Stability Yields Sublinear Time Algorithms for Geometric Optimization in Machine Learning

Hu Ding
School of Computer Science and Technology,
University of Science and Technology of China, Anhui, China

Abstract

In this paper, we study several important geometric optimization problems arising in machine learning. First, we revisit the Minimum Enclosing Ball (MEB) problem in Euclidean space $\mathbb{R}^d$. The problem has been extensively studied before, but real-world machine learning tasks often need to handle large-scale datasets so that we cannot even afford linear time algorithms. Motivated by the recent developments on beyond worst-case analysis, we introduce the notion of stability for MEB, which is natural and easy to understand. Roughly speaking, an instance of MEB is stable, if the radius of the resulting ball cannot be significantly reduced by removing a small fraction of the input points. Under the stability assumption, we present two sampling algorithms for computing radius-approximate MEB with sample complexities independent of the number of input points n. In particular, the second algorithm has the sample complexity even independent of the dimensionality d. We also consider the general case without the stability assumption. We present a hybrid algorithm that can output either a radius-approximate MEB or a covering-approximate MEB, which improves the running time and the number of passes for the previous sublinear MEB algorithms. Further, we extend our proposed notion of stability and design sublinear time algorithms for other geometric optimization problems including MEB with outliers, polytope distance, one-class and two-class linear SVMs (without or with outliers). Our proposed algorithms also work fine for kernels.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis

Keywords and phrases stability, sublinear time, geometric optimization, machine learning

Digital Object Identifier 10.4230/LIPIcs.ESA.2021.38

Acknowledgements The author wants to thank Jinhui Xu and the anonymous reviewers for their helpful comments and suggestions for improving the paper.

1 Introduction

Many real-world machine learning tasks can be formulated as geometric optimization problems in Euclidean space. We start with a fundamental geometric optimization problem, *Minimum Enclosing Ball (MEB)*, which has attracted a lot of attentions in past years. Given a set P of n points in Euclidean space $\mathbb{R}^d$, where d could be quite high, the problem of MEB is to find a ball with minimum radius to cover all the points in P [16,38,60]. MEB finds several important applications in machine learning [68]. For example, the popular classification model *Support Vector Machine (SVM)* can be formulated as an MEB problem in high dimensional space, if all the mapped points have the same norm by using kernel method, *e.g.,* the popular radial basis function kernel [80]. Hence fast MEB algorithms can be adopted to speed up its training procedure [24, 25, 80]. Recently, MEB has also been used for preserving privacy [37, 69] and quantum cryptography [46].

Usually, we consider the approximate solutions of MEB. If a ball covers all the n points but has a radius larger than the optimal one, we call it a **"radius-approximate solution"**; if a ball has the radius no larger than the optimal one but covers less than n points, we call it a **"covering-approximate solution"** instead (the formal definitions are shown in Section 3). In the era of big data, the dataset could be so large that we cannot even afford linear time algorithms. This motivates us to ask the following questions:

29th Annual European Symposium on Algorithms (ESA 2021).
Editors: Petra Mutzel, Rasmus Pagh, and Grzegorz Herman; Article No. 38; pp. 38:1–38:19
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Is it possible to develop approximation algorithms for MEB that run in sublinear time in the input size? And how about other high-dimensional geometric optimization problems arising in machine learning?

It is common to assume that the input data is represented by a $n \times d$ matrix, and thus any algorithm having complexity $o(nd)$ can be considered as a sublinear time algorithm. In practice, data items are usually represented as sparse vectors in $\mathbb{R}^d$; thus it can be fast to perform the operations, like distance computing, even though the dimensionality d is high (see the concluding remarks of [25]). Moreover, the number of input points n is often much larger than the dimensionality d in many real-world scenarios. **Therefore, we are interested in designing the algorithms that have complexities sublinear in n (or linear in n but with small factor before it).** Designing sublinear time algorithms has become a promising approach to handle many big data problems, and a detailed discussion on previous works is given in Section 2.

1.1 Our Main Ideas and Results

Our idea for designing sublinear time MEB algorithms is inspired by the recent developments on optimization with respect to stable instances, under the umbrella of *beyond worst-case analysis* [74]. For example, several recent works introduced the notion of stability for problems like clustering and max-cut [8, 10, 15]. In this paper, we give the notion of "stability" for MEB. Roughly speaking, an instance of MEB is stable, if the radius of the resulting ball cannot be significantly reduced by removing a small fraction of the input points (*e.g.,* the radius cannot be reduced by 10% if only 1% of the points are removed). The rationale behind this notion is quite natural: if the given instance is not stable, the small fraction of points causing significant reduction in the radius should be viewed as outliers (or we may need multiple balls to cover the input points as k-center clustering [45, 52]). To the best of our knowledge, this is the first study on MEB from the perspective of stability.

We prove an important implication of the stability assumption: informally speaking, if an instance of MEB is stable, its center should reveal a certain extent of robustness in the space (Section 4). Using this implication, we propose two sampling algorithms for computing $(1+\epsilon)$-radius approximate MEB with sublinear time complexities (Section 5); in particular, our second algorithm has the sample size (*i.e.,* the number of sampled points) independent of the number of input points n and dimensionality d (to the best of our knowledge, this is the first algorithm achieving $(1+\epsilon)$-radius approximation with such a sublinear complexity).

Moreover, we have an interesting observation: the ideas developed under the stability assumption can even help us to solve the general instance without the stability assumption, if we relax the requirement slightly. We introduce a hybrid approach that can output either a radius-approximate MEB or a covering-approximate MEB, depending upon whether the input instance is sufficiently stable[1] (Section 6). It is worth noting that the simple uniform sampling idea based on VC-dimension [49, 81] can only yield a "bi-criteria" approximation, which has errors on both the radius and the number of covered points (see the discussion on our first sampling algorithm in Section 5.1). Comparing with the sublinear time MEB algorithm proposed by Clarkson *et al.* [25], we reduce the total running time from $\tilde{O}(\epsilon^{-2}n + \epsilon^{-1}d + M)$ to $O(n + h(\epsilon, \delta) \cdot d + M)$, where M is the number of non-zero entries in the input $n \times d$ matrix and $h(\epsilon, \delta)$ is a factor depending on the pre-specified radius error bound ϵ and covering error bound δ. Thus, our improvement is significant if $n \gg d$. The only tradeoff is that we allow a

[1] We do not need to know whether the instance is stable or not, when running our algorithm.

covering approximation for unstable instance (given the lower bound proved by [25], it is quite unlikely to reduce the term $\epsilon^{-2}n$ if we keep restricting the output to be $(1+\epsilon)$-radius approximation). Moreover, our algorithm only needs uniform sampling and a single pass over the data; on the other hand, the algorithm of [25] needs $\tilde{O}(\epsilon^{-1})$ passes (the details are shown in Table 1).

Table 1 The existing and our results for computing MEB in high dimensions. In the table, "rad." and "cov." stand for "radius approximation" and "covering approximation", respectively. "M" is the number of non-zero entries in the input $n \times d$ matrix. The factor C_1 depends on ϵ and the stability degree of the given instance; the factor C_2 depends on ϵ and δ. The mark "$*$" means that the method can be extended for MEB with outliers.

Results		Quality	Time	Number of passes
Clarkson *et al.* [25]		$(1+\epsilon)$-rad.	$\tilde{O}(\epsilon^{-2}n + \epsilon^{-1}d + M)$	$\tilde{O}(\epsilon^{-1})$
Core-sets methods* [16, 24, 60, 71]		$(1+\epsilon)$-rad.	roughly $O(\epsilon^{-1}nd)$ or $O(\epsilon^{-1}(n+d+M))$ if $M = o(nd)$	$O(\epsilon^{-1})$
Numerical method [76]		$(1+\epsilon)$-rad.	$\tilde{O}(\epsilon^{-1/2}nd)$ or $\tilde{O}(\epsilon^{-1/2}(n+d+M))$ if $M = o(nd)$	$O(\epsilon^{-1/2})$
Numerical method [6]		$(1+\epsilon)$-rad.	$\tilde{O}(nd + n\sqrt{d}/\sqrt{\epsilon})$	$\tilde{O}(d + \sqrt{d/\epsilon})$
Streaming algorithm [4, 21]		1.22-rad.	$O(nd/\epsilon^5)$	one pass
This paper	stable instance*	$(1+\epsilon)$-rad.	$O(C_1 \cdot d)$	uniform sampling
	general instance*	$(1+\epsilon)$-rad. or $(1-\delta)$-cov.	$O\big((n + C_2)d\big)$ or $O(n + C_2 \cdot d + M)$ if $M = o(nd)$	uniform sampling plus a single pass

Our proposed notion of stability can be naturally extended to several other geometric optimization problems arising in machine learning.

MEB with outliers. In practice, we often assume the presence of outliers in given datasets. MEB with outliers is a natural generalization of the MEB problem, where the goal is to find the minimum ball covering at least a certain fraction of input points. The presence of outliers makes the problem not only non-convex but also highly combinatorial in high dimensions. We define the stability for MEB with outliers, and propose the sublinear time approximation algorithms. Our algorithms are the first sublinear time **single-criterion** approximation algorithms for the MEB with outliers problem (comparing with the previous bi-criteria approximations like [18, 31]), to the best of our knowledge.

Polytope distance and SVM. Given a set P of points in $\mathbb{R}^d$, the *polytope distance* problem is to compute the shortest distance of any point inside the convex hull of P to the origin. Similar to MEB, polytope distance is also a fundamental problem in computational geometry and has many important applications, such as *sparse approximation* [24]. The polytope distance problem is also closely related to SVMs. Actually, training linear SVM is equivalent to solving the polytope distance problem for the Minkowski difference of two differently labeled training datasets [41]. Though polytope distance is quite different from the MEB problem, they in fact share several common features. For instance, both of them can be solved by the greedy core-set construction method [24]. Following our ideas for MEB, we define the stability for polytope distance, and propose the sublinear time algorithms.

Because the geometric optimization problems studied in this paper are motivated from machine learning applications, we also take into account the **kernels** [78]. Our proposed algorithms only need to conduct the basic operations, like computing the distance and inner product, on the data items. Therefore, they also work fine for kernels.

The rest of the paper is organized as follows. In Section 2, we summarize the previous results that are related to our work. In Section 3, we present the important definitions and briefly introduce the coreset construction method for MEB from [16] (which will be used in our following algorithms design and analysis). In Section 4, we prove the implication of MEB stability. Further, we propose two sublinear time MEB algorithms in Section 5. We also briefly introduce several important extensions in Section 6; due to the space limit, we leave the details to our full paper.

2 Previous Work

The works most related to ours are [7,25]. Clarkson *et al.* [25] developed an elegant perceptron framework for solving several optimization problems arising in machine learning, such as MEB. Given a set of n points in $\mathbb{R}^d$ represented as an $n \times d$ matrix with M non-zero entries, their framework can compute the MEB in $\tilde{O}(\frac{n}{\epsilon^2} + \frac{d}{\epsilon})$ time [2]. Note that the parameter "ϵ" is an additive error (*i.e.,* the resulting radius is $r + \epsilon$ if r is the radius of the optimal MEB) which can be converted into a relative error (*i.e.,* $(1+\epsilon)r$) in $O(M)$ preprocessing time. Thus, if $M = o(nd)$, the running time is still sublinear in the input size nd (please see Table 1). The framework of [25] also inspires the sublinear time algorithms for training SVMs [51] and approximating Semidefinite Programs [40]. Hayashi and Yoshida [50] presented a sampling-based method for minimizing quadratic functions of which the MEB objective is a special case, but it yields a large additive error $O(\epsilon n^2)$.

Alon *et al.* [7] studied the following property testing problem: given a set of n points in some metric space, determine whether the instance is (k, b)-clusterable, where an instance is called (k, b)-clusterable if it can be covered by k balls with radius (or diameter) $b > 0$. They proposed several sampling algorithms to answer the question "approximately". Specifically, they distinguish between the case that the instance is (k, b)-clusterable and the case that it is ϵ-far away from (k, b')-clusterable, where $\epsilon \in (0, 1)$ and $b' \geq b$. "ϵ-far" means that more than ϵn points should be removed so that it becomes (k, b')-clusterable. Note that their method cannot yield a single criterion radius-approximation or covering-approximation algorithm for the MEB problem, since it will introduce unavoidable errors on the radius and the number of covered points due to the relaxation of "ϵ-far". But it is possible to convert it into a "**bi-criteria**" approximation, where it allows approximations on both the radius and the number of uncovered outliers (*e.g.,* discard more than the pre-specified number of outliers).

MEB and core-set. A *core-set* is a small set of points that approximates the structure/shape of a much larger point set [1, 35, 72]. The core-set idea has also been used to compute approximate MEB in high dimensional space [18,57,60,71]. Bădoiu and Clarkson [16] showed

[2] The asymptotic notation $\tilde{O}(f) = O\big(f \cdot \texttt{polylog}(\frac{nd}{\epsilon})\big)$.

that it is possible to find a core-set of size $\lceil 2/\epsilon \rceil$ that yields a $(1+\epsilon)$-radius approximate MEB. Several other methods can yield even lower core-set sizes, such as [17, 57]. In fact, the algorithm for computing the core-set of MEB is a *Frank-Wolfe* algorithm [39], which has been systematically studied by Clarkson [24]. Other MEB algorithms that do not rely on core-sets include [6, 38, 76]. Agarwal and Sharathkumar [4] presented a streaming $(\frac{1+\sqrt{3}}{2}+\epsilon)$-radius approximation algorithm for computing MEB; later, Chan and Pathak [21] proved that the same algorithm actually yields an approximation ratio less than 1.22.

MEB with outliers and bi-criteria approximations. The MEB with outliers problem can be viewed as the case $k = 1$ of the k-center clustering with outliers problem [22]. Bădoiu *et al.* [18] extended their core-set idea to the problems of MEB and k-center clustering with outliers, and achieved linear time bi-criteria approximation algorithms (if k is assumed to be a constant). Huang *et al.* [53] and Ding *et al.* [31, 33] respectively showed that simple uniform sampling approach can yield bi-criteria approximation of k-center clustering with outliers. Several algorithms for the low dimensional MEB with outliers have also been developed [5, 34, 47, 62]. There also exist a number of works on streaming MEB and k-center clustering with outliers [20, 23, 63, 82]. Other related topics include robust optimization [14], robust fitting [3, 48], and optimization with uncertainty [19].

Polytope distance and SVMs. The Gilbert's algorithm [42] is one of the earliest known algorithms for computing polytope distance. Similar to the core-set construction of MEB, the Gilbert's algorithm is also an instance of the Frank-Wolfe algorithm where the upper bound of the number of iterations is independent of the data size and dimensionality [24, 41]. In general, SVM can be formulated as a quadratic programming problem, and a number of efficient techniques have been developed besides the Gilbert's algorithm, such as the soft margin SVM [26, 73], ν-SVM [27, 77], and CVM [80].

Optimizations under stability. Bilu and Linial [15] showed that the Max-Cut problem becomes easier if the given instance is stable with respect to perturbation on edge weights. Ostrovsky *et al.* [70] proposed a separation condition for k-means clustering which refers to the scenario where the clustering cost of k-means is significantly lower than that of $(k-1)$-means for a given instance, and demonstrated the effectiveness of the Lloyd heuristic [61] under the separation condition. Balcan *et al.* [10] introduced the concept of approximation-stability for finding the ground-truth of k-median and k-means clustering. Awasthi *et al.* [8] introduced another notion of clustering stability and gave a PTAS for k-median and k-means clustering. More clustering algorithms under stability assumption were studied in [9, 11–13, 59].

Sublinear time algorithms. Besides the aforementioned sublinear MEB algorithm [25], a number of sublinear time algorithms have been studied for the problems like clustering [29, 54, 55, 64, 65] and property testing [44]. More detailed discussion on sublinear time algorithms can be found in the survey papers [28, 75].

3 Definitions and Preliminaries

We describe and analyze our algorithms in the unit-cost RAM model [66]. Suppose the input is represented by an $n \times d$ matrix (*i.e.,* n points in $\mathbb{R}^d$). As mentioned in [25], it is common to assume that any entry of the matrix can be recovered in constant time.

We let $|A|$ denote the number of points of a given point set A in $\mathbb{R}^d$, and $||x-y||$ denote the Euclidean distance between two points x and y in $\mathbb{R}^d$. We use $\mathbb{B}(c,r)$ to denote the ball centered at a point c with radius $r>0$. Below, we give the definitions for MEB and the notion of stability. To keep the structure of our paper more compact, we place other necessary definitions for our extensions to the full paper.

▶ **Definition 1** (Minimum Enclosing Ball (MEB)). *Given a set P of n points in $\mathbb{R}^d$, the MEB problem is to find a ball with minimum radius to cover all the points in P. The resulting ball and its radius are denoted by $\mathbf{MEB}(P)$ and $\mathbf{Rad}(P)$, respectively.*

▶ **Definition 2** (Radius Approximation and Covering Approximation). *Let $0<\epsilon,\delta<1$. A ball $\mathbb{B}(c,r)$ is called a $(1+\epsilon)$-radius approximation of $\mathbf{MEB}(P)$, if the ball covers all points in P and has radius $r\leq(1+\epsilon)\mathbf{Rad}(P)$. On the other hand, the ball is called a $(1-\delta)$-covering approximation of $\mathbf{MEB}(P)$, if it covers at least $(1-\delta)n$ points in P and has radius $r\leq\mathbf{Rad}(P)$.*

Both radius approximation and covering approximation are single-criterion approximations. When ϵ (*resp.,* δ) approaches to 0, the $(1+\epsilon)$-radius approximation (*resp.,* $(1-\delta)$-covering approximation) will approach to $\mathbf{MEB}(P)$. The "covering approximation" seems to be similar to "MEB with outliers", but actually they are quite different.

▶ **Definition 3** ((α,β)-stable). *Given a set P of n points in $\mathbb{R}^d$ with two parameters α and β in $(0,1)$, P is an (α,β)-stable instance if **(1)** $\mathbf{Rad}(P')>(1-\alpha)\mathbf{Rad}(P)$ for any $P'\subset P$ with $|P'|>(1-\beta)n$, and **(2)** there exists a $P''\subset P$ with $|P''|=(1-\beta)n$ having $\mathbf{Rad}(P'')\leq(1-\alpha)\mathbf{Rad}(P)$.*

The intuition of Definition 3. Actually, β can be viewed as a function of α. For any $\alpha>0$, there always exists a $\beta\geq\frac{1}{n}$ such that P is an (α,β)-stable instance ($\beta\geq\frac{1}{n}$ because we must remove at least one point). The property of stability indicates that $\mathbf{Rad}(P)$ cannot be significantly reduced unless removing a large enough fraction of points from P. For a fixed α, the larger β is, the more stable P becomes. Actually, our stability assumption is quite reasonable in practice. For example, if the radius can be reduced considerably (say by $\alpha=10\%$) after removing only a very small fraction (say $\beta=1\%$) of points, it is natural to view the small fraction of points as outliers. In practice, it is difficult to obtain the exact value of β for a fixed α. However, the value of β only affects the sample sizes in our proposed algorithms in Section 5, and thus only assuming a reasonable lower bound $\beta_0<\beta$ is already sufficient. To better understand the notion of stability in high dimensions, we consider the following two examples.

Example (i). Suppose that the distribution of P is uniform and dense inside $\mathbf{MEB}(P)$. Let $\alpha\in(0,1)$ be a fixed number, and we study the corresponding β of P. If we want the radius of the remaining $(1-\beta)n$ points to be as small as possible, intuitively we should remove the outermost βn points (since P is uniform and dense). Let P'' denote the set of innermost $(1-\beta)n$ points that has $\mathbf{Rad}(P'')\leq(1-\alpha)\mathbf{Rad}(P)$. Then we have $\frac{|P''|}{|P|}\approx\frac{Vol\left(\mathbf{MEB}(P'')\right)}{Vol\left(\mathbf{MEB}(P)\right)}=\frac{(\mathbf{Rad}(P''))^d}{(\mathbf{Rad}(P))^d}\leq(1-\alpha)^d$, where $Vol(\cdot)$ is the volume function. That is, $1-\beta\leq(1-\alpha)^d$ and thus $\lim_{d\to\infty}\beta=1$ when α is fixed; that means P tends to be very stable as d increases.

Example (ii). Consider a regular d-dimensional simplex P containing $d+1$ points where each pair of points have the pairwise distance equal to 1. It is not hard to obtain $\mathbf{Rad}(P) = \sqrt{\frac{d}{2(1+d)}}$, and we denote it by r_d. If we remove $\beta(d+1)$ points from P, namely it becomes a regular d'-dimensional simplex with $d' = (1-\beta)(d+1)-1$, the new radius $r_{d'} = \sqrt{\frac{d'}{2(1+d')}}$. To achieve $\frac{r_{d'}}{r_d} \leq 1-\alpha$ with a fixed α, it is easy to see that $1-\beta$ should be no larger than $\frac{1}{1+(2\alpha-\alpha^2)d}$ and thus $\lim_{d\to\infty}\beta = 1$. Similar to example (i), the instance P tends to be very stable as d increases.

3.1 Core-set Construction for MEB [16]

To compute a $(1+\epsilon)$-radius approximate MEB, Bădoiu and Clarkson [16] proposed an algorithm yielding an MEB core-set of size $2/\epsilon$ (for convenience, we always assume that $2/\epsilon$ is an integer). We first briefly introduce their main idea, since it will be used in our proposed algorithms (we do not use the MEB algorithm of [24] because it is not quite convenient to analyze under our stability assumption; the other construction algorithms [17, 57], though achieving lower core-set sizes, are more complicated and thus not applicable to our problems).

Given a point set $P \subset \mathbb{R}^d$, the algorithm is a simple iterative procedure. Initially, it selects an arbitrary point from P and places it into an initially empty set T. In each of the following $2/\epsilon$ iterations, the algorithm updates the center of $\mathbf{MEB}(T)$ and adds to T the farthest point from the current center of $\mathbf{MEB}(T)$. Finally, the center of $\mathbf{MEB}(T)$ induces a $(1+\epsilon)$-radius approximation for $\mathbf{MEB}(P)$. The selected set of $2/\epsilon$ points (*i.e.*, T) is called the core-set of MEB. However, computing the exact center of $\mathbf{MEB}(T)$ could be expensive; in practice, one may only compute an approximate center of $\mathbf{MEB}(T)$ in each iteration. In the i-th iteration, we let c_i denote the exact center of $\mathbf{MEB}(T)$; also, let r_i be the radius of $\mathbf{MEB}(T)$. Suppose ξ is a given number in $(0,1)$. Using another algorithm proposed in [16, Section 3], one can compute an approximate center o_i having the distance to c_i less than ξr_i in $O(\frac{1}{\xi^2}|T|d)$ time. Since we only compute o_i rather than c_i in each iteration, we in fact only select the farthest point to o_i (not c_i). In [31], Ding provided a more careful analysis on Bădoiu and Clarkson's method and presented the following theorem.

▶ **Theorem 4** ([31]). *In the core-set construction algorithm of [16], if one computes an approximate MEB for T in each iteration and the resulting center o_i has the distance to c_i less than ξr_i with $\xi = s\frac{\epsilon}{1+\epsilon}$ for some $s \in (0,1)$, the final core-set size is bounded by $z = \frac{2}{(1-s)\epsilon}$. Also, the bound could be arbitrarily close to $2/\epsilon$ when s is sufficiently small.*

▶ Remark 5. **(i)** We can simply set s to be any constant in $(0,1)$; for instance, if $s = 1/3$, the core-set size will be bounded by $z = 3/\epsilon$. Since $|T| \leq z$ in each iteration, the total running time is $O\Big(z\big(|P|d + \frac{1}{\xi^2}zd\big)\Big) = O\Big(\frac{1}{\epsilon}\big(|P| + \frac{1}{\epsilon^3}\big)d\Big)$. **(ii)** We also want to emphasize a simple observation mentioned in [18, 31] on the above core-set construction procedure, which will be used in our algorithms and analyses later on. The algorithm always selects the farthest point to o_i in each iteration. However, this is actually not necessary. As long as the selected point has distance at least $(1+\epsilon)\mathbf{Rad}(P)$, the result presented in Theorem 4 is still true. If no such a point exists (*i.e.*, $P \setminus \mathbb{B}\big(o_i, (1+\epsilon)\mathbf{Rad}(P)\big) = \emptyset$), a $(1+\epsilon)$-radius approximate MEB (*i.e.*, the ball $\mathbb{B}\big(o_i, (1+\epsilon)\mathbf{Rad}(P)\big)$) has been already obtained.

▶ Remark 6 (kernels). If each point $p \in P$ is mapped to $\psi(p)$ in $\mathbb{R}^D$ by some kernel function (*e.g.*, as the CVM [80]), where D could be $+\infty$, we can still run the core-set algorithm of [16, 58], since the algorithm only needs to compute the distances and the center o_i is always a convex combination of T in each iteration; instead of returning an explicit center, the algorithm will output the coefficients of the convex combination for the center. And similarly, our Algorithm 2 presented in Section 5.2 also works fine for kernels.

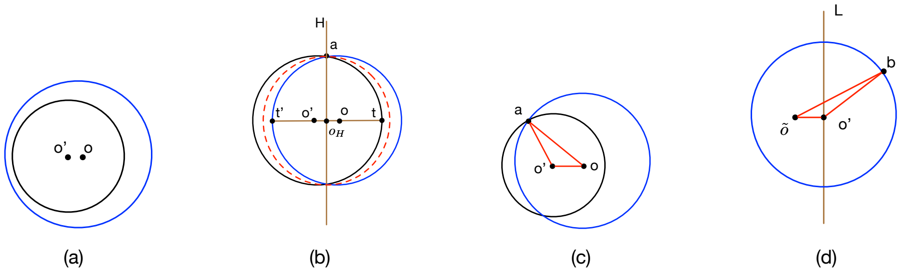

Figure 1 (a) The case $MEB(P') \subset MEB(P)$; (b) an illustration under the assumption $\angle ao'o < \pi/2$ in the proof of Claim 9; (c) the angle $\angle ao'o \geq \pi/2$; (d) an illustration of Lemma 10.

4 Implication of the Stability Property

In this section, we show an important implication of the stability property of Definition 3.

▶ **Theorem 7.** *Assume $\epsilon, \epsilon', \beta_0 \in (0,1)$. Let P be an (ϵ^2, β)-stable instance of the MEB problem with $\beta > \beta_0$, and o be the center of its MEB. Let $\tilde{o}$ be a given point in $\mathbb{R}^d$. Assume the number $r \leq (1+\epsilon'^2)\mathbf{Rad}(P)$. If the ball $\mathbb{B}(\tilde{o}, r)$ covers at least $(1-\beta_0)n$ points from P, the following holds*

$$||\tilde{o} - o|| < (2\sqrt{2}\epsilon + \sqrt{3}\epsilon')\mathbf{Rad}(P). \tag{1}$$

Theorem 7 indicates that if a ball covers a large enough subset of P and its radius is bounded, its center should be close to the center of $\mathbf{MEB}(P)$. Let $P' = \mathbb{B}(\tilde{o}, r) \cap P$, and assume o' is the center of $\mathbf{MEB}(P')$. To bound the distance between $\tilde{o}$ and o, we bridge them by the point o' (since $||\tilde{o} - o|| \leq ||\tilde{o} - o'|| + ||o' - o||$). The following are two key lemmas for proving Theorem 7.

▶ **Lemma 8.** *The distance $||o' - o|| \leq \sqrt{2}\epsilon\mathbf{Rad}(P)$.*

Proof. We consider two cases: $\mathbf{MEB}(P')$ is totally covered by $\mathbf{MEB}(P)$ and otherwise. For the first case (see Figure 1(a)), it is easy to see that

$$||o' - o|| \leq \mathbf{Rad}(P) - (1-\epsilon^2)\mathbf{Rad}(P) = \epsilon^2\mathbf{Rad}(P) < \sqrt{2}\epsilon\mathbf{Rad}(P), \tag{2}$$

where the first inequality comes from the fact that $\mathbf{MEB}(P')$ has radius at least $(1 - \epsilon^2)\mathbf{Rad}(P)$ (Definition 3). Thus, we can focus on the second case below.

Let a be any point located on the intersection of the two spheres of $\mathbf{MEB}(P')$ and $\mathbf{MEB}(P)$. Consequently, we have the following claim.

▷ Claim 9. The angle $\angle ao'o \geq \pi/2$.

Proof. Suppose that $\angle ao'o < \pi/2$. Note that $\angle aoo'$ is always smaller than $\pi/2$ since $||o - a|| = \mathbf{Rad}(P) \geq \mathbf{Rad}(P') = ||o' - a||$. Therefore, o and o' are separated by the hyperplane H that is orthogonal to the segment $\overline{o'o}$ and passes through the point a. See Figure 1(b). Now we show that P' can be covered by a ball smaller than $\mathbf{MEB}(P')$. Let o_H be the point $H \cap \overline{o'o}$, and t (*resp.*, t') be the point collinear with o and o' on the right side of the sphere of $\mathbf{MEB}(P')$ (*resp.*, left side of the sphere of $\mathbf{MEB}(P)$; see Figure 1(b)). Then, we have

$$\begin{aligned} &||t - o_H|| + ||o_H - o'|| = ||t - o'|| = ||a - o'|| < ||o' - o_H|| + ||o_H - a|| \\ &\qquad \Longrightarrow ||t - o_H|| < ||o_H - a||. \end{aligned} \tag{3}$$

Similarly, we have $||t' - o_H|| < ||o_H - a||$. Consequently, $\mathbf{MEB}(P) \cap \mathbf{MEB}(P')$ is covered by the ball $\mathbb{B}(o_H, ||o_H - a||)$. Further, because P' is covered by $\mathbf{MEB}(P) \cap \mathbf{MEB}(P')$ and $||o_H - a|| < ||o' - a|| = \mathbf{Rad}(P')$, P' is covered by the ball $\mathbb{B}(o_H, ||o_H - a||)$ that is smaller than $\mathbf{MEB}(P')$. This contradicts to the fact that $\mathbf{MEB}(P')$ is the minimum enclosing ball of P'. Thus, the claim $\angle ao'o \geq \pi/2$ is true. ◁

Given Claim 9, we know that $||o' - o|| \leq \sqrt{\big(\mathbf{Rad}(P)\big)^2 - \big(\mathbf{Rad}(P')\big)^2}$. See Figure 1(c). Moreover, Definition 3 implies that $\mathbf{Rad}(P') \geq (1 - \epsilon^2)\mathbf{Rad}(P)$. Therefore, we have

$$||o' - o|| \leq \sqrt{\big(\mathbf{Rad}(P)\big)^2 - \big((1 - \epsilon^2)\mathbf{Rad}(P)\big)^2} \leq \sqrt{2}\epsilon\mathbf{Rad}(P). \tag{4}$$

◀

▶ **Lemma 10.** *The distance* $||\tilde{o} - o'|| < (\sqrt{2}\epsilon + \sqrt{3}\epsilon')\mathbf{Rad}(P)$.

Proof. Let L be the hyperplane orthogonal to the segment $\overline{\tilde{o}o'}$ and passing through the center o'. Suppose $\tilde{o}$ is located on the left side of L. Then, there exists a point $b \in P'$ located on the right closed semi-sphere of $\mathbf{MEB}(P')$ divided by L (this result was proved in [18, 43] and see Lemma 2.2 in [18]). See Figure 1(d). That is, the angle $\angle bo'\tilde{o} \geq \pi/2$. As a consequence, we have

$$||\tilde{o} - o'|| \leq \sqrt{||\tilde{o} - b||^2 - ||b - o'||^2}. \tag{5}$$

Moreover, since $||\tilde{o} - b|| \leq r \leq (1 + \epsilon'^2)\mathbf{Rad}(P)$ and $||b - o'|| = \mathbf{Rad}(P') \geq (1 - \epsilon^2)\mathbf{Rad}(P)$, (5) implies that $||\tilde{o} - o'|| \leq \sqrt{(1 + \epsilon'^2)^2 - (1 - \epsilon^2)^2}\mathbf{Rad}(P)$, where this upper bound is equal to

$$\sqrt{2\epsilon'^2 + \epsilon'^4 + 2\epsilon^2 - \epsilon^4}\mathbf{Rad}(P) < \sqrt{3\epsilon'^2 + 2\epsilon^2}\mathbf{Rad}(P) < (\sqrt{2}\epsilon + \sqrt{3}\epsilon')\mathbf{Rad}(P). \tag{6}$$

◀

By triangle inequality, Lemmas 8 and 10, we immediately have

$$||\tilde{o} - o|| \leq ||\tilde{o} - o'|| + ||o' - o|| < (2\sqrt{2}\epsilon + \sqrt{3}\epsilon')\mathbf{Rad}(P). \tag{7}$$

This completes the proof of Theorem 7.

5 Sublinear Time Algorithms for MEB under Stability Assumption

Suppose $\epsilon \in (0, 1)$. We assume that the given instance P is an (ϵ^2, β)-stable instance where β is larger than a given lower bound β_0 (*i.e.*, $\beta > \beta_0$). Using Theorem 7, we present two different sublinear time sampling algorithms for computing MEB. Following most of the articles on sublinear time algorithms (*e.g.*, [29, 64, 65]), in each sampling step of our algorithms, we always take the sample **independently and uniformly at random**.

5.1 The First Algorithm

The first algorithm is based on the theory of VC dimension and ϵ-nets [49, 81]. Roughly speaking, we compute an approximate MEB of a small random sample (say, $\mathbb{B}(c, r)$), and expand the ball slightly; then we prove that this expanded ball is an approximate MEB of the whole data set (see Figure 2). Our key idea is to show that $\mathbb{B}(c, r)$ covers at least $(1 - \beta_0)n$ points and therefore c is close to the optimal center by Theorem 7. As emphasized

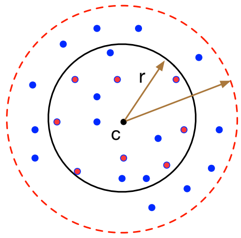

Figure 2 An illustration for the first sampling algorithm. The red points are the samples; we expand $\mathbb{B}(c, r)$ slightly and the larger ball is a radius-approximate MEB of the whole input point set.

in Section 1.1, our result is a single-criterion approximation. If simply applying the uniform sample idea without the stability assumption (as the ideas in [33, 53]), it will result in a bi-criteria approximation where the ball has to cover less than n points for achieving the desired bounded radius.

▶ **Theorem 11.** *With probability $1 - \eta$, Algorithm 1 returns a λ-radius approximate MEB of P, where*

$$\lambda = \frac{\big(1 + (2\sqrt{2} + \sqrt{3})\epsilon\big)(1 + \epsilon^2)}{1 - \epsilon^2} \tag{8}$$

and $\lambda = 1 + O(\epsilon)$ if ϵ is a fixed number in $(0, 1)$.

Algorithm 1 MEB Algorithm I.

Input: Two parameters $0 < \epsilon, \eta < 1$; an (ϵ^2, β)-stable instance P of MEB problem in $\mathbb{R}^d$, where β is larger than a known lower bound $\beta_0 > 0$.

1: Sample a set S of $\Theta(\frac{1}{\beta_0} \cdot \max\{\log \frac{1}{\eta}, d \log \frac{d}{\beta_0}\})$ points from P uniformly at random.

2: Apply any approximate MEB algorithm (such as the core-set based algorithm [16]) to compute a $(1 + \epsilon^2)$-radius approximate MEB of S, and let the obtained ball be $\mathbb{B}(c, r)$.

3: Output the ball $\mathbb{B}\big(c, \frac{1+(2\sqrt{2}+\sqrt{3})\epsilon}{1-\epsilon^2} r\big)$.

Before proving Theorem 11, we prove the following lemma first.

▶ **Lemma 12.** *Let S be a set of $\Theta(\frac{1}{\beta_0} \cdot \max\{\log \frac{1}{\eta}, d \log \frac{d}{\beta_0}\})$ points sampled randomly and independently from a given point set $P \subset \mathbb{R}^d$, and B be any ball covering S. Then, with probability $1 - \eta$, $|B \cap P| \geq (1 - \beta_0)|P|$.*

Proof. Consider the range space $\Sigma = (P, \Phi)$ where each range $\phi \in \Phi$ is the complement of a ball in the space. In a range space, a subset $Y \subset P$ is a β_0-net if for any $\phi \in \Phi$, $\frac{|P \cap \phi|}{|P|} \geq \beta_0 \Longrightarrow Y \cap \phi \neq \emptyset$. Since $|S| = \Theta(\frac{1}{\beta_0} \cdot \max\{\log \frac{1}{\eta}, d \log \frac{d}{\beta_0}\})$, we know that S is a β_0-net of P with probability $1 - \eta$ [49, 81]. Thus, if $|B \cap P| < (1 - \beta_0)|P|$, *i.e.*, $|P \setminus B| > \beta_0 |P|$, we have $S \cap \big(P \setminus B\big) \neq \emptyset$. This contradicts to the fact that S is covered by B. Consequently, $|B \cap P| \geq (1 - \beta_0)|P|$. ◀

Proof of Theorem 11. Denote by o the center of $\mathbf{MEB}(P)$. Since $S \subset P$ and $\mathbb{B}(c, r)$ is a $(1 + \epsilon^2)$-radius approximate MEB of S, we know that $r \leq (1 + \epsilon^2)\mathbf{Rad}(P)$. Moreover, Lemma 12 implies that $|\mathbb{B}(c, r) \cap P| \geq (1 - \beta_0)|P|$ with probability $1 - \eta$. Suppose it is true and let $P' = \mathbb{B}(c, r) \cap P$. Then, we have the distance

$$||c - o|| \leq (2\sqrt{2} + \sqrt{3})\epsilon \mathbf{Rad}(P) \tag{9}$$

via Theorem 7 (we set $\epsilon' = \epsilon$). For simplicity, we use x to denote $(2\sqrt{2}+\sqrt{3})\epsilon$. The inequality (9) implies that the point set P is covered by the ball $\mathbb{B}(c, (1+x)\mathbf{Rad}(P))$. Note that we cannot directly return $\mathbb{B}(c, (1+x)\mathbf{Rad}(P))$ as the final result, since we do not know the value of $\mathbf{Rad}(P)$. Thus, we have to estimate the radius $(1+x)\mathbf{Rad}(P)$.

Since P' is covered by $\mathbb{B}(c, r)$ and $|P'| \geq (1-\beta_0)|P|$, r should be at least $(1-\epsilon^2)\mathbf{Rad}(P)$ due to Definition 3. Hence, we have

$$\frac{1+x}{1-\epsilon^2} r \geq (1+x)\mathbf{Rad}(P). \tag{10}$$

That is, P is covered by the ball $\mathbb{B}(c, \frac{1+x}{1-\epsilon^2} r)$. Moreover, the radius

$$\frac{1+x}{1-\epsilon^2} r \leq \frac{1+x}{1-\epsilon^2}(1+\epsilon^2)\mathbf{Rad}(P). \tag{11}$$

This means that ball $\mathbb{B}(c, \frac{1+x}{1-\epsilon^2} r)$ is a λ-radius approximate MEB of P, where

$$\lambda = (1+\epsilon^2)\frac{1+x}{1-\epsilon^2} = \frac{\big(1+(2\sqrt{2}+\sqrt{3})\epsilon\big)(1+\epsilon^2)}{1-\epsilon^2} \tag{12}$$

and $\lambda = 1 + O(\epsilon)$ if ϵ is a fixed number in $(0, 1)$. ◀

Running time of Algorithm 1. For simplicity, we assume $\log\frac{1}{\eta} < d\log\frac{d}{\beta_0}$. If we use the core-set based algorithm [16] to compute $\mathbb{B}(c, r)$ (see Remark 5), the running time of Algorithm 1 is $O\big(\frac{1}{\epsilon^2}(|S|d + \frac{1}{\epsilon^6}d)\big) = O\big(\frac{d^2}{\epsilon^2\beta_0}\log\frac{d}{\beta_0} + \frac{d}{\epsilon^8}\big) = \tilde{O}(d^2)$ where the hidden factor depends on ϵ and β_0.

▶ Remark 13. If the dimensionality d is too high, the random projection technique *Johnson-Lindenstrauss (JL) transform* [30] can be used to approximately preserve the radius of enclosing ball [2, 56, 79]. However, it is not very useful for reducing the time complexity of Algorithm 1. If we apply the JL-transform on the sampled $\Theta(\frac{d}{\beta_0}\log\frac{d}{\beta_0})$ points in Step 1, the JL-transform step itself already takes $\Omega(\frac{d^2}{\beta_0}\log\frac{d}{\beta_0})$ time.

5.2 The Second Algorithm

Our first algorithm in Section 5.1 is simple, but has a sample size (*i.e.,* the number of sampled points) depending on the dimensionality d, while **the second algorithm has a sample size independent of both n and d** (it is particularly important when a kernel function is applied, because the new dimension could be very large or even $+\infty$). We briefly overview our idea first.

High level idea of the second algorithm. Recall our Remark 5 (ii). If we know the value of $(1+\epsilon)\mathbf{Rad}(P)$, we can perform almost the same core-set construction procedure described in Theorem 4 to achieve an approximate center of $\mathbf{MEB}(P)$, where the only difference is that we add a point with distance at least $(1+\epsilon)\mathbf{Rad}(P)$ to o_i in each iteration. In this way, we avoid selecting the farthest point to o_i, since this operation will inevitably have a linear time complexity. To implement our strategy in sublinear time, we need to determine the value of $(1+\epsilon)\mathbf{Rad}(P)$ first. We propose Lemma 14 below to estimate the range of $\mathbf{Rad}(P)$, and then perform a binary search on the range to determine the value of $(1+\epsilon)\mathbf{Rad}(P)$ approximately. Based on the stability property, we observe that the core-set construction procedure can serve as an "oracle" to help us to guess the value of $(1+\epsilon)\mathbf{Rad}(P)$ (see Algorithm 3). Let $h > 0$ be a candidate. We add a point with distance at least h to o_i in each iteration. We prove that the procedure cannot continue for more than z iterations if $h \geq (1+\epsilon)\mathbf{Rad}(P)$, and will continue more than z iterations with constant probability if $h < (1-\epsilon)\mathbf{Rad}(P)$, where z is the size of core-set described in Theorem 4. Also, during the core-set construction, we add the points to the core-set via random sampling, rather than a deterministic way. A minor issue here is that we need to replace ϵ by ϵ^2 in Theorem 4, so as to achieve the overall $(1+O(\epsilon))$-radius approximation in the following analysis. Below, we introduce Lemma 14 and Theorem 16 first, and then present the main result in Theorem 17.

▶ **Lemma 14.** *Given a parameter $\eta \in (0,1)$, one selects an arbitrary point $p_1 \in P$ and takes a sample $Q \subset P$ with $|Q| = \frac{1}{\beta_0}\log\frac{1}{\eta}$ uniformly at random. Let $p_2 = \arg\max_{p\in Q}||p-p_1||$. Then, with probability $1-\eta$,*

$$\mathbf{Rad}(P) \in [\frac{1}{2}||p_1-p_2||, \frac{1}{1-\epsilon^2}||p_1-p_2||]. \tag{13}$$

Proof. First, the lower bound of $\mathbf{Rad}(P)$ is obvious since $||p_1-p_2||$ is always no larger than $2\mathbf{Rad}(P)$. Then, we consider the upper bound. Let $\mathbb{B}(p_1,l)$ be the ball covering exactly $(1-\beta_0)n$ points of P, and thus $l \geq (1-\epsilon^2)\mathbf{Rad}(P)$ according to Definition 3. To complete our proof, we also need the following folklore lemma presented in [32].

▶ **Lemma 15** ([32]). *Let N be a set of elements, and N' be a subset of N with size $|N'| = \tau|N|$ for some $\tau \in (0,1)$. Given $\eta \in (0,1)$, if one randomly samples $\frac{\ln 1/\eta}{\ln 1/(1-\tau)} \leq \frac{1}{\tau}\ln\frac{1}{\eta}$ elements from N, then with probability at least $1-\eta$, the sample contains at least one element of N'.*

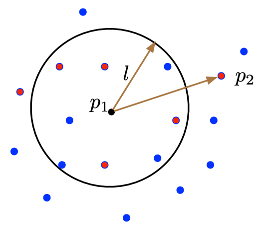

Figure 3 An illustration of Lemma 14; the red points are the sampled set Q.

In Lemma 15, let N and N' be the point set P and the subset $P \setminus \mathbb{B}(p_1,l)$, respectively. We know that Q contains at least one point from N' according to Lemma 15 (by setting $\tau = \beta_0$). Namely, Q contains at least one point outside $\mathbb{B}(p_1,l)$. Moreover, because $p_2 = \arg\max_{p\in Q}||p-p_1||$, we have $||p_1-p_2|| \geq l \geq (1-\epsilon^2)\mathbf{Rad}(P)$, *i.e.*, $\mathbf{Rad}(P) \leq \frac{1}{1-\epsilon^2}||p_1-p_2||$ (see Figure 3 for an illustration). ◀

Lemma 14 immediately implies the following result.

▶ **Theorem 16.** *In Lemma 14, the ball $\mathbb{B}(p_1, \frac{2}{1-\epsilon^2}||p_1-p_2||)$ is a $\frac{4}{1-\epsilon^2}$-radius approximate MEB of P, with probability $1-\eta$.*

Proof. From the upper bound in Lemma 14, we know that $\frac{2}{1-\epsilon^2}||p_1-p_2|| \geq 2\mathbf{Rad}(P)$. Since $||p_1-p|| \leq 2\mathbf{Rad}(P)$ for any $p \in P$, the ball $\mathbb{B}(p_1, \frac{2}{1-\epsilon^2}||p_1-p_2||)$ covers the whole point set P. From the lower bound in Lemma 14, we know that $\frac{2}{1-\epsilon^2}||p_1-p_2|| \leq \frac{4}{1-\epsilon^2}\mathbf{Rad}(P)$. Therefore, it is a $\frac{4}{1-\epsilon^2}$-radius approximate MEB of P. ◀

Since $|Q| = \frac{1}{\beta_0}\log\frac{1}{\eta}$ in Lemma 14, Theorem 16 indicates that we can easily obtain a $\frac{4}{1-\epsilon^2}$-radius approximate MEB of P in $O(\frac{1}{\beta_0}(\log\frac{1}{\eta})d)$ time. Below, we present our second sampling algorithm (Algorithm 2) that can achieve a much lower $(1+O(\epsilon))$-approximation ratio. Algorithm 3 serves as a subroutine in Algorithm 2. In Algorithm 3, we simply set $z = \frac{3}{\epsilon^2}$ with $s = 1/3$ as described in Theorem 4 (as mentioned before, we replace ϵ by ϵ^2); we compute o_i having distance less than $s\frac{\epsilon^2}{1+\epsilon^2}\mathbf{Rad}(T)$ to the center of $\mathbf{MEB}(T)$ in Step 2(1).

▶ **Theorem 17.** *With probability $1 - \eta_0$, Algorithm 2 returns a λ-radius approximate MEB of P, where*

$$\lambda = \frac{(1+x_1)(1+x_2)}{1+\epsilon^2} \quad \text{with} \quad x_1 = O(\frac{\epsilon^2}{1-\epsilon^2}), x_2 = O(\frac{\epsilon}{\sqrt{1-\epsilon^2}}), \tag{14}$$

and $\lambda = 1 + O(\epsilon)$ if ϵ is a fixed number in $(0, 1)$. The running time is $\tilde{O}\big((\frac{1}{\epsilon^2\beta_0} + \frac{1}{\epsilon^8})d\big)$, where $\tilde{O}(f) = O(f \cdot \mathtt{polylog}(\frac{1}{\epsilon}, \frac{1}{\eta_0}))$.

Algorithm 2 MEB Algorithm II.

Input: Two parameters $0 < \epsilon, \eta_0 < 1$; an (ϵ^2, β)-stable instance P of MEB problem in $\mathbb{R}^d$, where β is larger than a given lower bound $\beta_0 > 0$. Set the interval $[a, b]$ for $\mathbf{Rad}(P)$ that is obtained by Lemma 14.

1: Among the set $\{(1-\epsilon^2)a, (1+\epsilon^2)(1-\epsilon^2)a, \cdots, (1+\epsilon^2)^w(1-\epsilon^2)a = (1+\epsilon^2)b\}$ where $w = \lceil \log_{1+\epsilon^2} \frac{2}{(1-\epsilon^2)^2} \rceil + 1 = O(\frac{1}{\epsilon^2})$, perform binary search for the value h by using Algorithm 3 with $z = \frac{3}{\epsilon^2}$ and $\eta = \frac{\eta_0}{2 \log w}$.

2: Suppose that Algorithm 3 returns “no” when $h = (1+\epsilon^2)^{i_0}(1-\epsilon^2)a$ and returns “yes” when $h = (1+\epsilon^2)^{i_0+1}(1-\epsilon^2)a$.

3: Run Algorithm 3 again with $h = (1+\epsilon^2)^{i_0+2}a$, $z = \frac{3}{\epsilon^2}$, and $\eta = \eta_0/2$; let $\tilde{o}$ be the obtained ball center of T when the loop stops.

4: Return the ball $\mathbb{B}(\tilde{o}, r)$, where $r = \frac{1+(2\sqrt{2}+\frac{2\sqrt{6}}{\sqrt{1-\epsilon^2}})\epsilon}{1+\epsilon^2}h$.

Algorithm 3 Oracle for testing h.

Input: An instance P, a parameter $\eta \in (0, 1)$, $h > 0$, and a positive integer z.

1: Initially, arbitrarily select a point $p \in P$ and let $T = \{p\}$.

2: $i = 1$; repeat the following steps:
- **(1)** Compute an approximate MEB of T and let the ball center be o_i as described in Theorem 4 (replace ϵ by ϵ^2 and set $s = 1/3$).
- **(2)** Sample a set $Q \subset P$ with $|Q| = \frac{1}{\beta_0}\log\frac{z}{\eta}$ uniformly at random.
- **(3)** Select the point $q \in Q$ that is farthest to o_i, and add it to T.
- **(4)** If $||q - o_i|| < h$, stop the loop and output “yes”.
- **(5)** $i = i + 1$; if $i > z$, stop the loop and output “no”.

Before proving Theorem 17, we provide Lemma 18 first.

▶ **Lemma 18.** *If $h \geq (1+\epsilon^2)\mathbf{Rad}(P)$, Algorithm 3 returns “yes”; else if $h < (1-\epsilon^2)\mathbf{Rad}(P)$, Algorithm 3 returns “no” with probability at least $1 - \eta$.*

Proof. First, we assume that $h \geq (1+\epsilon^2)\mathbf{Rad}(P)$. Recall the remark following Theorem 4. If we always add a point q with distance at least $h \geq (1+\epsilon^2)\mathbf{Rad}(P)$ to o_i, the loop 2(1)-(5) cannot continue more than z iterations, *i.e.*, Algorithm 3 will return “yes”.

Now, we consider the case $h < (1-\epsilon^2)\mathbf{Rad}(P)$. Similar to the proof of Lemma 14, we consider the ball $\mathbb{B}(o_i, l)$ covering exactly $(1-\beta_0)n$ points of P. According to Definition 3, we know that $l \geq (1-\epsilon^2)\mathbf{Rad}(P) > h$. Also, with probability $1 - \eta/z$, the sample Q contains at least one point outside $\mathbb{B}(o_i, l)$ due to Lemma 15. By taking the union bound, with probability $(1 - \eta/z)^z \geq 1 - \eta$, $||q - o_i||$ is always larger than h and eventually Algorithm 3 will return "no". ◀

Proof of Theorem 17. Since Algorithm 3 returns "no" when $h = (1+\epsilon^2)^{i_0}(1-\epsilon^2)a$ and returns "yes" when $h = (1+\epsilon^2)^{i_0+1}(1-\epsilon^2)a$, from Lemma 18 we know that

$$(1+\epsilon^2)^{i_0}(1-\epsilon^2)a < (1+\epsilon^2)\mathbf{Rad}(P); \tag{15}$$

$$(1+\epsilon^2)^{i_0+1}(1-\epsilon^2)a \geq (1-\epsilon^2)\mathbf{Rad}(P). \tag{16}$$

The above inequalities together imply that

$$\frac{(1+\epsilon^2)^3}{1-\epsilon^2}\mathbf{Rad}(P) > (1+\epsilon^2)^{i_0+2}a \geq (1+\epsilon^2)\mathbf{Rad}(P). \tag{17}$$

Thus, when running Algorithm 3 with $h = (1+\epsilon^2)^{i_0+2}a$ in Step 3, the algorithm returns "yes" (by the right hand-side of (17)). Then, consider the ball $\mathbb{B}(\tilde{o}, h)$. We claim that $|P \setminus \mathbb{B}(\tilde{o}, h)| < \beta_0 n$. Otherwise, the sample Q contains at least one point outside $\mathbb{B}(\tilde{o}, h)$ with probability $1 - \eta/z$ in Step 2(2) of Algorithm 3, *i.e.,* the loop will continue. Thus, it contradicts to the fact that the algorithm returns "yes". Let $P' = P \cap \mathbb{B}(\tilde{o}, h)$, and then $|P'| \geq (1-\beta_0)n$. Moreover, the left hand-side of (17) indicates that

$$h = (1+\epsilon^2)^{i_0+2}a < (1 + \frac{8\epsilon^2}{1-\epsilon^2})\mathbf{Rad}(P). \tag{18}$$

Now, we can apply Theorem 7, where we set "ϵ'" to be "$\sqrt{\frac{8\epsilon^2}{1-\epsilon^2}}$" in the theorem. Let o be the center of $\mathbf{MEB}(P)$. Consequently, we have

$$||\tilde{o} - o|| < (2\sqrt{2} + 2\sqrt{6}/\sqrt{1-\epsilon^2})\epsilon \cdot \mathbf{Rad}(P). \tag{19}$$

For simplicity, we let $x_1 = \frac{8\epsilon^2}{1-\epsilon^2}$ and $x_2 = (2\sqrt{2} + 2\sqrt{6}/\sqrt{1-\epsilon^2})\epsilon$. Hence, $h \leq (1 + x_1)\mathbf{Rad}(P)$ and $||\tilde{o} - o|| \leq x_2\mathbf{Rad}(P)$ in (18) and (19). From (19), we know that $P \subset \mathbb{B}(\tilde{o}, (1+x_2)\mathbf{Rad}(P))$. From the right hand-side of (17), we know that $(1+x_2)\mathbf{Rad}(P) \leq \frac{1+x_2}{1+\epsilon^2}h$. Thus, we have $P \subset \mathbb{B}\Big(\tilde{o}, \frac{1+x_2}{1+\epsilon^2}h\Big)$ where $\frac{1+x_2}{1+\epsilon^2}h = \frac{1+(2\sqrt{2}+\frac{2\sqrt{6}}{\sqrt{1-\epsilon^2}})\epsilon}{1+\epsilon^2}h$. Also, the radius

$$\frac{1+x_2}{1+\epsilon^2}h \underbrace{\leq}_{\text{by (18)}} \frac{(1+x_2)(1+x_1)}{1+\epsilon^2}\mathbf{Rad}(P) = \lambda \cdot \mathbf{Rad}(P). \tag{20}$$

Thus $\mathbb{B}\Big(\tilde{o}, \frac{1+x_2}{1+\epsilon^2}h\Big)$ is a λ-radius approximate MEB of P, and $\lambda = 1 + O(\epsilon)$ if ϵ is fixed.

Success probability. The success probability of Algorithm 3 is $1-\eta$. In Algorithm 2, we set $\eta = \frac{\eta_0}{2\log w}$ in Step 1 and $\eta = \eta_0/2$ in Step 3, respectively. We take the union bound and the success probability of Algorithm 2 is $(1 - \frac{\eta_0}{2\log w})^{\log w}(1-\eta_0/2) > 1 - \eta_0$.

Running time. As the subroutine, Algorithm 3 runs in $O(z(\frac{1}{\beta_0}(\log\frac{z}{\eta})d + \frac{1}{\epsilon^6}d))$ time; Algorithm 2 calls the subroutine $O\big(\log(\frac{1}{\epsilon^2})\big)$ times. Note that $z = O(\frac{1}{\epsilon^2})$. Thus, the total running time is $\tilde{O}\big((\frac{1}{\epsilon^2\beta_0} + \frac{1}{\epsilon^8})d\big)$. ◀

6 Extensions

We also present two important extensions in this paper (due to the space limit, we place the details to our full paper). We briefly introduce the main ideas and summarize the results below.

We first consider MEB with outliers under the stability assumption and provide a sublinear time constant factor radius approximation. We also consider the general case without the stability assumption. An interesting observation is that the ideas developed for stable instance can even help us to develop a hybrid approach for MEB (without or with outliers) when the stability assumption does not hold. First, we "suppose" the input instance is (α, β)-stable where "α" and "β" are carefully designed based on the pre-specified radius error bound ϵ and covering error bound δ, and compute a "potential" $(1+\epsilon)$-radius approximation (say a ball B_1); then we apply the recently proposed sublinear time bi-criteria MEB with outliers algorithm [31] to compute a "potential" $(1-\delta)$-covering approximation (say a ball B_2); finally, we determine the final output based on the ratio of their radii. Specifically, we set a threshold τ that is determined by the given radius error bound ϵ. If the ratio is no larger than τ, we know that B_1 is a "true" $(1+\epsilon)$-radius approximation and return it; otherwise, we return B_2 that is a "true" $(1-\delta)$-covering approximation. Moreover, for the latter case (*i.e.,* returning a $(1-\delta)$-covering approximation), we will show that our proposed algorithm yields a radius not only being strictly smaller than $\mathbf{Rad}(P)$, but also having a gap of $\Theta(\epsilon^2) \cdot \mathbf{Rad}(P)$ to $\mathbf{Rad}(P)$ (*i.e.,* the returned radius is at most $\big(1-\Theta(\epsilon^2)\big) \cdot \mathbf{Rad}(P)$). Our algorithm only needs uniform sampling and a single pass over the input data, where the space complexity in memory is $O(d)$ (the hidden factor depends on ϵ and δ); if the input data matrix is sparse (*i.e.,* $M = o(nd)$), the time complexity is sublinear. Furthermore, we propose the similar results for the polytope distance and SVM problems (for both stable instance and general instance).

7 Future Work

Following our work, several interesting problems deserve to be studied in future. For example, different from radius approximation, the current research on covering approximation of MEB is still inadequate. In particular, can we provide a lower bound for the complexity of computing covering approximate MEB, as the lower bound result for radius approximate MEB proved by [25]? Also, is it possible to extend the stability notion to other geometric optimization problems with more complicated structures (like subspace fitting and clustering [36], and regression problems [67])?

References

1. Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52:1–30, 2005.
2. Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Embeddings of surfaces, curves, and moving points in euclidean space. In *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8, 2007*, pages 381–389, 2007.
3. Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. *Discrete & Computational Geometry*, 39(1-3):38–58, 2008.
4. Pankaj K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015.
5. Alok Aggarwal, Hiroshi Imai, Naoki Katoh, and Subhash Suri. Finding k points with minimum diameter and related problems. *Journal of algorithms*, 12(1):38–56, 1991.

6 Zeyuan Allen Zhu, Zhenyu Liao, and Yang Yuan. Optimization algorithms for faster computational geometry. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 53:1–53:6, 2016.

7 Noga Alon, Seannie Dar, Michal Parnas, and Dana Ron. Testing of clustering. *SIAM Journal on Discrete Mathematics*, 16(3):393–417, 2003.

8 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Stability yields a PTAS for k-median and k-means clustering. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 309–318, 2010.

9 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Center-based clustering under perturbation stability. *Inf. Process. Lett.*, 112(1-2):49–54, 2012.

10 Maria-Florina Balcan, Avrim Blum, and Anupam Gupta. Clustering under approximation stability. *Journal of the ACM (JACM)*, 60(2):8, 2013.

11 Maria-Florina Balcan and Mark Braverman. Finding low error clusterings. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.

12 Maria-Florina Balcan, Nika Haghtalab, and Colin White. k-center clustering under perturbation resilience. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 68:1–68:14, 2016.

13 Maria-Florina Balcan and Yingyu Liang. Clustering under perturbation resilience. *SIAM J. Comput.*, 45(1):102–155, 2016.

14 Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Oper. Res.*, 52(1):35–53, 2004.

15 Yonatan Bilu and Nathan Linial. Are stable instances easy? *Combinatorics, Probability & Computing*, 21(5):643–660, 2012.

16 Mihai Bădoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2003.

17 Mihai Bădoiu and Kenneth L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, 2008.

18 Mihai Bădoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 250–257, 2002.

19 Giuseppe Carlo Calafiore and Marco C. Campi. Uncertain convex programs: randomized solutions and confidence levels. *Math. Program.*, 102(1):25–46, 2005.

20 Matteo Ceccarello, Andrea Pietracaprina, and Geppino Pucci. Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *PVLDB*, 12(7):766–778, 2019.

21 Timothy M. Chan and Vinayak Pathak. Streaming and dynamic algorithms for minimum enclosing balls in high dimensions. *Comput. Geom.*, 47(2):240–247, 2014.

22 Moses Charikar, Samir Khuller, David M Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 642–651. Society for Industrial and Applied Mathematics, 2001.

23 Moses Charikar, Liadan O'Callaghan, and Rina Panigrahy. Better streaming algorithms for clustering problems. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 30–39. ACM, 2003.

24 Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):63, 2010.

25 Kenneth L. Clarkson, Elad Hazan, and David P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5):23:1–23:49, 2012.

26 Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20:273, 1995.

27 David J. Crisp and Christopher J. C. Burges. A geometric interpretation of v-SVM classifiers. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *NIPS*, pages 244–250. The MIT Press, 1999.

28 Artur Czumaj and Christian Sohler. Sublinear-time algorithms. *Survey paper*, 2004.

29 Artur Czumaj and Christian Sohler. Sublinear-time approximation for clustering via random sampling. In *International Colloquium on Automata, Languages, and Programming*, pages 396–407. Springer, 2004.

30 Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

31 Hu Ding. A sub-linear time framework for geometric optimization with outliers in high dimensions. In Fabrizio Grandoni, Grzegorz Herman, and Peter Sanders, editors, *28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference)*, volume 173 of *LIPIcs*, pages 38:1–38:21, 2020.

32 Hu Ding and Jinhui Xu. Sub-linear time hybrid approximations for least trimmed squares estimator and related problems. In *Proceedings of the International Symposium on Computational geometry (SoCG)*, page 110, 2014.

33 Hu Ding, Haikuo Yu, and Zixiu Wang. Greedy strategy works for k-center clustering with outliers and coreset construction. In *27th Annual European Symposium on Algorithms, ESA 2019, September 9-11, 2019, Munich/Garching, Germany.*, pages 40:1–40:16, 2019.

34 Alon Efrat, Micha Sharir, and Alon Ziv. Computing the smallest k-enclosing circle and related problems. *Computational Geometry*, 4(3):119–136, 1994.

35 Dan Feldman. Core-sets: An updated survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 10(1), 2020.

36 Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC 2011, San Jose, CA, USA, 6-8 June 2011*, pages 569–578. ACM, 2011.

37 Dan Feldman, Chongyuan Xiang, Ruihao Zhu, and Daniela Rus. Coresets for differentially private k-means clustering and applications to privacy in mobile sensor networks. In *Proceedings of the 16th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2017, Pittsburgh, PA, USA, April 18-21, 2017*, pages 3–15, 2017.

38 Kaspar Fischer, Bernd Gärtner, and Martin Kutz. Fast smallest-enclosing-ball computation in high dimensions. In *Algorithms - ESA 2003, 11th Annual European Symposium, Budapest, Hungary, September 16-19, 2003, Proceedings*, pages 630–641, 2003.

39 Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

40 Dan Garber and Elad Hazan. Approximating semidefinite programs in sublinear time. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1080–1088, 2011.

41 Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. In *Proceedings of the International Symposium on Computational geometry (SoCG)*, pages 33–42, 2009.

42 Elmer G. Gilbert. An iterative procedure for computing the minimum of a quadratic form on a convex set. *SIAM Journal on Control*, 4(1):61–80, 1966.

43 Ashish Goel, Piotr Indyk, and Kasturi R Varadarajan. Reductions among high dimensional proximity problems. In *SODA*, volume 1, pages 769–778. Citeseer, 2001.

44 Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

45 Teofilo F Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

46 Laszlo Gyongyosi and Sandor Imre. Geometrical analysis of physically allowed quantum cloning transformations for quantum cryptography. *Information Sciences*, 285:1–23, 2014.

47 Sariel Har-Peled and Soham Mazumdar. Fast algorithms for computing the smallest k-enclosing circle. *Algorithmica*, 41(3):147–157, 2005.

48 Sariel Har-Peled and Yusu Wang. Shape fitting with outliers. *SIAM Journal on Computing*, 33(2):269–285, 2004.

49 David Haussler and Emo Welzl. eps-nets and simplex range queries. *Discrete & Computational Geometry*, 2(2):127–151, 1987.

50 Kohei Hayashi and Yuichi Yoshida. Minimizing quadratic functions in constant time. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 2217–2225, 2016.

51 Elad Hazan, Tomer Koren, and Nati Srebro. Beating SGD: learning svms in sublinear time. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, pages 1233–1241, 2011.

52 Dorit S Hochbaum and David B Shmoys. A best possible heuristic for the k-center problem. *Mathematics of operations research*, 10(2):180–184, 1985.

53 Lingxiao Huang, Shaofeng Jiang, Jian Li, and Xuan Wu. Epsilon-coresets for clustering (with outliers) in doubling metrics. In *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 814–825, 2018.

54 Piotr Indyk. Sublinear time algorithms for metric space problems. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 428–434, 1999.

55 Piotr Indyk. A sublinear time approximation scheme for clustering in metric spaces. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 154–159, 1999.

56 Michael Kerber and Sharath Raghvendra. Approximation and streaming algorithms for projective clustering via random projections. In *Proceedings of the 27th Canadian Conference on Computational Geometry, CCCG 2015, Kingston, Ontario, Canada, August 10-12, 2015*, 2015.

57 Michael Kerber and R. Sharathkumar. Approximate čech complex in low and high dimensions. In *Algorithms and Computation - 24th International Symposium, ISAAC 2013, Hong Kong, China, December 16-18, 2013, Proceedings*, pages 666–676, 2013.

58 Amer Krivosija and Alexander Munteanu. Probabilistic smallest enclosing ball in high dimensions via subgradient sampling. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG 2019, June 18-21, 2019, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 47:1–47:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

59 Amit Kumar and Ravindran Kannan. Clustering with spectral norm and the k-means algorithm. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 299–308. IEEE, 2010.

60 Piyush Kumar, Joseph S. B. Mitchell, and E. Alper Yildirim. Approximate minimum enclosing balls in high dimensions using core-sets. *ACM Journal of Experimental Algorithmics*, 8, 2003.

61 Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

62 Jiří Matoušek. On enclosing k points by a circle. *Information Processing Letters*, 53(4):217–221, 1995.

63 Richard Matthew McCutchen and Samir Khuller. Streaming algorithms for k-center clustering with outliers and with anonymity. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 165–178. Springer, 2008.

64 Adam Meyerson, Liadan O'callaghan, and Serge Plotkin. A k-median algorithm with running time independent of data size. *Machine Learning*, 56(1-3):61–87, 2004.

65 Nina Mishra, Dan Oblinger, and Leonard Pitt. Sublinear time approximate clustering. In *Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 439–447. Society for Industrial and Applied Mathematics, 2001.

66 Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, USA, 1995.

67 David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. On the least trimmed squares estimator. *Algorithmica*, 69(1):148–183, 2014.

68 Frank Nielsen and Richard Nock. Approximating smallest enclosing balls with applications to machine learning. *Int. J. Comput. Geom. Appl.*, 19(5):389–414, 2009.

69 Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Locating a small cluster privately. In *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016, San Francisco, CA, USA, June 26 - July 01, 2016*, pages 413–427, 2016.

70 Rafail Ostrovsky, Yuval Rabani, Leonard J Schulman, and Chaitanya Swamy. The effectiveness of lloyd-type methods for the k-means problem. *Journal of the ACM (JACM)*, 59(6):28, 2012.

71 Rina Panigrahy. Minimum enclosing polytope in high dimensions. *arXiv preprint cs/0407020*, 2004.

72 Jeff M. Phillips. Coresets and sketches. *Computing Research Repository*, 2016.

73 J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods — Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

74 Tim Roughgarden. Beyond worst-case analysis. *Commun. ACM*, 62(3):88–96, 2019.

75 Ronitt Rubinfeld. Sublinear time algorithms. *Citeseer*, 2006.

76 Ankan Saha, S. V. N. Vishwanathan, and Xinhua Zhang. New approximation algorithms for minimum enclosing convex shapes. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1146–1160, 2011.

77 B. Scholkopf, A. J. Smola, K. R. Muller, and P. L. Bartlett. New support vector algorithms. *Neural Computation*, 12:1207–1245, 2000.

78 Bernhard Schölkopf and Alexander Johannes Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.

79 Donald R. Sheehy. The persistent homology of distance functions under random projection. In *30th Annual Symposium on Computational Geometry, SOCG'14, Kyoto, Japan, June 08 - 11, 2014*, page 328, 2014.

80 Ivor W. Tsang, James T. Kwok, and Pak-Ming Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

81 Vladimir N Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. In *Measures of complexity*, pages 11–30. Springer, 2015.

82 Hamid Zarrabi-Zadeh and Asish Mukhopadhyay. Streaming 1-center with outliers in high dimensions. In *Proceedings of the Canadian Conference on Computational Geometry (CCCG)*, pages 83–86, 2009.