”, pp. 50–58 in *Proc. 35th Annu. ACM Sympos. Theory Comput.*, 2003.
- [Wesolowsky 1993] G. Wesolowsky, “The Weber problem: History and perspective”, *Location Science* **1** (1993), 5–23.

- [Yu et al. 2004] H. Yu, P. K. Agarwal, R. Poreddy, and K. R. Varadarajan, “Practical methods for shape fitting and kinetic data structures using core sets”, pp. 263–272 in *Proc. 20th Annu. ACM Sympos. Comput. Geom.*, 2004.
- [Zhou and Suri 2002] Y. Zhou and S. Suri, “Algorithms for a minimum volume enclosing simplex in three dimensions”, *SIAM J. Comput.* **31**:5 (2002), 1339–1357.

PANKAJ K. AGARWAL  
DEPARTMENT OF COMPUTER SCIENCE  
BOX 90129  
DUKE UNIVERSITY  
DURHAM NC 27708-0129  
pankaj@cs.duke.edu

SARIEL HAR-PELED  
DEPARTMENT OF COMPUTER SCIENCE  
DCL 2111  
UNIVERSITY OF ILLINOIS  
1304 WEST SPRINGFIELD AVE.  
URBANA, IL 61801  
sariel@uiuc.edu

KASTURI R. VARADARAJAN  
DEPARTMENT OF COMPUTER SCIENCE  
THE UNIVERSITY OF IOWA  
IOWA CITY, IA 52242-1419  
varadar@cs.uiowa.edu