Stability of ε-Kernels

Pankaj K. Agarwal, Jeff M. Phillips, and Hai Yu

Duke University, University of Utah, and Google

Abstract. Given a set P of n points in $\mathbb{R}^d$, an ε-kernel $K \subseteq P$ approximates the directional width of P in every direction within a relative $(1-\varepsilon)$ factor. In this paper we study the stability of ε-kernels under dynamic insertion and deletion of points to P and by changing the approximation factor ε. In the first case, we say an algorithm for dynamically maintaining a ε-kernel is stable if at most $O(1)$ points change in K as one point is inserted or deleted from P. We describe an algorithm to maintain an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ in $O(1/\varepsilon^{(d-1)/2} + \log n)$ time per update. Not only does our algorithm maintain a stable ε-kernel, its update time is faster than any known algorithm that maintains an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$. Next, we show that if there is an ε-kernel of P of size κ, which may be dramatically less than $O(1/\varepsilon^{(d-1)/2})$, then there is an $(\varepsilon/2)$-kernel of P of size $O(\min\{1/\varepsilon^{(d-1)/2}, \kappa^{\lfloor d/2 \rfloor} \log^{d-2}(1/\varepsilon)\})$. Moreover, there exists a point set P in $\mathbb{R}^d$ and a parameter $\varepsilon > 0$ such that if every ε-kernel of P has size at least κ, then any $(\varepsilon/2)$-kernel of P has size $\Omega(\kappa^{\lfloor d/2 \rfloor})$. [1]

1 Introduction

With recent advances in sensing technology, massive geospatial data sets are being acquired at an unprecedented rate in many application areas, including GIS, sensor networks, robotics, and spatial databases. Realizing the full potential of these data sets requires developing scalable algorithms for analyzing and querying them. Among many interesting algorithmic developments to meet this challenge, there is an extensive amount of work on computing a "small summary" of large data sets that preserves certain desired properties of the input data and on obtaining a good trade-off between the quality of the summary and its size. A coreset is one example of such approximate summaries. Specifically, for an input set P and a function f, a *coreset* $C \subseteq P$ is a subset of P (with respect to f) with the property that $f(C)$ approximates $f(P)$. If a small-size coreset C can be computed quickly (much faster than computing $f(P)$), then one can compute an approximate value of $f(P)$ by first computing C and then computing $f(C)$. This coreset-based approach has been successfully used in a wide range of geometric optimization problems over the last decade; see [2].

[1] Research supported by subaward CIF-32 from NSF grant 0937060 to CRA, by NSF under grants CNS-05-40347, CFF-06-35000, and DEB-04-25465, by ARO grants W911NF-04-1-0278 and W911NF-07-1-0376, by an NIH grant 1P50-GM-08183-01, by a DOE grant OEG-P200A070505, and by a grant from the U.S.–Israel Binational Science Foundation.

ε-kernels. Agarwal *et al.* [1] introduced the notion of ε-kernels and proved that it is a coreset for many functions. For any direction $u \in \mathbb{S}^{d-1}$, let $P[u] = \arg\max_{p \in P} \langle p, u \rangle$ be the extreme point in P along u; $\omega(P, u) = \langle P[u] - P[-u], u \rangle$ is called the *directional width* of P in direction u. For a given $\varepsilon > 0$, $K \subset P \subset \mathbb{R}^d$ is called an *ε-kernel* of P if

$$\langle P[u] - K[u], u \rangle \leq \varepsilon\omega(P, u)$$

for all directions $u \in \mathbb{S}^{d-1}$.[2] For simplicity, we assume $\varepsilon \in (0, 1)$, because for $\varepsilon \geq 1$, one can choose a constant number of points to form an ε-kernel. By definition, if X is an ε-kernel of P and K is a δ-kernel of X, then K is a $(\delta + \varepsilon)$-kernel of P.

Agarwal et al. [1] showed that there exists an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ and it can be computed in time $O(n + 1/\varepsilon^{3d/2})$, when d is fixed (assumed throughout the paper). The running time was improved by Chan [6] to $O(n + 1/\varepsilon^{d-3/2})$ (see also [10]). In a number of applications, the input point set is being updated periodically, so algorithms have also been developed to maintain ε-kernels dynamically. Agarwal *et al.* [1] had described a data structure to maintain an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ in $(\log(n)/\varepsilon)^{O(d)}$ time per update. The update time was recently improved by Chan [7] to $O((1/\varepsilon^{(d-1)/2}) \log n + 1/\varepsilon^{d-3/2})$. His approach can also maintain an ε-kernel of size $O((1/\varepsilon^d) \log n)$ with update time $O(\log n)$. If only insertions are allowed (e.g. in a streaming model), the size of the data structure can be improved to $O(1/\varepsilon^{(d-1)/2})$ [4, 11].

In this paper we study two problems related to the *stability* of ε-kernels: how ε-kernels change as we update the input set or vary the value of ε.

Dynamic stability. Since the aforementioned dynamic algorithms for maintaining an ε-kernel focus on minimizing the size of the kernel, changing a single point in the input set P may drastically change the resulting kernel. This is particularly undesirable when the resulting kernel is used to build a dynamic data structure for maintaining another information. For example, kinetic data structures (KDS) based on coresets have been proposed to maintain various extent measures of a set of moving points [2]. If an insertion or deletion of an object changes the entire summary, then one has to reconstruct the entire KDS instead of locally updating it. In fact, many other dynamic data structures for maintaining geometric summaries also suffer from this undesirable property [9].

We call an ε-kernel *s-stable* if the insertion or deletion of a point causes the ε-kernel to change by at most s points. For brevity, if $s = O(1)$, we call the ε-kernel to be *stable*. Chan's dynamic algorithm can be adapted to maintain a stable ε-kernel of size $O((1/\varepsilon^{d-1}) \log n)$; see Lemma 1 below. An interesting question is whether there is an efficient algorithm for maintaining a stable ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$, as points are being inserted or deleted. Maintaining a stable ε-kernel dynamically is difficult for two main reasons. First, for an input set P,

[2] This is a slightly stronger version of the definition than defined in [1] and an ε-kernel K gives a relative $(1 + 2\varepsilon)$-approximation of $\omega(P, u)$ for all $u \in \mathbb{S}^{d-1}$ (i.e. $\omega(K, u) \leq \omega(P, u) \leq (1 + 2\varepsilon)\omega(K, u)$).

many algorithms compute ε-kernels in two or more steps. They first construct a large ε-kernel K' (e.g. see [1, 7]), and then use a more expensive algorithm to create a small ε-kernel of K'. However, if the first algorithm is unstable, then K' may change completely each time P is updated. Second, all of the known ε-kernel algorithms rely on first finding a "rough shape" of the input set P (e.g., finding a small box that contains P), estimating its fatness [5]. This rough approximation is used crucially in the computation of the ε-kernel. However, this shape is itself very unstable under insertions or deletions to P. Overcoming these difficulties, we prove the following in Section 2:

Theorem 1. *Given a parameter $0 \le \varepsilon \le 1$, a stable ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ of a set of n points in $\mathbb{R}^d$ can be maintained under insertions and deletions in $O(1/\varepsilon^{(d-1)/2} + \log n)$ amortized time.*

Note that the update time of maintaining an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$ is better than that in [7].

Approximation stability. If the size of an ε-kernel K is $O(1/\varepsilon^{(d-1)/2})$, then decreasing ε changes K quite predictably. However, this is the worst-case bound, and it is possible that the size of K may be quite small, e.g., $O(1)$, or in general much smaller than the $1/\varepsilon^{(d-1)/2}$ maximum (efficient algorithms are known for computing ε-kernels of near-optimal size [2]). Then how much can the size increase as we reduce the allowable error from ε to $\varepsilon/2$? For any $\varepsilon > 0$, let $\kappa(P, \varepsilon)$ denote the minimum size of an ε-kernel of P. Unlike many shape simplification problems, in which the size of simplification can change drastically as we reduce the value of ε, we show (Section 3) that this does not happen for ε-kernels and that $\kappa(P, \varepsilon/2)$ can be expressed in terms of $\kappa(P, \varepsilon)$.

Theorem 2. *For any point set P and for any $\varepsilon > 0$,*

$$\kappa(P, \varepsilon/2) = O(\min\{\kappa(P, \varepsilon)^{\lfloor d/2 \rfloor} \log^{d-2}(1/\varepsilon), 1/\varepsilon^{(d-1)/2}\}).$$

Moreover, there exist a point set P and some $\varepsilon > 0$ such that $\kappa(P, \varepsilon/2) = \Omega(\kappa(P, \varepsilon)^{\lfloor d/2 \rfloor})$.

2 Dynamic Stability

In this section we describe an algorithm that proves Theorem 1. The algorithm is composed of a sequence of modules, each with certain property. We first state that Chan's dynamic coreset algorithm [7] can be made stable (see proof in full version [3]):

Lemma 1. *For any $0 < \varepsilon < 1$, an ε-kernel K of P of size $O((1/\varepsilon^{d-1}) \log n)$ can be maintained in $O(\log n)$ time with $O(1)$ changes to K per update.*

We first define the notion of anchor points and fatness of a point set and describe two algorithms for maintaining stable ε-kernels with respect to a fixed anchor: one of them maintains a kernel of size $O(1/\varepsilon^{d-1})$ and the other of size

$O(1/\varepsilon^{(d-1)/2})$; the former has smaller update time. Then we describe the algorithm for updating anchor points and maintaining a stable kernel as the anchors change. Finally, we put these modules together to obtain the final algorithm. We make the following simple observation, which will be crucial for combining different modules.

Lemma 2 (Composition Lemma). *If K is an s-stable ε-kernel of P and K' is an s'-stable ε'-kernel of K, then K' is an $(s \cdot s')$-stable $(\varepsilon + \varepsilon')$-kernel of P.*

Anchors and fatness of a point set. We call a point set P *β-fat* if

$$\max_{u,v \in \mathbb{S}^{d-1}} \omega(P,u)/\omega(P,v) \le \beta.$$

If β is a constant, we sometimes just say that P is *fat*. An arbitrary point set P can be made fat by applying an affine transform: we first choose a set of $d+1$ *anchor points* $A = \{a_0, a_1, \ldots, a_d\}$ using the following procedure of Barequet and Har-Peled [5]. Choose a_0 arbitrarily. Let a_1 be the farthest point from a_0. Then inductively, let a_i be the farthest point from the flat $\mathrm{span}(a_0, \ldots, a_{i-1})$. (See Figure 1.) The anchor points A define a bounding box I_A with center at a_0 and orthogonal directions defined by vectors from the flat $\mathrm{span}(a_0, \ldots, a_{i-1})$ to a_i. The extents of I_A in each orthogonal direction is defined by placing each a_i on a bounding face and extending I_A the same distance from a_0 in the opposite direction. Next we perform an affine transform T_A on P such that the vector from the flat $\mathrm{span}(a_0, \ldots, a_{i-1})$ to a_i is equal to e_i, where $e_0 = (0, \ldots, 0), e_1 = (1, 0, \ldots, 0), \ldots, e_d = (0, \ldots, 0, 1)$. This ensures that $T_A(P) \subseteq T_A(I_A) = [-1,1]^d$. The next lemma shows that $T_A(P)$ is fat, and follows easily from [8].

Lemma 3. *For all $u \in \mathbb{S}^{d-1}$ and for $\beta_d \le 2^d d^{5/2} d!$,*

$$\omega(T_A(A), u) \le \omega(T_A(P), u) \le \omega(T_A(I_A), u) \le \beta_d \cdot \omega(T_A(A), u). \qquad (1)$$

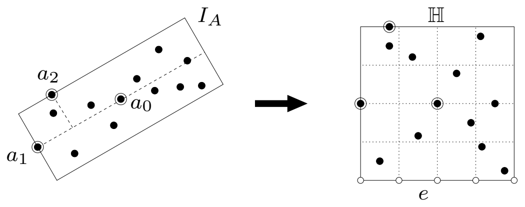


Fig. 1. Anchor points $A = \{a_0, a_1, a_2\}$, rectangle I_A, and transform T_A applied to P; square $\mathbb{H}$, two-dimensional grid $\mathbb{G}$, and one-dimensional grid $\mathbb{G}_e$ on the edge e of $\mathbb{H}$.

Agarwal *et al.* [1] show if K is an ε-kernel of P, then $T(K)$ is an ε-kernel of $T(P)$ for any affine transform T, which implies that one can compute an ε-kernel of $T(P)$. We will need the following generalization of the definition of ε-kernel. For two points sets P and Q, a subset $K \subseteq P$ is called an *ε-kernel of P with respect to Q* if $\langle P[u] - K[u], u \rangle \le \varepsilon \omega(Q,u)$ for all $u \in \mathbb{S}^{d-1}$.

Stable ε-kernels for a fixed anchor. Let A be a set of anchor points of P, as described above. We describe algorithms for maintaining stable ε-kernels (with respect to A) under the assumption that A remains a set of anchor points of P, i.e., $A \subseteq P \subset I_A$, as P is being updated by inserting and deleting points. In view of the above discussion, without loss of generality, we assume $I_A = [-1, +1]^d$ and denote it by $\mathbb{H}$. As for the static case [1, 6], we first describe a simpler algorithm that maintains a stable ε-kernel of size $O(1/\varepsilon^{d-1})$, and then a more involved one that maintains a stable ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$.

Set $\delta = \varepsilon/\sqrt{d}$ and draw a d-dimensional grid $\mathbb{G}$ inside $\mathbb{H}$ of size δ, i.e., the side-length of each grid cell is at most δ; $\mathbb{G}$ has $O(1/\delta^d)$ cells. For each grid cell τ, let $P_\tau = P \cap \tau$. For a point $x \in \mathbb{H}$ lying in a grid cell τ, let $\hat{x}$ be the vertex of τ nearest to the origin; we can view x being *snapped* to the vertex $\hat{x}$. For each facet f of $\mathbb{H}$, $\mathbb{G}$ induces a $(d-1)$-dimensional grid $\mathbb{G}_f$ on f; $\mathbb{G}$ contains a *column* of cells for each cell in $\mathbb{G}_f$. For each cell $\Delta \in \mathbb{G}_f$, we choose (at most) one point of P as follows: let τ be the nonempty grid cell in the column of $\mathbb{G}$ corresponding to Δ that is closest to f. We choose an arbitrary point from P_τ; if there is no nonempty cell in the column, no point is chosen. Let L_f be the set of chosen points. Set $\mathcal{L} = \bigcup_{f \in \mathbb{H}} L_f$. Agarwal *et al.* [1] proved that $\mathcal{L}$ is an ε-kernel of P. Insertion or deletion of a point in P affects at most one point in L_f, and it can be updated in $O(\log(1/\varepsilon))$ time. Hence, we obtain the following:

Lemma 4. *Let P be a set of n points in $\mathbb{R}^d$, let $A \subseteq P$ be a set of anchor points of P, and let $0 < \varepsilon < 1$ be a parameter. P can be preprocessed in $O(n + 1/\varepsilon^{d-1})$ time, so that a (2d)-stable ε-kernel of P with respect to A of size $O(1/\varepsilon^{d-1})$ can be maintained in $O(\log 1/\varepsilon)$ time per update provided that A remains an anchor set of P.*

Agarwal *et al.* [1] and Chan [6] have described algorithms for computing an ε-kernel of size $O(1/\varepsilon^{(d-1)/2})$. We adapt Chan's algorithm to maintain a stable ε-kernel with respect to a fixed anchor A. We begin by mentioning a result of Chan that lies at the heart of his algorithm.

Lemma 5 (Chan [6]). *Let $E \in \mathbb{N}$, $E^\tau \le F \le E$ for some $0 < \tau < 1$, and $P \subseteq [0:E]^{d-1} \times \mathbb{R}$ a set of at most n points. For all grid points $b \in [0:F]^{d-1} \times \mathbb{R}$, the nearest neighbors of each b in P can be computed in time $O(n + E^{d-2}F)$.*

We now set $\gamma = \sqrt{\varepsilon}/c$ for a constant $c > 1$ to be used in a much sparser grid than with δ. Let $\mathbb{C} = [-2, +2]^d$ and f be a facet of $\mathbb{C}$. We draw a $(d-1)$-dimensional grid on f of size γ. Assuming f lies on the plane $x_d = -2$, we choose a set $B_f = \{(i_1\gamma, \ldots, i_{d-1}\gamma, -2) \in \mathbb{Z}^d \mid -\lceil 2/\gamma \rceil \le i_1, \ldots, i_{d-1} \le \lceil 2/\gamma \rceil\}$ of grid points. For a subset $X \subseteq P$ and a point b, we define $\psi(X, b) = \arg\min_{x \in X} \|\hat{x} - b\|$, i.e., the point in X such that the snapped point is nearest to b. For a set R, $\psi(X, R) = \{\psi(X, r) \mid r \in R\}$. There is a one to one mapping between the faces of $\mathbb{C}$ and $\mathbb{H}$, so we also use f to denote the corresponding facet of $\mathbb{H}$. Let L_f be the set of points chosen in the previous algorithm corresponding to facet f of $\mathbb{H}$ for computing an $(\varepsilon/2)$-kernel of P. Set $G_f = \psi(L_f, B_f)$. Chan showed that $\mathcal{G} = \bigcup_{f \in \mathbb{C}} G_f$ is an $(\varepsilon/2)$-kernel of $\mathcal{L}$ and thus an ε-kernel of P. Scaling $\mathbb{G}$ and

B_f appropriately and using Lemma 5, G_f can be computed in $O(n + 1/\varepsilon^{d-3/2})$ time. Hence, $\mathcal{G}$ can be computed in $O(n + 1/\varepsilon^{d-3/2})$ time.

Note that $\psi(L_f, b)$ can be the same for many points $b \in B_f$, so insertion or deletion of a point in P (and thus in L_f) may change G_f significantly, thereby making $\mathcal{G}$ unstable. We circumvent this problem by introducing two new ideas. First, $\psi(L_f, B_f)$ is computed in two stages, and second it is computed in an iterative manner. We describe the construction and the update algorithm for f; the same algorithm is repeated for all facets.

We partition $\mathbb{H}$ into $O(1/\gamma^{d-1})$ boxes: for $J = \langle i_1 \ldots, i_{d-1}\rangle \in [-1/\gamma, 1/\gamma]^{d-1} \cap \mathbb{Z}^{d-1}$, we define $\mathbb{H}_J = [i_1\gamma, (i_1 + 1)\gamma] \times \cdots \times [i_{d-1}\gamma, (i_{d-1} + 1)\gamma] \times [-1, +1]$. We maintain a subset $X \subseteq L_f$. Initially, we set $X = L_f$. Set $X_J = X \cap \mathbb{H}_J$. We define a total order on the points of B_f. Initially, we sort B_f in lexicographic order, but the ordering will change as insertions and deletions are performed on P. Let $\langle b_1, \ldots, b_u\rangle$ be the current ordering of B_f. We define a map $\varphi : B_f \to L_f$ as follows. Suppose $\varphi(b_1), \ldots, \varphi(b_{i-1})$ have been defined. Let $J_i = \arg\min_J \|\hat{\psi}(X_J, b_i) - b_i\|$; here $\hat{\psi}(\cdot)$ denotes the snapped point of $\psi(\cdot)$. We set $\varphi(b_i) = \psi(X_{J_i}, b_i)$. We delete $\varphi(b_i)$ from X (and from X_{J_i}) and recompute $\hat{\psi}(X_{J_i}, B_f)$. Set $K_f = \{\varphi(b) \mid b \in B_f\}$ and $K = \bigcup_f K_f$. Computing J_i and $\varphi(b_i)$ takes $O(1/\varepsilon^{(d-1)/2})$ time, and, by Lemma 5, $\psi(X_{J_i}, B_f)$ can be computed in $O(|X_J| + 1/\gamma^{d-2} \cdot 1/\gamma) = O(1/\varepsilon^{(d-1)/2})$ time.

It can be proved that the map φ and the set K_f satisfy the following properties:

(P1) $\varphi(b_i) \neq \varphi(b_j)$ for $i \neq j$,
(P2) $\varphi(b_i) = \psi(L_f \setminus \{\varphi(b_j) \mid j < i\}, b_i)$,
(P3) $K_f \supseteq \psi(L_f, B_f)$.

Indeed, (P1) and (P2) follow from the construction, and (P3) follows from (P2). (P3) immediately implies that K is an ε-kernel of P. Next, we describe the procedures for updating K_f when L_f changes. These procedures maintain (P1)–(P3), thereby ensuring that the algorithm maintains an ε-kernel.

Inserting a point. Suppose a point p is inserted into L_f. We add p to X. Suppose $p \in \mathbb{H}_J$. We recompute $\psi(X_J, B_f)$. Next, we update $\varphi(\cdot)$ and K as follows. We maintain a point $\xi \in L_f$. Initially, ξ is set to p. Suppose we have processed $b_1, \ldots, b_{i-1}$. Let $\eta \in L_f$ be the current $\varphi(b_i)$. If $\|\hat{\xi} - b_i\| \leq \|\hat{\eta} - b_i\|$, then we swap ξ and $\varphi(b_i)$, otherwise neither ξ nor $\varphi(b_i)$ is updated. We then process b_{i+1}. After processing all points of B_f if $\xi = p$, i.e., no $\varphi(b_i)$ is updated, we stop. Otherwise, we add p to K_f and delete ξ from K_f. The insertion procedure makes at most two changes in K_f, and it can be verified that (P1)-(P3) are maintained.

Deleting a point. Suppose p is deleted from L_f. Suppose $p \in \mathbb{H}_J$. If $p \notin K_f$, then $p \in X$. We delete p from X and X_J and recompute $\psi(X_J, B)$. If $p \in K_f$, i.e., there is a $b_i \in B$ with $p = \varphi(b_i)$, then $p \notin X$. We delete p from K_f and K, recompute $\varphi(b_i)$, and add the new $\varphi(b_i)$ to K_f. Let $\varphi(b_i) \in \mathbb{H}_J$; we remove $\varphi(b_i)$ from X_J and recompute $\psi(X_J, B_f)$. We modify the ordering of B_f by moving b_i from its current position to the end. This is the only place where the ordering of B_f is modified. Since b_i is now the last point in the ordering of B_f, the new

$\varphi(b_i)$ does not affect any other $\varphi(b_j)$. The deletion procedure also makes at most two changes in K_f and maintains (P1)–(P3).

Finally, insertion or deletion of a point in P causes at most one insertion plus one deletion in L_f, therefore we can conclude the following:

Lemma 6. *Let P be a set of n points in $\mathbb{R}^d$, A a set of anchor points of P, and $0 < \varepsilon < 1$ a parameter. P can be preprocessed in $O(n + 1/\varepsilon^{d-1})$ time into a data structure so that a stable ε-kernel of P with respect to A of size $O(1/\varepsilon^{(d-1)/2})$ can be maintained in $O(1/\varepsilon^{(d-1)/2})$ time under insertion and deletion, provided that A remains an anchor set of P.*

Updating anchors. We now describe the algorithm for maintaining a stable ε-kernel when anchors of P are no longer fixed and need to be updated dynamically. Roughly speaking, we divide P into *inner* and *outer* subsets of points. The outer subset acts as a *shield* so that a stable kernel of the inner subset with respect to a fixed anchor can be maintained using Lemma 4 or 6. When the outer subset can no longer act as a shield, we reconstruct the inner and outer sets and start the algorithm again. We refer to the duration between two consecutive reconstruction steps as an *epoch.* The algorithm maintains a stable kernel within each epoch, and the amortized number of changes in the kernel because of reconstruction at the beginning of a new epoch will be $O(1)$. We can use a de-amortization technique to make the ε-kernel stable across epochs. We now describe the algorithm in detail.

In the beginning of each epoch, we perform the following preprocessing. Set $\alpha = 1/10$ and compute a α-kernel $\mathcal{L}$ of P of size $O(\log n)$ using Chan's dynamic algorithm; we do not need the stable version of his algorithm advertised above. $\mathcal{L}$ can be updated in $O(\log n)$ time per insertion/deletion. We choose a parameter m, which is set to $1/\varepsilon^{d-1}$ or $1/\varepsilon^{(d-1)/2}$. We create the outer subset of P by peeling off m "layers" of anchor points $A_1, \ldots, A_m$. Initially, we set $P_0 = P$. Suppose we have constructed $A_0, \ldots, A_{i-1}$. Set $P_{i-1} = P \setminus \bigcup_{j=1}^{i-1} A_j$, and $\mathcal{L}$ is an α-kernel of P_{i-1}. Next, we construct the anchor set A_i of $\mathcal{L}$ as described earlier in this section. We set $P_i = P_{i-1} \setminus A_i$ and update $\mathcal{L}$ so that it is an α-kernel of P_i. Let $\mathcal{A} = \bigcup_i A_i$, $A = A_m$, and $P_I = P \setminus \mathcal{A}$. Let $\mathbb{H} = (1+\alpha)I_A$. By construction $P_I \subset \mathbb{H}$. $\mathcal{A}$ forms the outer subset and acts as a shield for P_I, which is the inner subset. Set $\delta = \varepsilon/(2(1+\alpha)(\beta_d)^2)$, where β_d is the constant in Lemma 3.

If $m = 1/\varepsilon^{d-1}$ (resp. $1/\varepsilon^{(d-1)/2}$), we maintain a stable δ-kernel K_I of P_I with respect to A of size $O(m)$ using Lemma 4 (resp. Lemma 6). Set $K = K_I \cup \mathcal{A}$; $|K| = O(m)$. We prove below that K is an ε-kernel of P. Let p be a point that is inserted into or deleted from P. If $p \in \mathbb{H}$, then we update K_I using Lemma 4 or 6. On the other hand, if p lies outside $\mathbb{H}$, we insert it into or delete it from $\mathcal{A}$. Once $\mathcal{A}$ has been updated m times, we end the current epoch and discard the current K. We begin a new epoch and reconstruct $\mathcal{A}$, P_I, and K_I as described above.

The preprocessing step at the beginning of a new epoch causes $O(m)$ changes in K and there are at least m updates in each epoch, therefore the algorithm maintains a stable kernel in the amortized sense. Again, using a de-amortization technique, we can ensure that K is stable. The correctness of the algorithm follows from the following lemma (proved in full version [3]).

Lemma 7. *K is always an ε-kernel of P.*

Using Lemmas 4 and 6, we can bound the amortized update time and conclude the following.

Lemma 8. *For a set P of n points in $\mathbb{R}^d$ and a parameter $0 < \varepsilon < 1$, there is a data structure that can maintain a stable ε-kernel of P of size $O(1/\varepsilon^{(d-1)/2})$ under insertions and deletions in amortized time $O(n\varepsilon^{(d-1)/2} + 1/\varepsilon^{(d-1)/2} + \log n)$, or of size $O(1/\varepsilon^{d-1})$ in amortized time $O(n\varepsilon^{d-1} + \log n + \log(1/\varepsilon))$.*

Putting it together. For a point set $P \subset \mathbb{R}^d$ of size n, we can produce the best size and update time tradeoff for stable ε-kernels by invoking Lemma 2 to compose three stable ε-kernel algorithms, as illustrated in Figure 2. We first apply Lemma 1 to maintain a stable $(\varepsilon/3)$-kernel K_1 of P of size $O(\min\{n, (1/\varepsilon^{d-1}) \cdot \log n\})$ with update time $O(\log n)$. We then apply Lemma 8 to maintain a stable $(\varepsilon/3)$-kernel K_2 of K_1 of size $O(1/\varepsilon^{d-1})$ with update time $O(|K_1|\varepsilon^{d-1} + \log|K_1| + \log(1/\varepsilon)) = O(\log n + \log(1/\varepsilon))$. Finally we apply Lemma 8 again to maintain a stable $(\varepsilon/3)$-kernel K of K_2 of size $O(1/\varepsilon^{(d-1)/2})$ with update time $O(|K_2|\varepsilon^{(d-1)/2} + 1/\varepsilon^{(d-1)/2} + \log|K_2|) = O(1/\varepsilon^{(d-1)/2})$. K is a stable ε-kernel of P of size $O(1/\varepsilon^{(d-1)/2})$ with update time $O(\log n + 1/\varepsilon^{(d-1)/2})$. This completes the proof of Theorem 1.

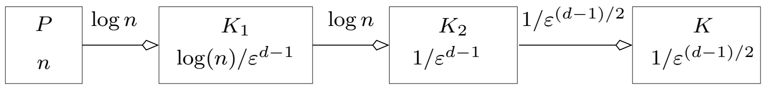

P
n
$\log n$
K_1
$\log(n)/\varepsilon^{d-1}$
$\log n$
K_2
$1/\varepsilon^{d-1}$
$1/\varepsilon^{(d-1)/2}$
K
$1/\varepsilon^{(d-1)/2}$

Fig. 2. Composing stable ε-kernel algorithms.

3 Approximation Stability

In this section we prove the upper bound in Theorem 2. Due to lack of space we only prove the upper bound for $d = 2, 3$; the remainder is in the full version [3].

By [1], it suffices to consider the case in which P is fat and the diameter of P is normalized to 1. Let K be an ε-kernel of P of the smallest size. Let $\mathcal{P} = \text{conv}(K)$, and $\mathcal{P}_\varepsilon = \mathcal{P} \oplus \varepsilon\mathbb{B}^d$. We have $\mathcal{P} \subseteq \text{conv}(P) \subseteq \mathcal{P}_\varepsilon$ by the definition of ε-kernels. It suffices to show that there is a set $K' \subseteq P$ such that for $\mathcal{P}' = \text{conv}(K')$, $\mathcal{P}' \subseteq \text{conv}(P) \subseteq \mathcal{P}'_{\varepsilon/2}$, and $|K'| = O(|K|^{\lfloor d/2 \rfloor} \log^{d-2}(1/\varepsilon))$ [1].

For convenience, we assume that K' is not necessarily a subset of points in P; instead, we only require K' to be a subset of points in $\text{conv}(P)$. By Caratheodory's theorem, for each point $x \in K$, we can choose a set $P_x \subseteq P$ of at most $d+1$ points such that $x \in \text{conv}(P_x)$. We set $\bigcup_{x \in K'} P_x$ as the desired $(\varepsilon/2)$-kernel of P; $|\bigcup_{x \in K'} P_x| \leq (d+1)|K'| = O(\kappa(P,\varepsilon)^{\lfloor d/2 \rfloor} \log^{d-2}(1/\varepsilon))$.

Initially, we add a point into K' for each point in K. If $p \in K$ lies on $\partial \,\text{conv}(P)$, we add p to K'. Otherwise we project p onto $\partial \,\text{conv}(P)$ in a direction in which p is maximal in K and add the projected point to K'. Abusing the notation slightly, we use $\mathcal{P}$ to denote hull of these initial points. For simplicity, we assume $\mathcal{P}$ to be a simplicial polytope.

Decomposition of $\mathcal{P}_\varepsilon \setminus \operatorname{intr} \mathcal{P}$. There are d types of simplices on $\partial\mathcal{P}$. In $\mathbb{R}^2$ these are points and edges. In $\mathbb{R}^3$ these are points, edges, and triangles. We can decompose $\mathcal{P}_\varepsilon \setminus \operatorname{intr} \mathcal{P}$ into a set of regions, each region $\sigma(f)$ corresponding to a simplex f in $\mathcal{P}$. For each simplex f in $\mathcal{P}$ let $f^* \subseteq \mathbb{S}^{d-1}$ denote the dual of f in the Gaussian diagram of $\mathcal{P}$. Recall that if f has dimension k ($0 \le k \le d-1$), then f^* has dimension $d-1-k$. The region $\mathcal{P}_\varepsilon \setminus \operatorname{intr} \mathcal{P}$ is partitioned into a collection of $|\mathcal{P}|$ regions (where $|\mathcal{P}|$ is the number of faces of all dimensions in $\mathcal{P}$). Each simplex f in $\mathcal{P}$ corresponds to a region defined

$$\sigma(f) = \{f + zu \mid 0 \le z \le \varepsilon,\, u \in f^*\}.$$

For a subsimplex $\tau \in f$, we can similarly define a region $\sigma(\tau) = \{\tau + zu \mid 0 \le z \le \varepsilon,\, u \in f^*\}$. In $\mathbb{R}^2$, there are two types of regions: point regions and edge regions. In $\mathbb{R}^3$, there are three types of regions: point regions (see Figure 4(a)), edge regions (see Figure 4(b)), and triangle regions (see Figure 4(c)).

For convenience, for any point $q = \bar{q} + z \cdot u \in \sigma(f)$, where $\bar{q} \in f, 0 \le z \le \varepsilon$, and $u \in f^*$, we write $q = \bar{q}[u, z]$ (which intuitively reads, the point whose projection onto f is $\bar{q}$ and which is at a distance z above f in direction u). We also write $q[v] = \bar{q} + z \cdot v$ (intuitively, $q[v]$ is obtained by rotating q w.r.t. f from direction u to direction v). Similarly, we write a simplex $\bar{\Delta}[u, z] = \bar{\Delta} \oplus z \cdot u$, where $\bar{\Delta}$ is a simplex inside f, $0 \le z \le \varepsilon$, and $u \in f^*$, and write $\Delta[v] = \bar{\Delta} \oplus z \cdot v$.

We will proceed to prove the upper bound as follows. For each type of region $\sigma(f)$ we place a bounded number of points from $\sigma(f) \cap \operatorname{conv}(P)$ into K' and then prove that all points in $\sigma(f) \cap \operatorname{conv}(P)$ are within a distance $\varepsilon/2$ from some point in $\mathcal{P}' = \operatorname{conv}(K')$. We begin by introducing three ways of "gridding" $\sigma(f)$ and then use these techniques to directly prove results for several base cases, which illustrate the main conceptual ideas. These base cases will already be enough to prove the results in $\mathbb{R}^2$ and $\mathbb{R}^3$. In the full version [3] we generalize this to $\mathbb{R}^d$ using an involved recursive construction. We set a few global values: $\delta = \varepsilon/12d$, $\theta = 2\arcsin(\delta/2\varepsilon)$, and $\rho = \delta/\varepsilon$.

1: Creating layers. For a point $q = \bar{q}[u, z] \in \sigma(f)$ we classify it depending on the value $z = |q - \bar{q}|$. If $z \le \varepsilon/2$, then q is already within $\varepsilon/2$ of $\mathcal{P}$. We then divide the range $[\varepsilon/2, \varepsilon]$ into a constant $H = (\varepsilon/2)/\delta$ number of cases using $\mathcal{H} = \{h_1 = \varepsilon/2, h_2 = \varepsilon/2 + \delta, \ldots, h_H = \varepsilon - \delta\}$. If $z \in [h_i, h_{i+1})$, then we set $q_{h_i} = \bar{q}[u, h_i]$. We define $\Psi_{f,h_i} \subset \sigma(f) \cap \operatorname{conv}(P)$ to be the set of points that are a distance exactly h_i from f.

2: Discretize angles. We create a constant size θ-net $U_{f,h} = \{u_1, u_2, \ldots\} \subset f^*$ of directions with the following properties. (1) For each $q = \bar{q}[u, h] \in \Psi_{f,h}$ there is a direction $u_i \in U_{f,h}$ such that the angle between u and u_i is at most θ. (2) For each $u_i \in U_{f,h}$ there is a point $p_i = \bar{p}_i[u_i, h] \in \Psi_{f,h}$; let $N_{f,h} = \{p_i \mid i \ge 1\}$. $U_{f,h}$ is constructed by first taking a $(\theta/2)$-net U_f of f^*, then for each $u_i' \in U_f$ choosing a point $p_i = \bar{q}_i[u_i, h] \in \Psi_{f,h}$ where u_i is within an angle $\theta/2$ of u_i' (if one exists), and finally placing u_i in $U_{f,h}$.

3: Exponential grid. Define a set $\mathcal{D} = \{d_0, d_1 = (1+\rho)d_0, \ldots, d_m = (1+\rho)^m d_0\}$ of distances where $d_m < 1$ and $d_0 = \delta$, so $m = O(\log 1/\varepsilon)$. For a face $f \in \mathcal{P}$, let any $r \in \sigma(f)$ be called a *support point of* f. Let $p_1, \ldots, p_k$ be the vertices of the k-simplex f. For each p_j, and each $d_i \in \mathcal{D}$ (where $d_i < ||p_j - \bar{r}||$), let $p_{j,i}$ be the point at distance d_i from p_j on the segment $p_j\bar{r}$. For each boundary facet F of f, define a sequence of at most m simplices $F_0, F_1, \ldots \in \text{conv}(F \cup \bar{r})$, each a homothet of F, so the vertices of F_i lie on segments $p_j\bar{r}$ where $p_j \in \partial F$ (see Figure 5(a)). The translation of each F_i is defined so it intersects a point $p_{j,i}$ (where $p_j \in \partial F$) and is as close to F as possible. This set of $(k-1)$-simplices for each F defines the exponential grid $G_{r,f}$. The full grid structure is revealed as this is applied recursively on each F_i.

The exponential grid $G_{r,\Delta}$ on a simplex Δ has two important properties for a point $q \in \Delta$:

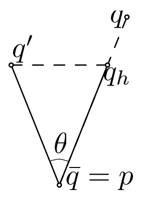
(G1) If $q \in \text{conv}(F \cap \bar{r})$ lies between boundary facet F and F_0, let q_0 be the intersection of the line segment $q\bar{r}$ with F_0; then $||q - q_0|| \leq d_0 = \delta$.

(G2) If $q \in \text{conv}(F \cap \bar{r})$ lies between F_{i-1} and F_i and the segment $q\bar{r}$ intersects F_i at q_i, let q_F be the intersection of F with the ray $\overrightarrow{rq}$; then $||q_i - q||/||q_i - q_F|| \leq \rho = \delta/\varepsilon$.

We now describe how to handle certain simple types of regions: where f is a point or an edge. These will be handled the same regardless of the dimension of the problem, and they (the edge case in particular) will be used as important base cases for higher dimensional problems.

Point regions. Consider a point region $\sigma(p)$. For each $h \in \mathcal{H}$ create θ-net $U_{p,h}$ for $\Psi_{p,h}$, so $N_{p,h}$ are the corresponding points where each $p_i = p[h, u_i] \in N_{p,h}$ has $u_i \in U_{p,h}$. Put each $N_{p,h}$ in K'.

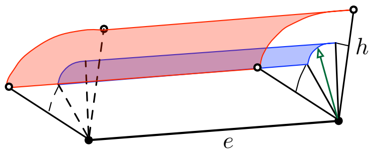
For any point $q = \bar{q}[u', z] \in \sigma(p) \cap \text{conv}(P)$, let $q' = \bar{q}[u, h]$ where $h \in \mathcal{H}$ is the largest value such that $h \leq z$ and $u \in U_{p,h}$ is the closest direction to u'; set $q_h = \bar{q}[u', h] = q'[u']$. First $||q - q_h|| \leq \delta$ because $z - h \leq \delta$. Second $||q_h - q'|| \leq \delta$ because the angle between u' and u is at most θ, and they are rotated about the point p. Thus $||q - q'|| \leq ||q - q_h|| + ||q_h - q'|| \leq 2\delta \leq \varepsilon/2$.



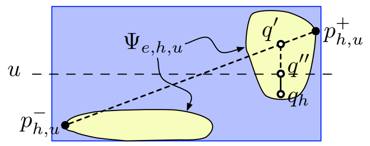
Lemma 9. *For a point region $\sigma(p)$, there exists a constant number of points $K_p \subset \sigma(p) \cap \text{conv}(P)$ such that all points $q \in \sigma(p) \cap \text{conv}(P)$ are within a distance $\varepsilon/2$ of $\text{conv}(K_p)$.*

Edge regions. Consider an edge region $\sigma(e)$ for an edge e of $\mathcal{P}$. Orient e along the x-axis. For each $h \in \mathcal{H}$ and $u \in U_{e,h}$, let $\Psi_{e,h,u}$ be the set of points in $\Psi_{e,h}$ within an angle θ of u. For each $\Psi_{e,h,u}$, we add to K_e the (two) points of $\Psi_{e,h,u}$ with the largest and smallest x-coordinates, denoted by $p^+_{h,u}$ and $p^-_{h,u}$.

For any point $q = \bar{q}[v, z] \in \sigma(e) \cap \text{conv}(P)$, there is a point $q'' = \bar{q}[u, h]$ such that $h \in \mathcal{H}$ is the largest value less than z and $u \in U_{e,h}$ is the closest direction to v. Furthermore, $||q - q''|| \leq ||q - q_h|| + ||q_h - q''|| \leq (z-h) + 2\varepsilon \sin(\theta/2) = \delta + \delta = 2\delta$. We can also argue that there is a point $q' = \bar{q}[u', z'] \in p^-_{h,u}p^+_{h,u}$, because if $\bar{q}$ has smaller x-coordinate than $\bar{p}^-_{h,u}$ or larger x-coordinate than $\bar{p}^+_{h,u}$, then q' cannot be in $\Psi_{e,h,u}$. Clearly the angle between u and u' is less than θ. This also implies that $h - z' < \delta$. Thus $||q'' - q'|| \leq 2\delta$, implying $||q - q'|| \leq 4\delta \leq \varepsilon/2$.



(a) $\sigma(e)$ in $\mathbb{R}^3$



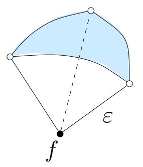
(b) top view of $\sigma(e)$ at height h

Fig. 3. Illustration of 2 points in K' for edge case with specific $h \in \mathcal{H}$ and $u \in U_{e,h,\theta}$.

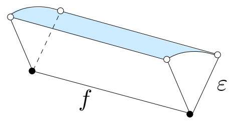
Lemma 10. *For an edge region $\sigma(e)$, there exists $O(1)$ points $K_e \subset \sigma(e) \cap \operatorname{conv}(P)$ such that for any point $q = \bar{q}[z, v] \in \sigma(e) \cap \operatorname{conv}(P)$ there is a point $p = \bar{q}[h, u] \in \operatorname{conv}(K_e)$ such that $z - h \leq 2\delta$, $||v - u|| \leq 2\delta$, and, in particular, $||q - p|| \leq 4\delta \leq \varepsilon/2$.*

For $K \subset P \in \mathbb{R}^2$ there are $|K|$ points and edges in $\mathcal{P}$. Thus combining Lemmas 9 and 10 $|K'|/|K| = O(1)$ and we have proven Theorem 2 for $d = 2$. Next, we prove the theorem for $d = 3$.

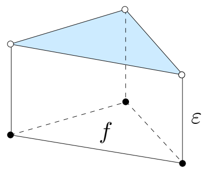
Construction of K'. Now consider $K \subset P \in \mathbb{R}^3$ and the point regions, edge regions, and triangle regions in the decomposition of $\mathcal{P}_\varepsilon \setminus \operatorname{intr} \mathcal{P}$ (see Figure 4). By Lemmas 9 and 10 we can add $O(|K|)$ points to K' to account for all point and edge regions. We can now focus on the $O(|K|)$ triangle regions.



(a) f is a vertex of $\mathcal{P}$



(b) f is an edge of $\mathcal{P}$

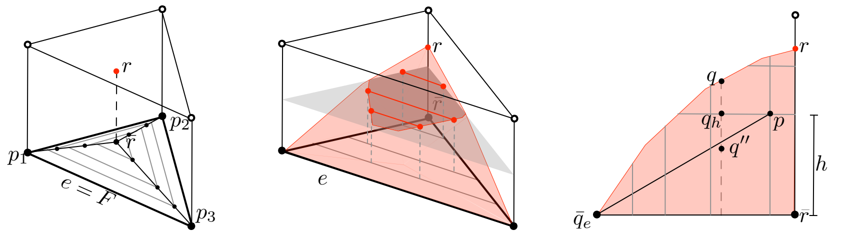


(c) f is a facet of $\mathcal{P}$

Fig. 4. Illustration of regions in the partition of $\mathcal{P}_\varepsilon \setminus \operatorname{intr} \mathcal{P}$ in three dimensions.

Consider a triangle region $\sigma(t)$ for a triangle t in $\mathcal{P}$ (see Figure 5(a)), t^* consists of a single direction, the one normal to t. Let r be the highest point of $\sigma(t) \cap \operatorname{conv}(P)$ in direction t^*. We add r to K' and we create an exponential grid $G_{r,t}$ with r as the support point. For each edge $e \in G_{r,t}$ and $h \in \mathcal{H}$ we add the intersection of $e[t^*, h]$ with the boundary of $\sigma(t) \cap \operatorname{conv}(P)$ to K', as shown in Figure 5(b). Thus, in total we add $O(|K| \log(1/\varepsilon))$ points to K'.

Proof of correctness. Consider any point $q = \bar{q}[t^*, z] \in \sigma(t) \cap \operatorname{conv}(P)$ and associate it with a boundary edge e of t such that $\bar{q} \in \operatorname{conv}(e \cup \bar{r})$. Let $q_h = \bar{q}[t^*, h]$ where $h \in \mathcal{H}$ is the largest height such that $h \leq z$. If segment $\bar{q}\bar{r}$ does not intersect any edge e_i parallel to e in $G_{r,t}$, let $\bar{p} = \bar{r}$. Otherwise, let e_i be the first segment parallel to e in $G_{r,t}$ intersected by the ray $\overrightarrow{\bar{q}\bar{r}}$, and let $\bar{p}$ be the intersection. Let $p = \bar{p}[t^*, h]$ which must be in $\operatorname{conv}(K')$ by construction. If $e_i = e_0$, then by (G1) we have $||q_h - p|| = ||\bar{q} - \bar{p}|| \leq \delta$, thus $||q - p|| \leq 2\delta \leq \varepsilon/2$ and we are done. Otherwise, let $\bar{q}_e$ be the intersection of e with ray $\overrightarrow{\bar{r}\bar{q}}$. By (G2) $||\bar{p} - \bar{q}||/||\bar{p} - \bar{q}_e|| \leq \rho = \delta/\varepsilon$. Thus, $q'' = \bar{q}[t^*, h - \varepsilon\rho]$ is below the segment $\bar{q}_e p$ (see



(a) $\sigma(t)$ with r and $G_{r,t}$ (b) Subtriangle t_e of $\sigma(t)$ (c) Slice of (b) through r, q

Fig. 5. Illustration to aid correctness of approximation of triangle regions in $\mathbb{R}^3$.

Figure 5(c)) and thus $q'' \in \text{conv}(K')$ since triangle $p\bar{p}\bar{q}_e$ is in $\text{conv}(K')$. Finally, $||q - q''|| = ||q - q_h|| + ||q_h - q''|| \leq 2\delta \leq \varepsilon/2$. This proves Theorem 2 for $d = 3$.

3.1 Remarks

(1) For $d = 2, 3$, $\kappa(P, \varepsilon/2)$ is only a factor of $O(1)$ and $O(\log(1/\varepsilon))$, respectively, larger than $\kappa(P, \varepsilon)$; therefore, the sizes of optimal ε-kernels in these dimensions are relative stable. However, for $d \geq 4$, the stability drastically reduces in the worst case because of the superlinear dependency on $\kappa(P, \varepsilon)$.

(2) Neither the upper nor the lower bound in the theorem is tight. For $d = 3$, we can prove a tighter lower bound of $\Omega\big(\kappa(P, \varepsilon) \log(1/(\varepsilon \cdot \kappa(P, \varepsilon)))\big)$. We conjecture in $\mathbb{R}^d$ that

$$\kappa(P, \varepsilon/2) = \Theta\big(\kappa(P, \varepsilon)^{\lfloor d/2 \rfloor} \log^{d-2}(1/(\varepsilon^{(d-1)/2} \cdot \kappa(P, \varepsilon)))\big).$$

References

1. P. K. Agarwal, S. Har-Peled, and K. Varadarajan, Approximating extent measure of points, *Journal of ACM*, 51 (2004), 606–635.
2. P. K. Agarwal, S. Har-Peled, and K. Varadarajan, Geometric approximations via coresets, in: *Combinatorial and Computational Geometry*, 2005, pp. 1–31.
3. P. K. Agarwal, J. M. Phillips, and H. Yu. Stability of ε-kernels. arXiv:1003.5874.
4. P. K. Agarwal and H. Yu, A space-optimal data-stream algorithm for coresets in the plane, *SoCG*, 2007, pp. 1–10.
5. G. Barequet and S. Har-Peled, Efficiently approximating the minimum-volume bounding box of a point set in three dimensions, *Journ. of Algs*, 38 (2001), 91–109.
6. T. Chan, Faster core-set constructions and data-stream algorithms in fixed dimensions, *Computational Geometry: Theory and Applications*, 35 (2006), 20–35.
7. T. Chan, Dynamic coresets, *SoCG*, 2008, pp. 1–9.
8. S. Har-Peled, *Approximation Algorithm in Geometry (Chapter 22)*, http://valis.cs.uiuc.edu/˜sariel/teach/notes/aprx/, 2010.
9. J. Hershberger and S. Suri, Adaptive sampling for geometric problems over data streams, *Computational Geometry: Theory and Applications*, 39 (2008), 191–208.
10. H. Yu, P. K. Agarwal, R. Poreddy, and K. Varadarajan, Practical methods for shape fitting and kinetic data structures using coresets, *Algorithmica*, 52 (2008).
11. H. Zarrabi-Zadeh, An almost space-optimal streaming algorithm for coresets in fixed dimensions, *ESA*, 2008, pp. 817–829.