A Sub-linear Time Framework for Geometric Optimization with Outliers in High Dimensions

Hu Ding
School of Computer Science and Engineering, University of Science and Technology of China
He Fei, China
huding@ustc.edu.cn

Abstract. Many real-world problems can be formulated as geometric optimization problems in high dimensions, especially in the fields of machine learning and data mining. Moreover, we often need to take into account of outliers when optimizing the objective functions. However, the presence of outliers could make the problems to be much more challenging than their vanilla versions. In this paper, we study the fundamental minimum enclosing ball (MEB) with outliers problem first; partly inspired by the core-set method from Bădiu and Clarkson, we propose a sub-linear time bi-criteria approximation algorithm based on two novel techniques, the Uniform-Adaptive Sampling method and Sandwich Lemma. To the best of our knowledge, our result is the first sub-linear time algorithm, which has the sample size (i.e., the number of sampled points) independent of both the number of input points $n$ and dimensionality $d$, for MEB with outliers in high dimensions. Furthermore, we observe that these two techniques can be generalized to deal with a broader range of geometric optimization problems with outliers in high dimensions, including flat fitting, $k$-center clustering, and SVM with outliers, and therefore achieve the sub-linear time algorithms for these problems respectively.
1 Introduction

Geometric optimization is a fundamental topic that has been extensively studied in the community of computational geometry [1]. The minimum enclosing ball (MEB) problem is one of the most popular geometric optimization problems who has attracted a lot of attentions in past years, where the goal is to compute the smallest ball covering a given set of points in the Euclidean space $\mathbb{R}^d$ [11,31,50]. Though its formulation is very simple, MEB has a number of applications in real world, such as classification [20,21,69], preserving privacy [30,60], and quantum cryptography [39]. A more general geometric optimization problem is called flat fitting that is to compute the smallest slab (centered at a low-dimensional flat) to cover the input data [12,61,73]. Another closely related important topic is the k-center clustering problem, where the goal is to find $k > 1$ balls to cover the given input data and minimize the maximum radius of the balls [36]; the problem has been widely applied to many areas, such as facility location [18] and data analysis [68]. Moreover, some geometric optimization problems are trying to maximize their size functions. As an example, the well known classification technique support vector machine (SVM) [17] is to maximum the margin separating two differently labeled point sets in the space.

Real-world datasets are often noisy and contain outliers. Moreover, outliers could seriously affect the final optimization results. For example, it is easy to see that even one outlier could make the MEB arbitrarily large. In particular, as the rapid development of machine learning, the field of adversarial machine learning concerning about the potential vulnerabilities of the algorithms has attracted a great amount of attentions [9,37,46,51]. A small set of outliers could be added by some adversarial attacker to make the decision boundary severely deviate and cause unexpected mis-classification [8,48]. Furthermore, the presence of outliers often results in a quite challenging combinatorial optimization problem; as an example, if $m$ of the input $n$ data items are outliers ($m < n$), we have to consider an exponentially large number $\binom{n}{m}$ of different possible cases when optimizing the objective function. Therefore, the design of efficient and robust optimization algorithms is urgently needed to meet these challenges.

1.1 Our Contributions

In big data era, the data size could be so large that we cannot even afford to read the whole dataset once. In this paper, we consider to develop sub-linear time algorithms for several geometric optimization problems involving outliers. We study the aforementioned MEB with outliers problem first. Informally speaking, given a set of $n$ points in $\mathbb{R}^d$ and a small parameter $\gamma \in (0, 1)$, the problem is to find the smallest ball covering at least $(1 - \gamma)n$ points from the input. We are aware of several existing sub-linear time bi-criteria approximation algorithms based on uniform sampling for MEB and $k$-center clustering with outliers [7,28,45], where the “bi-criteria” means that the ball (or the union of the $k$ balls) is allowed to exclude a little more points than the pre-specified number of outliers. Their ideas are based on the theory of VC dimension [71]. But the sample size usually depends on the dimensionality $d$, which is roughly $O\left(\frac{1}{\gamma^3}kd \cdot \text{polylog}(\frac{1}{\gamma})\right)$, if allowing to discard $(1 + \delta)\gamma n$ outliers with $\delta \in (0, 1)$ ($k = 1$ in the complexity for the MEB with outliers problem). A detailed overview on previous works is shown in Section 1.2

Since many optimization problems in practice need to consider high-dimensional datasets, especially in the fields of machine learning and data mining, the above sample size from [7,28,45] could be very large. Partly inspired by the core-set method from Bádoiu and Clarkson [11] for computing MEB in high dimensions, we are wondering that whether it is possible to remove the dependency on $d$ in the sample size for MEB with outliers and other related high dimensional geometric optimization problems. Given a parameter $\epsilon \in (0, 1)$, the method of [11] is a simple greedy algorithm that selects $\frac{2}{\epsilon^2}$ points (as the core-set) for constructing a $(1 + \epsilon)$-approximate MEB, where the resulting radius is at most $1 + \epsilon$ times the optimal one. A highlight of their method is that the core-set size $\frac{2}{\epsilon^2}$ is independent of $d$. However, there are several substantial challenges when applying their method to design sub-linear time algorithm for MEB with outliers. First, we need to implement the “greedy selection” step by a random sampling manner, but it is challenging to guarantee the resulting quality especially when the data is mixed with outliers. Second, the random sampling approach often yields a set of candidates for the ball center (e.g., we may need to repeatedly run the algorithm multiple times for boosting the success probability, and each time generates a candidate solution), and thus it is necessary to design an efficient strategy to determine which candidate is the best one in sub-linear time.

To tackle these challenges, we propose two key techniques, the novel “Uniform-Adaptive Sampling” method and “Sandwich Lemma”. Roughly speaking, the Uniform-Adaptive Sampling method can help us to bound the error induced in each “randomized greedy selection” step; the Sandwich Lemma enables us to estimate the objective value of each candidate and select the best one in sub-linear time. To the best of our knowledge, our result is the first sub-linear time approximation algorithm for the MEB with...
outliers problem with sample size being independent of the number of points \( n \) and the dimensionality \( d \), which significantly improves the time complexities of existing algorithms.

Moreover, we observe that our proposed techniques can be used to solve a broader range of geometric optimization problems. We define a general optimization problem called minimum enclosing “x” (MEX) with Outliers, where the “x” could be a specified kind of shape (e.g., the shape is a ball for MEB with outliers). We prove that it is able to generalize the Uniform-Adaptive Sampling method and Sandwich Lemma to adapt the shape “x”, as long as it satisfies several properties. In particular we focus on the MEX with outlier problems including flat fitting, \( k \)-center clustering, and SVM with outliers; a common characteristic of these problems is that each of them has an iterative algorithm based on greedy selection for its vanilla version (without outliers) that is similar to the MEB algorithm of \([11]\). Though these problems have been widely studied before, the research in terms of their sub-linear time algorithms is till quite limited.

1.2 Related Work

Sub-linear time algorithms. It has a long history of the research on sub-linear time algorithms design in theory \([21]\). For example, a number of sub-linear time clustering algorithms have been studied in \([25, 47, 56, 58]\). Another important application of sub-linear time algorithms is property testing on graphs or probability distributions \([35]\).

As mentioned before, the uniform sampling idea can be used to design sub-linear time algorithms for the problems of MEB and \( k \)-center clustering with outliers \([7, 28, 52]\), but the sample size depends on the dimensionality \( d \) that could be very large in practice. Note that Alon et al. \([7]\) presented another sub-linear time algorithm, which has the sample size independent of \( d \), to test whether an input point set can be covered by a ball with a given radius; however, it is difficult to apply their method to solve the MEB with outliers problem as the algorithm relies on some nice properties of minimum enclosing ball, but these properties are not easy to be utilized when inliers and outliers are mixed. In \([26]\), we proposed a notion of stability for MEB and developed the sub-linear time MEB algorithms for stable instance. Clarkson et al. \([21]\) developed an elegant perceptron framework for solving several optimization problems arising in machine learning, such as MEB. For a set of \( n \) points in \( \mathbb{R}^d \), their framework can solve the MEB problem in \( O \left( \frac{n^2}{\epsilon^6} + \frac{\sqrt{d}}{\epsilon} \right) \) time. Based on a stochastic primal-dual approach, Hazan et al. \([44]\) provided an algorithm for solving the SVM problem in sub-linear time.

MEB and \( k \)-center clustering with outliers. Core-set is a popular technique to reduce the time complexities for many optimization problems \([2, 62]\). The core-set idea has also been used to compute approximate MEB in high dimensional space \([13, 20, 49, 50, 61]\). Bădoiu and Clarkson \([11]\) showed that it is possible to find a core-set of size \([2/\epsilon]\) that yields a \((1+\epsilon)\)-approximate MEB. There are also several exact and approximation algorithms for MEB that do not rely on core-sets \([6, 31, 65]\). Streaming algorithms for computing MEB were also studied before \([4, 16]\).

Bădoiu et al. \([13]\) 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). Several algorithms for the low dimensional MEB with outliers problem have also been developed \([5, 29, 40, 54]\). A 3-approximation algorithm for \( k \)-center clustering with outliers in arbitrary metrics was proposed by Charikar et al. \([18]\). Chakrabarty et al. \([15]\) proposed a 2-approximation algorithm for \( k \)-center clustering with outliers. These algorithms often have high time complexities (e.g., \( O(n^2d) \)). Recently, Ding et al. \([28]\) provided a linear time greedy algorithm for \( k \)-center clustering with outliers based on the idea of the Gonzalez’s algorithm \([36]\). Furthermore, there exist a number of works on streaming and distributed algorithms, such as \([14, 19, 35, 52, 53, 55, 74]\).

Flat fitting with outliers. Given an integer \( j \geq 0 \) and a set of points in \( \mathbb{R}^d \), the flat fitting problem is to find a \( j \)-dimensional flat having the smallest maximum distance to the input points \([41]\). Obviously, the MEB problem is a special case with \( j = 0 \). In high dimensions, Har-Peled and Varadarajan \([42]\) provided a linear time algorithm if \( j \) is assumed to be fixed; their running time was further reduced by Panagrathy \([61]\) based on a core-set approach. There also exist several methods considering flat fitting with outliers but only for low-dimensional case \([3, 43]\).

SVM with outliers. Given two point sets \( P_1 \) and \( P_2 \) in \( \mathbb{R}^d \), the problem of Support Vector Machine (SVM) is to find the largest margin to separate \( P_1 \) and \( P_2 \) (if they are separable) \([17]\). SVM can be formulated as a quadratic programming problem, and a number of efficient techniques have been developed in the past, such as the soft margin SVM \([22, 63]\), \( \nu \)-SVM \([23, 66]\), and Core-SVM \([70]\). There also exist a number of works on designing robust algorithms of SVM with outliers \([27, 67, 72]\).

\(^1\) The asymptotic notation \( \tilde{O}(f) = O(f \cdot \text{polylog}(\frac{nd}{\epsilon})) \).
2 Definitions and Preliminaries

In this paper, we let $|A|$ denote the number of points of a given 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 $B(c, r)$ to denote the ball centered at a point $c$ with radius $r > 0$. Below, we give several definitions used throughout this 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 $MEB(P)$ and $Rad(P)$, respectively.

A ball $B(c, r)$ is called a $\lambda$-approximation of $MEB(P)$ for some $\lambda \geq 1$, if the ball covers all points in $P$ and has radius $r \leq \lambda Rad(P)$.

**Definition 2 (MEB with Outliers).** Given a set $P$ of $n$ points in $\mathbb{R}^d$ and a small parameter $\gamma \in (0, 1)$, the MEB with outliers problem is to find the smallest ball that covers $(1 - \gamma)n$ points. Namely, the task is to find a subset of $P$ with size $(1 - \gamma)n$ such that the resulting MEB is the smallest among all possible choices of the subset. The obtained ball is denoted by $MEB(P, \gamma)$.

For convenience, we use $P_{opt}$ to denote the optimal subset of $P$ with respect to $MEB(P, \gamma)$. That is, $P_{opt} = \arg\min_Q \{ Rad(Q) \mid Q \subset P, |Q| = (1 - \gamma)n \}$. From Definition 2, we can see that the main issue is to determine the subset of $P$. Actually, solving such combinatorial problems involving outliers are often challenging. In Section 5, we will present an example to show that it is impossible to achieve an approximation factor less than 2 for MEB with outliers, if the time complexity is required to be independent of $n$. Therefore, we consider finding the bi-criteria approximation. Actually, it is also a common way for solving other optimization problems with outliers. For example, Mount et al. [59] and Meyerson et al. [57] studied the bi-criteria approximation algorithms respectively for the problems of linear regression and $k$-median clustering with outliers before.

**Definition 3 (Bi-criteria Approximation).** Given an instance $(P, \gamma)$ for MEB with outliers and two small parameters $0 < \epsilon, \delta < 1$, a $(1 + \epsilon, 1 + \delta)$-approximation of $(P, \gamma)$ is a ball that covers at least $(1 - (1 + \delta)\gamma)n$ points and has radius at most $(1 + \epsilon)Rad(P_{opt})$.

When both $\epsilon$ and $\delta$ are small, the bi-criteria approximation is very close to the optimal solution with only slight changes on the number of covered points and the radius.

We also extend Definition 2 to the problem called minimum enclosing “x” (MEX) with Outliers, where the “x” could be any specified shape. To keep the structure of our paper more compact, we state the formal definition of MEX with outliers and the corresponding results in Section 5.

2.1 A More Careful Analysis for Core-set Construction in [11]

Before presenting our main results, we first revisit the core-set construction algorithm for MEB of Bădoiu and Clarkson [11], since their method will be used in our algorithms for MEB with outliers.

Let $0 < \epsilon < 1$. The algorithm in [11] yields an MEB core-set of size $2/\epsilon$ (for convenience, we always assume that $2/\epsilon$ is an integer). However, there is a small issue in their paper. The analysis assumes that the exact MEB of the core-set is computed in each iteration, but in fact one may only compute an approximate MEB. Thus, an immediate question is whether the quality is still guaranteed with such a change. Kumar et al. [50] fixed this issue, and showed that computing a $(1 + O(\epsilon^2))$-approximate MEB for the core-set in each iteration still guarantees a core-set with size $O(1/\epsilon)$, where the hidden constant is larger than 80. Clarkson [20] systematically studied the Frank-Wolfe algorithm [32], and showed that the greedy core-set construction algorithm of MEB, as a special case of the Frank-Wolfe algorithm, yields a core-set with size slightly larger than $4/\epsilon$. Note that there exist several other methods yielding even lower core-set size [12] [19], but their construction algorithms are more complicated and thus not applicable to our problems. Increasing the core-set size from $2/\epsilon$ to $a/\epsilon$ (for some $a > 2$) is neglectable in asymptotic analysis. But in Section 4, we will show that it could cause serious issue if outliers exist. Hence, a core-set of size $2/\epsilon$ is still desirable. For this purpose, we provide a new analysis which is also interesting in its own right.

For the sake of completeness, we first briefly introduce the idea of the core-set construction algorithm in [11]. 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 $MEB(T)$ and adds to $T$ the farthest point from the current center of
Finally, the center of MEB(T) induces a $(1 + \epsilon)$-approximation for MEB(P). The selected set of $2/\epsilon$ points (i.e., $T$) is called the core-set of MEB. To ensure the expected improvement in each iteration, 

$$r_{i+1} \geq (1 + \epsilon)\text{Rad}(P) - L_i; \quad r_{i+1} \geq \sqrt{r_i^2 + L_i^2}, \quad (1)$$

where $r_i$ and $r_{i+1}$ are the radii of MEB(T) in the $i$-th and $(i + 1)$-th iterations, respectively, and $L_i$ is the shifting distance of the center of MEB(T) from the $i$-th to $(i + 1)$-th iteration.

As mentioned earlier, we often compute only an approximate MEB(T) in each iteration. In the $i$-th iteration, we let $c_i$ and $o_i$ denote the centers of the exact and the approximate MEB(T), respectively. Suppose that $||c_i - o_i|| \leq \xi r_i$, where $\xi \in (0, \frac{1}{1+\epsilon})$ (we will see why this bound is needed later). Using another algorithm proposed in [11], one can obtain the point $o_i$ in $O(\frac{1}{\epsilon} |T| d)$ time. Note that we only compute $o_i$ rather than $c_i$ in each iteration. Hence we can only select the farthest point (say $q$) to $o_i$. If $||q - o_i|| \leq (1 + \epsilon)\text{Rad}(P)$, we are done and a $(1 + \epsilon)$-approximation for MEB is already obtained. Otherwise, we have

$$(1 + \epsilon)\text{Rad}(P) < ||q - o_i|| \leq ||q - c_{i+1}|| + ||c_{i+1} - c_i|| + ||c_i - o_i|| \leq r_{i+1} + L_i + \xi r_i \quad (2)$$

by the triangle inequality (see Figure 1). In other words, we should replace the first inequality of (1) by $r_{i+1} > (1 + \epsilon)\text{Rad}(P) - L_i - \xi r_i$. Also, the second inequality of (1) still holds since it depends only on the property of the exact MEB (see Lemma 2.1 in [11]). Thus, we have

$$r_{i+1} \geq \max \left\{ \sqrt{r_i^2 + L_i^2}, (1 + \epsilon)\text{Rad}(P) - L_i - \xi r_i \right\}. \quad (3)$$

This leads to the following theorem whose proof can be found in Section 4.

**Theorem 1.** In the core-set construction algorithm of [11], 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 $\xi r_i$, then $\xi = \frac{2}{1 - s^2}$ for some $s \in (0, 1)$, the final core-set size is bounded by $z = \frac{2}{1 - s^2}$. Also, the bound could be arbitrarily close to $2/\epsilon$ when $s$ is small enough.

**Remark 1.** We want to emphasize a simple observation on the above core-set construction procedure, which will be used in our algorithms and analysis 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 $(1 + \epsilon)\text{Rad}(P)$, the inequality (2) always holds and the following analysis is still valid. If no such a point exists (i.e., $P \setminus B(o_i, (1 + \epsilon)\text{Rad}(P)) = \emptyset$), a $(1 + \epsilon)$-approximate MEB (i.e., $B(o_i, (1 + \epsilon)\text{Rad}(P))$) has already been obtained.

### 3 Two Key Lemmas for Handling Outliers

In this section, we introduce two important techniques, Lemma 1 and 2 for solving the problem of MEB with outliers in sub-linear time; the proofs are placed in Section 3.1 and 3.2 respectively. The algorithms are presented in Section 4. Moreover, these techniques can be generalized to solve a broader range of optimization problems, and we show the details in Section 5.

To shed some light on our ideas, consider using the core-set construction method in Section 2.1 to compute a bi-criteria $(1 + \epsilon, 1 + \delta)$-approximation for an instance $(P, \gamma)$ of MEB with outliers. Let $o_i$ be the obtained ball center in the current iteration, and $Q$ be the set of $(1 + \delta)\gamma n$ farthest points to $o_i$ from $P$. A key step for updating $o_i$ is finding a point in the set $P_{opt} \cap Q$ (the formal analysis is given in Section 4). Actually, this can be done by performing a random sampling from $Q$. However, it requires to compute the set $Q$ in advance, which takes an $\Omega(n d)$ time complexity. To keep the running time to be sub-linear, we need to find a point from $P_{opt} \cap Q$ by a more sophisticated way.

Since $P_{opt}$ is mixed with outliers in the set $Q$, simple uniform sampling cannot realize our goal. To remedy this issue, we propose a “two level” sampling procedure which is called “Uniform-Adaptive Sampling” (see Lemma 1). Roughly speaking, we take a random sample $A$ of size $n'$ first (i.e., the uniform sampling step), and then randomly select a point from $Q'$, the set of the farthest $\frac{3}{2}(1 + \delta)\gamma n'$ points from $A$ to $o_i$ (i.e., the adaptive sampling step). According to Lemma 1 with probability at least $(1 - \eta) \frac{3}{2}(1 + \delta)\gamma n'$, the selected point belongs to $P_{opt} \cap Q$; more importantly, the sample size $n'$ is independent of $n$ and $d$. The key to prove Lemma 1 is to show that the size of the intersection $Q' \cap (P_{opt} \cap Q)$ is large enough. By setting an appropriate value for $n'$, we can prove a lower bound of $|Q' \cap (P_{opt} \cap Q)|$. 4
Lemma 1 (Uniform-Adaptive Sampling). Let \( \eta_1 \in (0, 1) \). If we sample \( n' = O\left(\frac{1}{\delta^3} \log \frac{1}{\eta_1}\right) \) points independently and uniformly at random from \( P \) and let \( Q' \) be the set of farthest \( (1 + \delta)\gamma n' \) points to \( o_i \) from the sample, then, with probability at least \( 1 - \eta_1 \), the following holds

\[
\frac{|Q' \cap (P_{opt} \cap Q)|}{|Q'|} \geq \frac{\delta}{3(1 + \delta)}.
\]  

The Uniform-Adaptive Sampling procedure will result in a “side-effect”. To boost the overall success probability, we have to repeatedly run the algorithm multiple times and each time the algorithm will generate a candidate solution (i.e., the ball center). Consequently we have to select the best one as our final solution. With a slight abuse of notation, we still use \( o_i \) to denote a candidate ball center; to achieve a \((1 + \epsilon, 1 + \delta)\)-approximation, its radius should be the \((1 + \delta)\gamma n + 1\)-th largest distance from \( P \) to \( o_i \), which is denoted as \( l_i \). A straightforward way is to compute the value “\( l_i \)” in linear time for each candidate and return the one having the smallest \( l_i \). In this section, we propose the “Sandwich Lemma” to estimate \( l_i \) in sub-linear time (see Lemma 2). Let \( B \) be the set of \( n'' \) sampled points from \( P \) in Lemma 2 and \( l_i \) be the \((1 + \delta)^2 \gamma n'' + 1\)-th largest distance from \( B \) to \( o_i \). The key idea is to prove that the ball \( B(o_i, l_i) \) is “sandwiched” by two balls \( B(o_i, l_i') \) and \( B(o_i, l_i) \), where \( l_i' \) is a carefully designed value satisfying (i) \( l_i' \leq l_i \leq l_i \) and (ii) \( |P \setminus B(o_i, l_i')| \leq (1 + O(\delta))\gamma n \). See Figure 2 for an illustration. These two conditions of \( l_i' \) can imply the inequalities \((5)\) and \((6)\) of Lemma 2. Further, the inequalities \((5)\) and \((6)\) jointly imply that \( l_i \) is a qualified estimation of \( l_i' \); if \( B(o_i, l_i) \) is a \((1 + \epsilon, 1 + \delta)\)-approximation, the ball \( B(o_i, l_i) \) should be a \((1 + \epsilon, 1 + O(\delta))\)-approximation. Similar to Lemma 2, the sample size \( n'' \) is also independent of \( n \) and \( d \).

Lemma 2 (Sandwich Lemma). Let \( \eta_2 \in (0, 1) \) and assume \( \delta < 1/3 \). If we sample \( n'' = O\left(\frac{1}{\delta^3} \log \frac{1}{\eta_2}\right) \) points independently and uniformly at random from \( P \) and let \( l_i \) be the \((1 + \delta)^2 \gamma n'' + 1\)-th largest distance from the sample to \( o_i \), then, with probability \( 1 - \eta_2 \), the following holds

\[
\bar{l}_i \leq l_i; \quad |P \setminus B(o_i, \bar{l}_i)| \leq (1 + O(\delta))\gamma n. \tag{5}
\]

3.1 Proof of Lemma 2

Let \( A \) denote the set of sampled \( n' \) points from \( P \). First, we know \(|Q| = (1 + \delta)\gamma n\) and \(|P_{opt} \cap Q| \geq \delta \gamma n\) (since there are at most \( \gamma n \) outliers in \( Q \)). For ease of presentation, let \( \lambda = \frac{|P_{opt} \cap Q|}{\gamma n} \geq \delta \). Let \( \{x_i \mid 1 \leq i \leq n'\} \) be \( n' \) independent random variables with \( x_i = 1 \) if the \( i \)-th sampled point of \( A \) belongs to \( P_{opt} \cap Q \), and \( x_i = 0 \) otherwise. Thus, \( E[x_i] = \lambda \) for each \( i \). Let \( \sigma \) be a small parameter in \((0, 1)\). By using the Chernoff bound, we have \( \Pr\left(\sum_{i=1}^{n'} x_i \notin (1 \pm \sigma)\lambda n'\right) \leq e^{-O(\sigma^2 \lambda n')} \). That is,

\[
\Pr\left(|A \cap (P_{opt} \cap Q)| \in (1 \pm \sigma)\lambda n'\right) \geq 1 - e^{-O(\sigma^2 \lambda n')}. \tag{7}
\]

Similarly, we have

\[
\Pr\left(|A \cap Q| \in (1 \pm \sigma)(1 + \delta)\gamma n'\right) \geq 1 - e^{-O(\sigma^2 (1 + \delta) \gamma n')}. \tag{8}
\]

Note that \( n' = O\left(\frac{1}{\delta^3} \log \frac{1}{\eta_1}\right) \). By setting \( \sigma < 1/2 \) in \((7)\) and \((8)\), we have

\[
|A \cap (P_{opt} \cap Q)| \geq \frac{1}{2} \delta \gamma n' \quad \text{and} \quad |A \cap Q| < \frac{3}{2} (1 + \delta) \gamma n'. \tag{9}
\]

with probability \( 1 - \eta_1 \). Note that \( Q \) contains all the farthest \((1 + \delta)\gamma n\) points to \( o_i \). Denote by \( l_i \) the \(((1 + \delta)\gamma n + 1)\)-th largest distance from \( P \) to \( o_i \). Thus

\[
A \cap Q = \{p \in A \mid \|p - o_i\| > l_i\}. \tag{10}
\]
Also, since $Q'$ is the set of the farthest $\frac{2}{3}(1+\delta)\gamma n'$ points to $o_i$ from $A$, there exists some $l'_i > 0$ such that

$$Q' = \{p \in A \mid ||p - o_i|| > l'_i\}. \quad (11)$$

and (11) imply that either $(A \cap Q) \subseteq Q'$ or $Q' \subseteq (A \cap Q)$. Since $|A \cap Q| < \frac{3}{2}(1+\delta)\gamma n'$ and $|Q'| = \frac{2}{3}(1+\delta)\gamma n'$, we know $(A \cap Q) \subseteq Q'$. Therefore,

$$(A \cap (P_{opt} \cap Q)) = (P_{opt} \cap (A \cap Q)) \subseteq Q'. \quad (12)$$

Obviously,

$$(A \cap (P_{opt} \cap Q)) \subseteq (P_{opt} \cap Q). \quad (13)$$

The above (12) and (13) together imply

$$(A \cap (P_{opt} \cap Q)) \subseteq (Q \cap (P_{opt} \cap Q)). \quad (14)$$

Moreover, since $Q' \subseteq A$, we have

$$(Q' \cap (P_{opt} \cap Q)) \subseteq (A \cap (P_{opt} \cap Q)). \quad (15)$$

Consequently, (14) and (15) together imply $Q' \cap (P_{opt} \cap Q) = A \cap (P_{opt} \cap Q)$ and hence

$$\frac{|Q' \cap (P_{opt} \cap Q)|}{|Q'|} = \frac{|A \cap (P_{opt} \cap Q)|}{|Q'|} \geq \frac{\delta}{3(1+\delta)}, \quad (16)$$

where the final inequality comes from the first inequality of (6) and the fact $|Q'| = \frac{2}{3}(1+\delta)\gamma n'$.

### 3.2 Proof of Lemma 2

Let $B$ denote the set of sampled $n''$ points from $P$. For simplicity, let $t = (1+\delta)\gamma n$. Assume $\bar{l}_i > 0$ is the value such that $|P \setminus B(o_i, \bar{l}_i)| = \frac{(1+\delta)^2}{1-\delta}\gamma n$. Recall that $l_i$ is the $(t+1)$-th largest distance from $P$ to $o_i$. Since $(1+\delta)\gamma n < \frac{(1+\delta)^2}{1-\delta}\gamma n$, it is easy to know $\bar{l}_i \leq l_i$. Below, we aim to prove that the $((1+\delta)^2\gamma n'' + 1)$-th farthest point from $B$ is in the ring bounded by the spheres $B(o_i, \bar{l}_i)$ and $B(o_i, l_i)$ (see Figure 2).

Again, using the Chernoff bound (let $\sigma = \delta/2$) and the same idea for proving (6), since $|B| = n'' = O\left(\frac{1}{\gamma n} \log \frac{1}{\eta_2}\right)$, we have

$$|B \setminus B(o_i, \bar{l}_i)| \geq (1-\delta/2) \frac{(1+\delta)^2}{1-\delta} \gamma n'' > (1-\delta) \frac{(1+\delta)^2}{1-\delta} \gamma n'' = (1+\delta)^2\gamma n''; \quad (17)$$

$$|B \cap Q| \leq (1+\delta/2) \frac{t}{n} n'' < (1+\delta) \frac{t}{n} n'' = (1+\delta)^2\gamma n'', \quad (18)$$

with probability $1 - \eta_2$. Suppose that (17) and (18) both hold. Recall that $\bar{l}_i$ is the $((1+\delta)^2\gamma n'' + 1)$-th largest distance from the sampled points $B$ to $o_i$, so $|B \setminus B(o_i, \bar{l}_i)| = (1+\delta)^2\gamma n''$, and thus $\bar{l}_i \geq \bar{l}_i$ by (17).

The inequality (18) implies that the $((1+\delta)^2\gamma n'' + 1)$-th farthest point (say $q_z$) from $B$ to $o_i$ is not in $Q$. Then, we claim that $B(o_i, \bar{l}_i) \cap Q = \emptyset$. Otherwise, let $q_y \in B(o_i, \bar{l}_i) \cap Q$. Then we have

$$||q_y - o_i|| < \bar{l}_i = ||q_x - o_i||. \quad (19)$$

Note that $Q$ is the set of farthest $t$ points to $o_i$ of $P$. So $q_x \notin Q$ implies

$$||q_x - o_i|| < \min_{q \in Q} ||q - o_i|| \leq ||q_y - o_i|| \quad (20)$$

which is in contradiction to (19). Therefore, $B(o_i, \bar{l}_i) \cap Q = \emptyset$. Further, since $B(o_i, l_i)$ excludes exactly the farthest $t$ points (i.e., $Q$), $B(o_i, \bar{l}_i) \cap Q = \emptyset$ implies $\bar{l}_i \leq l_i$.

Overall, we have $\bar{l}_i \leq [\bar{l}_i, l_i]$, i.e., the $((1+\delta)^2\gamma n'' + 1)$-th farthest point from $B$ locates in the ring bounded by the spheres $B(o_i, \bar{l}_i)$ and $B(o_i, l_i)$ as shown in Figure 2. Also, $\bar{l}_i \geq \bar{l}_i$ implies

$$|P \setminus B(o_i, \bar{l}_i)| \leq |P \setminus B(o_i, \bar{l}_i)| = \frac{(1+\delta)^2}{1-\delta} \gamma n = (1+O(\delta))\gamma n, \quad (21)$$

where the last equality comes from the assumption $\delta < 1/3$. So (5) and (6) are true in Lemma 2.
4 Sub-linear Time Algorithm of MEB with Outliers

Recall the remark following Theorem 1. As long as the selected point has a distance to the center of $\text{MEB}(T)$ larger than $(1 + \epsilon)$ times the optimal radius, the expected improvement will always be guaranteed. Following this observation, we investigate the following approach. Suppose we run the core-set construction procedure described in Theorem 1 (we should replace $P$ by $P_{\text{opt}}$ in our following analysis). In the $i$-th step, we add an arbitrary point from $P_{\text{opt}} \setminus B(o_i, (1 + \epsilon) \text{Rad}(P_{\text{opt}}))$ to $T$. We know that a $(1 + \epsilon)$-approximation is obtained after at most $\frac{2}{(1 - \gamma)^2}$ steps, that is, $P_{\text{opt}} \subset B(o_i, (1 + \epsilon) \text{Rad}(P_{\text{opt}}))$ for some $i \leq \frac{2}{(1 - \gamma)^2}$.

However, we need to solve two key issues in order to implement the above approach: (i) how to determine the value of Algorithm 1. We can implicitly avoid the first issue via replacing $(1 + \epsilon)$ with an arbitrary point from $P_{\text{opt}}$. With probability $\gamma$, we add an arbitrary point from $P_{\text{opt}} \setminus B(o_i, (1 + \epsilon) \text{Rad}(P_{\text{opt}}))$. Actually, we can implicitly avoid the first issue via replacing $(1 + \epsilon) \text{Rad}(P_{\text{opt}})$ by the $t$-th largest distance from the points of $P$ to $o_i$, where we set $t = (1 + \delta) \gamma n$ for achieving a $(1 + \epsilon, 1 + \delta)$-approximation in the following analysis. For the second issue, we randomly select one point from the farthest $t$ points of $P$ to $o_i$, and show that it belongs to $P_{\text{opt}} \setminus B(o_i, (1 + \epsilon) \text{Rad}(P_{\text{opt}}))$ with a certain probability.

Based on the above idea, we present a sub-linear time $(1 + \epsilon, 1 + \delta)$-approximation algorithm in this section. To better understand the algorithm, we show a linear time algorithm first (Algorithm 1 in Sections 4.1). Note that Bádoiu et al. [13] also achieved a $(1 + \epsilon, 1 + \delta)$-approximation algorithm but with a higher complexity. Please see more details in our analysis on the running time at the end of Sections 4.1. More importantly, we improve the running time of Algorithm 1 to be sub-linear. For this purpose, we need to avoid computing the farthest $t$ points to $o_i$, since this operation will take linear time. Also, Algorithm 1 generates a set of candidates for the solution and we need to select the best one. This process also costs linear time. By using the techniques proposed in Section 3, we can remedy these issues and develop a sub-linear time algorithm that has the sample complexity independent of $n$ and $d$, in Section 4.2.

4.1 A Linear Time Algorithm

**Algorithm 1** $(1 + \epsilon, 1 + \delta)$-approximation Algorithm for MEB with Outliers

**Input:** A point set $P$ with $n$ points in $\mathbb{R}^d$, the fraction of outliers $\gamma \in (0, 1)$, and the parameters $0 < \epsilon, \delta < 1$, $z \in \mathbb{Z}^+$.  
1: Let $t = (1 + \delta) \gamma n$.  
2: Initially, randomly select a point $p \in P$ and let $T = \{p\}$.  
3: $i = 1$; repeat the following steps until $i > z$:  
   (1) Compute the approximate MEB center $o_i$ of $T$.  
   (2) Let $Q$ be the set of farthest $t$ points from $P$ to $o_i$; denote by $l_i$ the $(t + 1)$-th largest distance from $P$ to $o_i$.  
   (3) Randomly select a point $q \in Q$; and add it to $T$.  
   (4) $i = i + 1$.  
4: Output the ball $B(o_i, l_i)$ where $i = \arg \min \{l_i \mid 1 \leq i \leq z\}$.

In this section, we present our linear time $(1 + \epsilon, 1 + \delta)$-approximation algorithm for MEB with outliers (see Algorithm 1). For convenience, denote by $e_i$ and $r_i$ the exact center and radius of $\text{MEB}(T)$ respectively in the $i$-th round of Step 3 of Algorithm 1. In Step 3(1), we compute the approximate center $o_i$ with a distance to $c_i$ of less than $\frac{\gamma \text{Rad}(T)}{1 + \delta + \epsilon} \text{Rad}(T)$, where $s \in (0, 1)$ as described in Theorem 1 (we will determine the value of $s$ in our following analysis on the running time). The following theorem shows the success probability of Algorithm 1.

**Theorem 2.** If the input parameter $z = \frac{2}{(1 - \gamma)^2}$ (we assume it is an integer for convenience), then with probability $(1 - \gamma)(\frac{\delta}{1 + \gamma})^2$, Algorithm 1 outputs a $(1 + \epsilon, 1 + \delta)$-approximation for the MEB with outliers problem.

Before proving Theorem 2, we present the following two lemmas first.

**Lemma 3.** With probability $(1 - \gamma)(\frac{\delta}{1 + \gamma})^2$, after running $z$ rounds in Step 3, the set $T \subset P_{\text{opt}}$ in Algorithm 1.
Proof. Initially, because $|P_{opt}|/|P| = 1 - \gamma$, the first selected point in Step 2 belongs to $P_{opt}$ with probability $1 - \gamma$. In each of the $z$ rounds in Step 3, the selected point belongs to $P_{opt}$ with probability $1 - \frac{z}{1+\delta}$, since

$$\frac{|P_{opt} \cap Q|}{|Q|} = 1 - \frac{|Q \setminus P_{opt}|}{|Q|} \geq 1 - \frac{|P \setminus P_{opt}|}{|Q|} = 1 - \frac{\gamma n}{(1 + \delta)\gamma n} = \frac{\delta}{1 + \delta}. \quad (22)$$

Therefore, $T \subset P_{opt}$ with probability $(1 - \gamma)(\frac{\delta}{1 + \delta})^z \Box$.

Lemma 4. In the $i$-th round of Step 3 for $1 \leq i \leq z$, at least one of the following two events happens: (1) $o_i$ is the ball of radius $1 + \epsilon$ and $\delta$-approximation; (2) $r_{i+1} > (1 + \epsilon)Rad(P_{opt}) - ||c_i - c_{i+1}|| - \xi r_i$.

Proof. If $l_i \leq (1 + \epsilon)Rad(P_{opt})$, then we are done. That is, the ball $B(o_i, l_i)$ covers $(1 - (1 + \delta)\gamma)n$ points with radius $l_i \leq (1 + \epsilon)Rad(P_{opt})$ (the first event happens). Otherwise, $l_i > (1 + \epsilon)Rad(P_{opt})$ and we consider the second event. Let $q$ be the point added to $T$ in the $i$-th round. Using the triangle inequality, we have

$$||o_i - q|| \leq ||o_i - c_i|| + ||c_i - c_{i+1}|| + ||c_{i+1} - q|| \leq \xi r_i + ||c_i - c_{i+1}|| + r_{i+1}. \quad (23)$$

Since $l_i > (1 + \epsilon)Rad(P_{opt})$ and $q$ lies outside of $B(o_i, l_i)$, i.e., $||o_i - q|| \geq l_i > (1 + \epsilon)Rad(P_{opt})$, (23) implies that the second event happens and the proof is completed. $\Box$

Proof. (of Theorem 2) Suppose that the first event of Lemma 4 never happens. As a consequence, we obtain a series of inequalities for each pair of radii $r_{i+1}$ and $r_i$, i.e., $r_{i+1} > (1 + \epsilon)Rad(P_{opt}) - ||c_i - c_{i+1}|| - \xi r_i$. Assume that $T \subset P_{opt}$ in Lemma 3, i.e., each time the algorithm correctly adds a point from $P_{opt}$ to $T$. Using the almost identical idea for proving Theorem 1 in Section 2.1, we know that a $(1 + \epsilon)$-approximate MEB of $P_{opt}$ is obtained after at most $z$ rounds. The success probability directly comes from Lemma 3. Overall, we obtain Theorem 2. $\Box$

Moreover, Theorem 2 implies the following corollary.

Corollary 1. If one repeatedly runs Algorithm 1 $O((\frac{1}{1-\gamma})(1 + \frac{1}{\epsilon})^z)$ times, with constant probability, the algorithm outputs a $(1 + \epsilon, 1 + \delta)$-approximation for the problem of MEB with outliers.

Running time. In Theorem 2 we set $z = \frac{2}{1 - \epsilon \gamma}$ and $s \in (0, 1)$. To keep $z$ small, according to Theorem 1 we set $s = \frac{1}{1 - \epsilon \gamma}$ so that $z = \frac{2}{\epsilon} + 1$ (only larger than the lower bound $\frac{2}{\epsilon}$ by 1). For each round of Step 3, we need to compute an approximate center $o_i$ that has a distance to the exact one less than $\xi r_i = s^2 r_i = O(\epsilon^2) r_i$. Using the algorithm proposed in [11], this can be done in $O(\frac{1}{\epsilon^2}(\frac{1}{\epsilon} + d))$ time. Also, the set $Q$ can be obtained in linear time by the algorithm in [10]. In total, the time complexity for obtaining a $(1 + \epsilon, 1 + \delta)$-approximation in Corollary 1 is

$$O\left(C\frac{1}{\epsilon}(n + \frac{1}{\epsilon}d)\right), \quad (24)$$

where $C = O((\frac{1}{1-\gamma})(1 + \frac{1}{\epsilon})^{\frac{2}{\epsilon}+1})$. As mentioned before, Bädoiu et al. [13] also achieved a linear time bi-criteria approximation. However, the hidden constant of their running time is exponential in $\Theta(C_{opt})$ (where $\mu$ is defined in [13], and should be $\delta \gamma$ to ensure a $(1 + \epsilon, 1 + \delta)$-approximation) that is much larger than $\frac{2}{\epsilon} + 1$.

4.2 Improvement on Running Time

In this section, we show that the running time of Algorithm 1 can be further improved to be independent of the number of points $n$. First, we observe that it is not necessary to compute the set $Q$ of the farthest $t$ points in Step 3(2) of the algorithm. Actually, as long as the selected point $q$ is part of $P_{opt} \cap Q$ in Step 3(3), a $(1 + \epsilon, 1 + \delta)$-approximation is still guaranteed. The Uniform-Adaptive Sampling procedure proposed in Section 3 can help us to obtain a point $q \in P_{opt} \cap Q$ without computing the set $Q$. Moreover, in Lemma 2 we show that the radius of each candidate solution can be estimated via random sampling. Overall, we achieve a sub-linear time algorithm (Algorithm 2). Following the analysis in Section 4.1 we set $s = \frac{1}{1 - \epsilon \gamma}$ so that $z = \frac{2}{\epsilon} + 1$. We present the results in Theorem 3 and Corollary 2. Comparing with Theorem 2 we have an extra $(1 - \eta_1)(1 - \eta_2)$ in the success probability in Theorem 3 due to the probabilities from Lemmas 1 and 2.
Lemma 1 and 2 can be further extended to the ideas of Lemma 1 and 2 of MEB with outliers. We observe that the shape "x" is a union of the shapes determined by the following two components: "c" and "d". To deal with MEX with outliers, as long as the shape "x" satisfies several properties.

### The Extension: MEX with Outliers

To boost the success probability in Theorem 3, we need to repeatedly run Algorithm 2 and output the best candidate. However, we need to be careful on setting the parameters. The success probability in Theorem 3 consists of two parts, \( P_1 = (1 - \gamma)(1 - \eta_1)(1 - \eta_2)\), and \( P_2 = (1 - \eta_0)^2 \), where \( P_1 \) indicates the probability that \( \{o_1, \ldots, o_2\} \) contains a qualified candidate, and \( P_2 \) indicates the success probability of Lemma 2 over all the \( z \) rounds. Therefore, if we run Algorithm 2 \( N = O\left(\frac{1}{\gamma^2}\right) \) times, with constant probability, the set of all the generated candidates contains at least one that yields a \( (1 + \epsilon, 1 + O(\delta)) \)-approximation; moreover, to guarantee that we can correctly estimate the resulting radii of all the candidates via the Sandwich Lemma with constant probability, we need to set \( \eta_2 = O\left(\frac{1}{\gamma^2}\right) \) (because there are \( O(zN) \) candidates).

### Corollary 2

If one repeatedly runs Algorithm 2 \( N = O\left(\frac{1}{\gamma^2}\right) \) times with setting \( \eta_2 = O\left(\frac{1}{\gamma^2}\right) \), with constant probability, the algorithm outputs a \( (1 + \epsilon, 1 + O(\delta)) \)-approximation for the problem of MEX with outliers.

The calculation of running time is similar to (24) in Section 4.1. We just replace \( n \) by \( \max\{n', n''\} = O\left(\frac{1}{\gamma^2} \log \frac{1}{\eta_2}\right) = O\left(\frac{1}{\gamma^2} \log(zN)\right) = O\left(\frac{1}{\gamma^2}\right) \) and change the value of \( C \) to be \( O\left(\frac{1}{\gamma^2}(1 - \eta_1)(1 - \eta_2)(3 + \frac{3}{4})\right)^{2+1} \).

So the total running time is independent of \( n \). Also, to covert the result from \( (1 + \epsilon, 1 + O(\delta)) \)-approximation to \( (1 + \epsilon, 1 + \delta) \)-approximation, we just need to reduce the value of \( \delta \) in the input of Algorithm 2 appropriately.

### 5 The Extension: MEX with Outliers

In this section, we extend Definition 2 and define a more general problem called minimum enclosing "x" (MEX) with Outliers. We observe that the ideas of Lemma 1 and 2 can be further extended to deal with MEX with outliers problems, as long as the shape "x" satisfies several properties. To describe a shape "x", we need to clarify three basic concepts: center, size, and distance function.

Let \( \mathcal{X} \) be the set of specified shapes in \( \mathbb{R}^d \). In this paper, we require that each shape \( x \in \mathcal{X} \) is uniquely determined by the following two components: "center" of \( x \), and "size" of \( x \). For any two shapes \( x_1, x_2 \in \mathcal{X} \), \( x_1 = x_2 \) if and only if \( c(x_1) = c(x_2) \) and \( s(x_1) = s(x_2) \). Moreover, given a center \( o_0 \) and a value \( l_0 \geq 0 \), we use \( x(o_0, l_0) \) to denote the shape \( x \) with \( c(x) = o_0 \) and \( s(x) = l_0 \). For different shapes, we have different definitions for the center and size. For example, if \( x \) is a ball, \( c(x) \) and \( s(x) \) should be the ball center and the radius respectively; given \( o_0 \in \mathbb{R}^d \) and \( l_0 \geq 0 \), \( x(o_0, l_0) \) should be the ball \( B(o_0, l_0) \). As a more complicated example, consider the \( k \)-center clustering with outliers problem, which is to find \( k \) balls to cover the input point set excluding a certain number of outliers and minimize the maximum radius (w.l.o.g., we can assume that the \( k \) balls have the same radius). For this problem, the shape "x" is a union of \( k \) balls in \( \mathbb{R}^d \); the center \( c(x) \) is the set of the \( k \) ball centers and the size \( s(x) \) is the radius.

\(^2\) The asymptotic notation \( O(f) = O(f \cdot \text{polylog}(\frac{1}{\delta})) \).
For any point \( p \in \mathbb{R}^d \) and any shape \( x \in \mathcal{X} \), we also need to define a distance function \( f(c(x), p) \) between the center \( c(x) \) and \( p \). For example, if \( x \) is a ball, \( f(c(x), p) \) is simply equal to \( ||p - c(x)|| \); if \( x \) is a union of \( k \) balls with the center \( c(x) = \{c_1, c_2, \ldots, c_k\} \), the distance should be \( \min_{1 \leq j \leq k} ||p - c_j|| \). Note that the distance function is only for ranking the points to \( c(x) \), and not necessary to be non-negative (e.g., in Section 10 we define a distance function \( f(c(x), p) \leq 0 \) for SVM). By using this distance function, we can define the set “\( Q \)” and the value “\( f_i \)” when generalizing Lemma 1 and 2 below. To guarantee their correctness, we also require \( \mathcal{X} \) to satisfy the following three properties.

**Property 1.** For any two shapes \( x_1 \neq x_2 \in \mathcal{X} \), if \( c(x_1) = c(x_2) \), then
\[
s(x_1) \leq s(x_2) \iff x_1 \text{ is covered by } x_2,
\]
where “\( x_1 \) is covered by \( x_2 \)” means “for any point \( p \in \mathbb{R}^d \), \( p \in x_1 \Rightarrow p \in x_2 \).”

**Property 2.** Given any shape \( x \in \mathcal{X} \) and any point \( p_0 \in x \), the set
\[
\{p \mid p \in \mathbb{R}^d \text{ and } f(c(x), p) \leq f(c(x), p_0)\} \subseteq x.
\]
**Property 3.** Given any shape center \( o_0 \) and any point \( p_0 \in \mathbb{R}^d \), they together determine a value \( r_0 = \min \{r \mid r \geq 0, p_0 \in x(o_0, r)\} \); that is, \( p_0 \in x(o_0, r_0) \) and \( p_0 \notin x(o_0, r) \) for any \( r < r_0 \). (Note: usually the value \( r_0 \) is just the distance from \( p_0 \) to the shape center \( o_0 \); but for some cases, such as the SVM problem in Section 10, the shape size and distance function have different meanings).

Intuitively, Property 1 shows that \( s(x) \) defines an order of the shapes sharing the same center \( c(x) \). Property 2 shows that the distance function \( f \) defines an order of the points to a given shape center \( c(x) \). Property 3 shows that a center \( o_0 \) and a point \( p_0 \) can define a shape just “touching” \( p_0 \). We can take \( \mathcal{X} = \{\text{all } d\text{-dimensional balls}\} \) as an example. For any two concentric balls, the smaller one is always covered by the larger one (Property 1); if a point \( p_0 \) is inside a ball \( x \), any point \( p \) having the distance \( ||p - c(x)|| \leq ||p_0 - c(x)|| \) should be inside \( x \) too (Property 2); also, given a ball center \( o_0 \) and a point \( p_0 \), \( p_0 \in x(o_0, ||p_0 - o_0||) \) and \( p_0 \notin x(o_0, r) \) for any \( r < ||p_0 - o_0|| \) (Property 3).

Now, we introduce the formal definitions of the MEX with outliers problem and its bi-criteria approximation.

**Definition 4 (MEX with Outliers).** Suppose the shape set \( \mathcal{X} \) satisfies Property 1, 2, and 3. Given a set \( P \) of \( n \) points in \( \mathbb{R}^d \) and a small parameter \( \gamma \in (0, 1) \), the MEX with outliers problem is to find the smallest shape \( x \in \mathcal{X} \) that covers \((1 - \gamma)n \) points. Namely, the task is to find a subset of \( P \) with size \((1 - \gamma)n \) such that its minimum enclosing shape of \( \mathcal{X} \) is the smallest among all possible choices of the subset. The obtained solution is denoted by \( MEX(P, \gamma) \).

**Definition 5 (Bi-criteria Approximation).** Given an instance \((P, \gamma)\) for MEX with outliers and two small parameters \( 0 < \epsilon, \delta < 1 \), a \((1 + \epsilon, 1 + \delta)\)-approximation of \((P, \gamma)\) is a solution \( x \in \mathcal{X} \) that covers at least \((1 - (1 + \delta)\gamma)n \) points and has the size at most \((1 + \epsilon)s(x_{opt}) \), where \( x_{opt} \) is the optimal solution.

It is easy to see that Definition 2 of MEB with outliers actually is a special case of Definition 4 Similar to MEB with outliers, we still use \( P_{opt} \), where \( P_{opt} \subset P \) and \( |P_{opt}| = (1 - \gamma)n \), to denote the subset covered by the optimal solution of MEX with outliers.

Now, we provide the generalized versions of Lemma 1 and 2. Similar to the core-set construction method in Section 2.1, we assume that there exists an iterative algorithm \( \Gamma \) to compute MEX (without outliers); actually, this is an important prerequisite to design the sub-linear time algorithms under our framework (we will discuss the iterative algorithms for the MEX with outliers problems considered in our paper, including flat fitting, \( k \)-center clustering, and SVM, in the appendix). In the \( i \)-th iteration of \( \Gamma \), it maintains a shape center \( o_i \). Also, let \( Q \) be the set of \((1 + \delta)\gamma n \) farthest points from \( P \) to \( o_i \) with respect to the distance function \( f \). First, we need to define the value “\( l_i \)” by \( Q \) in the following claim.

**Claim 1.** There exists a value \( l_i \geq 0 \) satisfying \( P \setminus x(o_i, l_i) = Q \).

**Proof.** The points of \( P \) can be ranked based on their distances to \( o_i \). Without loss of generality, let \( P = \{p_1, p_2, \ldots, p_n\} \) with \( f(o_i, p_1) > f(o_i, p_2) > \cdots > f(o_i, p_n) \) (for convenience, we assume that any two distances are not equal; if there is a tie, we can arbitrarily decide their order to \( o_i \)). Therefore, the set \( Q = \{p_j \mid 1 \leq j \leq (1 + \delta)\gamma n\} \). Moreover, from Property 3, we know that each point \( p_j \in P \) corresponds to a value \( r_j \) that \( p_j \in x(o_i, r_j) \) and \( p_j \notin x(o_i, r) \) for any \( r < r_j \). Denote by \( x_j \) the shape \( x(o_i, r_j) \). We select the point \( p_{j_0} \) with \( j_0 = (1 + \delta)\gamma n + 1 \). From Property 2, we know that \( p_j \in x_{j_0} \) for any \( j \geq j_0 \),
i.e., (a) \( P \setminus Q \subseteq x_j \). We also need to prove that \( p_j \notin x_j \) for any \( j < j_0 \). Assume there exists some \( p_{j_1} \in x_{j_0} \) with \( j_1 < j_0 \). Then we have \( r_{j_1} < r_{j_0} \) and thus \( p_{j_0} \notin x_{j_1} \) (by Property 3). By Property 2 \( p_{j_0} \notin x_{j_1} \) implies \( f(o_i, p_{j_0}) > f(o_i, p_{j_1}) \), which is in contradiction to the fact \( f(o_i, p_{j_0}) < f(o_i, p_{j_1}) \). So we have (b) \( Q \cap x_{j_0} = \emptyset \).

From the above (a) and (b), we know \( P \setminus x_{j_0} = Q \). Therefore, we can set the value \( l_i = r_{j_0} \) and then \( P \setminus x(o_i, l_i) = Q \). \( \square \)

**Lemma 5 (Generalized Uniform-Adaptive Sampling).** Let \( \eta_1 \in (0, 1) \). If we sample \( n' = O \left( \frac{1}{\gamma} \log \frac{1}{\eta_1} \right) \) points independently and uniformly at random from \( P \) and let \( Q' \) be the set of farthest \( \frac{1}{3} (1 + \delta) \gamma n' \) points to \( o_i \) from the sample, then, with probability at least \( 1 - \eta_1 \), the following holds

\[
\frac{|Q' \cap (P_{opt} \cap Q)|}{|Q'|} \geq \frac{\delta}{3(1 + \delta)}.
\]  

**Proof.** Let \( A \) denote the set of sampled \( n' \) points from \( P \). Similar to \[10\], we have

\[
|A \cap (P_{opt} \cap Q)| > \frac{1}{2} \gamma n' \quad \text{and} \quad |A \cap Q| < \frac{3}{2} (1 + \delta) \gamma n'
\]  

with probability \( 1 - \eta_1 \). Similar to \[10\], we have

\[
A \cap Q = \{ p \in A \mid f(o_i, p) > f(o_i, p_{j_0}) \},
\]  

where \( p_{j_0} \) is the point selected in the proof of Claim \[1\] By using the same manner of Claim \[1\] we also can select a point \( p_{j_0}' \in A \) with

\[
Q' = \{ p \in A \mid f(o_i, p) > f(o_i, p_{j_0}') \}.
\]  

Then, we can prove

\[
\left( A \cap (P_{opt} \cap Q) \right) = \left( Q' \cap (P_{opt} \cap Q) \right) \tag{31}
\]

by using the same idea of \[10\]. Hence,

\[
\frac{|Q' \cap (P_{opt} \cap Q)|}{|Q'|} = \frac{|A \cap (P_{opt} \cap Q)|}{|Q'|} \geq \frac{\delta}{3(1 + \delta)}, \tag{32}
\]

where the final inequality comes from the first inequality of \[26\] and the fact \(|Q'| = \frac{2}{3} (1 + \delta) \gamma n' \). \( \square \)

**Lemma 6 (Generalized Sandwich Lemma).** Let \( \eta_2 \in (0, 1) \) and assume \( \delta < 1/3 \). \( l_i \) is the value from Claim \[4\]. We sample \( n'' = O \left( \frac{1}{\delta} \log \frac{1}{\eta_2} \right) \) points independently and uniformly at random from \( P \) and let \( l \) be the \((1 + \delta) \gamma n'' + 1\)-th farthest one from the sampled points to \( o_i \). If \( \tilde{l}_i = \min \{ r \mid r \geq 0, q \in x(o_i, r) \} \) (similar to the way defining “\( r_0 \)” in Property 3), then, with probability \( 1 - \eta_2 \), the following holds

\[
\tilde{l}_i \leq l_i; \quad |P \setminus x(o_i, \tilde{l}_i)| \leq (1 + O(\delta)) \gamma n. \tag{33} \tag{34}
\]

**Proof.** Let \( B \) denote the set of sampled \( n'' \) points from \( P \). By using the same manner of Claim \[1\] we know that there exists a value \( \tilde{l}_i > 0 \) satisfying \(|P \setminus x(o_i, \tilde{l}_i)| = \frac{1 + \delta}{1 - \delta} \gamma n \). Similar to the proof of Lemma 2, we can prove that \( \tilde{l}_i \in [\tilde{l}_i, l_i] \). Due to Property 1, we know that \( x(o_i, \tilde{l}_i) \) is “sandwiched” by the two shapes \( x(o_i, \tilde{l}_i) \) and \( x(o_i, l_i) \). Further, since \( x(o_i, l_i) \) is covered by \( x(o_i, \tilde{l}_i) \), we have

\[
|P \setminus x(o_i, \tilde{l}_i)| \leq |P \setminus x(o_i, l_i)| = \frac{(1 + \delta)^2 \gamma n = (1 + O(\delta)) \gamma n}, \tag{35}
\]

where the last equality comes from the assumption \( \delta < 1/3 \). So \[33\] and \[34\] are true. \( \square \)

**Appendix.** Due to the space limit, we place other parts in our appendix. In Section 8, 9, and 10 we propose the sub-linear time bi-criteria approximation algorithms for three different MEX with outlier problems: \( k \)-center clustering, flat fitting, and SVM with outliers. All of these problems have important applications in real world, and our results significantly reduce their time complexities comparing with existing approaches.
References

1. P. K. Agarwal, E. Ezra, and K. Fox. Geometric optimization revisited. In B. Steffen and G. J. Woeginger, editors, *Computing and Software Science - State of the Art and Perspectives*, volume 10000 of *Lecture Notes in Computer Science*, pages 66–84. Springer, 2019.
2. P. K. Agarwal, S. Har-Peled, and K. R. Varadarajan. Geometric approximation via coresets. *Combinatorial and Computational Geometry*, 52:1–30, 2005.
3. P. K. Agarwal, S. Har-Peled, and H. Yu. Robust shape fitting via peeling and grating coresets. *Discrete & Computational Geometry*, 39(1-3):38–58, 2008.
4. P. K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015.
5. A. Aggarwal, H. Imai, N. Katoh, and S. Suri. Finding k points with minimum diameter and related problems. *Journal of algorithms*, 12(1):38–56, 1991.
6. Z. Allen Zhu, Z. Liao, and Y. 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. N. Alon, S. Dar, M. Parnas, and D. Ron. Testing of clustering. *SIAM Journal on Discrete Mathematics*, 16(3):393–417, 2003.
8. B. Biggio, B. Nelson, and P. Laskov. Poisoning attacks against support vector machines. In *Proceedings of the 29th International Conference on Machine Learning, ICML 2012, Edinburgh, Scotland, UK, June 26 - July 1, 2012*, 2012.
9. B. Biggio and F. Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, 2018.
10. M. Blum, R. W. Floyd, V. Pratt, R. L. Rivest, and R. E. Tarjan. Time bounds for selection. *Journal of Computer and System Sciences*, 7(4):448–461, 1973.
11. M. Bădăoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2003.
12. M. Bădăoiu and K. L. Clarkson. Optimal core-sets for balls. *Computational Geometry*, 40(1):14–22, 2008.
13. M. Bădăoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, pages 250–257, 2002.
14. M. Ceccarello, A. Pietracaprina, and G. Pucci. Solving k-center clustering (with outliers) in mapreduce and streaming, almost as accurately as sequentially. *PVLDB*, 12(7):766–778, 2019.
15. D. Chakrabarty, P. Goyal, and R. Krishnaswamy. The non-uniform k-center problem. In 43rd *International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, pages 67:1–67:15, 2016.
16. T. M. Chan and V. Pathak. Streaming and dynamic algorithms for minimum enclosing balls in high dimensions. *Comput. Geom.*, 47(2):240–247, 2014.
17. C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3), 2011.
18. M. Charikar, S. Khuller, D. M. Mount, and G. 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.
19. M. Charikar, L. O’Callaghan, and R. 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.
20. K. L. Clarkson. Coresets, sparse greedy approximation, and the frank-wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):63, 2010.
21. K. L. Clarkson, E. Hazan, and D. P. Woodruff. Sublinear optimization for machine learning. *J. ACM*, 59(5):23:1–23:49, 2012.
22. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20:273, 1995.
23. D. J. Crisp and C. J. C. Burges. A geometric interpretation of v-SVM classifiers. In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *NIPS*, pages 244–250. The MIT Press, 1999.
24. A. Czumaj and C. Sohler. Sublinear-time algorithms.
25. A. Czumaj and C. Sohler. Sublinear-time approximation for clustering via random sampling. In *International Colloquium on Automata, Languages, and Programming*, pages 396–407. Springer, 2004.
26. H. Ding. Minimum enclosing ball revisited: Stability and sub-linear time algorithms. *CoRR*, abs/1904.03796, 2019.
27. H. Ding and J. Xu. Random gradient descent tree: A combinatorial approach for svm with outliers. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 2561–2567, 2015.
28. H. Ding, H. Yu, and Z. 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.
29. A. Efrat, M. Sharir, and A. Ziv. Computing the smallest k-enclosing circle and related problems. *Computational Geometry*, 4(3):119–136, 1994.
6 The Lower Bound of Sample Size for MEB with Outliers

Actually, it is easy to obtain a sub-linear time randomized 2-approximation algorithm for MEB with outliers (if only returning the center); one can randomly select one point \( p \in P \), and it belongs \( P_{opt} \) with probability \( 1 - \gamma \); thus, with probability \( 1 - \gamma \), the point \( p \) yields a 2-approximation by using the triangle inequality. Below, we consider an example and show that it is impossible to achieve a single-criterion \( \lambda \)-approximation for any \( \lambda < 2 \), if the time complexity is required to be independent of \( n \).

Let \( q_a, q_b, \) and \( q_c \) be three colinear points in \( \mathbb{R}^d \), where \( ||q_a - q_b|| = x \) and \( ||q_b - q_c|| = y \) (see Figure 3).

![Figure 3: \( ||q_a - q_b|| = x \) and \( ||q_b - q_c|| = y \).](image)

Let \( q_a, q_b, \) and \( q_c \) be three colinear points in \( \mathbb{R}^d \), where \( ||q_a - q_b|| = x \) and \( ||q_b - q_c|| = y \) (see Figure 3).

Let \( \gamma \in (0, 1) \) and the point set \( P = P_a \cup P_b \cup P_c \), where \( P_a \) contains a single point located at \( q_a \), \( P_b \) contains \((1 - \gamma)n - 1\) points overlapping at \( q_a \), and \( P_c \) contains \( \gamma n \) points overlapping at \( q_c \). Consider the instance \((P, \gamma)\) for the problem of MEB with outliers. Suppose \( x \ll y \). Consequently, the optimal subset \( P_{opt} \) should be \( P_a \cup P_b \) and the optimal radius is \( x/2 \).

If we take a random sample \( S \) of size \( m = o(n) \) from \( P \), with high probability, \( P_a \cap S = \emptyset \) (even if we repeat our sampling a constant number of times, the single point \( P_a \) will still be missing with high probability). Therefore, \( S \) only contains the points from \( P_b \) and \( P_c \). If we run an existing algorithm on \( S \), the resulting ball center, say \( a \), should always be inside the convex hull of \( S \), i.e., the segment \( \overline{q_a q_c} \).
Let $B(o, r)$ be the ball covering $(1 - \gamma)n$ points of $P$. We consider two cases: (1) $q_a \in B(o, r)$ and (2) $q_a \notin B(o, r)$. For case (1), since $o \in MEB$, it is easy to know that the radius $r \geq ||q_a - q_o|| = x$ and therefore the approximation ratio is at least $2$. For case (2), since $|P_b| = (1 - \gamma)n - 1$, $B(o, r)$ must cover some point from $P_b$ and therefore $r = y/2$; because $x \ll y$, the approximation ratio is also larger than 2.

7 Proof of Theorem $[1]$

Similar to the analysis in $[11]$, we let $\lambda_i = \frac{r}{(1 + \epsilon)\text{Rad}(P)}$. Because $r_i$ is the radius of $MEB(T)$ and $T \subset P$, we know $r_i \leq \text{Rad}(P)$ and then $\lambda_i \leq 1/(1 + \epsilon)$. By simple calculation, we know that when $L_i = \frac{(1 + \epsilon)\text{Rad}(P) - \xi_i}{2}$, the lower bound of $r_{i+1}$ in (3) achieves the minimum value. Plugging this value of $L_i$ into (3), we have

$$\lambda_{i+1}^2 \geq \lambda_i^2 + \frac{(1 - \xi\lambda_i)^2 - \lambda_i^2}{2}.$$

(36)

To simplify inequality (36), we consider the function $g(x) = \frac{(1 - x)^2 - \lambda_i^2}{1 - x}$, where $0 < x < \xi$. Its derivative $g'(x) = -1 - \frac{\lambda_i^2}{(1 - x)^2}$ is always negative, thus we have

$$g(x) \geq g(\xi) = \frac{(1 - \xi)^2 - \lambda_i^2}{1 - \xi}. \tag{37}$$

Because $\xi < \frac{1}{1 + \epsilon}$ and $\lambda_i \leq 1/(1 + \epsilon)$, we know that the right-hand side of (37) is always non-negative. Using (37), the inequality (36) can be simplified to

$$\lambda_{i+1}^2 \geq \lambda_i^2 + \frac{1}{4} (g(\xi))^2$$

$$= \lambda_i^2 + \frac{(1 - \xi)^2 - \lambda_i^2}{4(1 - \xi)^2}.$$

(38)

(38) can be further rewritten as

$$\frac{\lambda_{i+1}}{1 - \xi} \geq \frac{1}{2} \left(1 + \left(\frac{\lambda_i}{1 - \xi}\right)^2\right).$$

(39)

Now, we can apply a similar transformation of $\lambda_i$ which was used in $[11]$. Let $\gamma_i = \frac{1}{1 - \frac{\lambda_i}{1 - \epsilon}}$. We know $\gamma_i > 1$ (note $0 \leq \lambda_i \leq \frac{1}{1 + \epsilon}$ and $\xi < \frac{\epsilon}{1 + \epsilon}$). Then, (39) implies that

$$\gamma_{i+1} \geq \frac{\gamma_i}{1 - \frac{2}{\gamma_i}}$$

$$= \gamma_i \left(1 + \frac{1}{\gamma_i^2} + \left(\frac{1}{2\gamma_i^2}\right)^2 + \cdots\right)$$

$$\geq \gamma_i + \frac{1}{2}.$$

(40)

where the equation comes from the fact that $\gamma_i > 1$ and thus $\frac{1}{\gamma_i^2} \in (0, \frac{1}{4})$. Note that $\lambda_0 = 0$ and thus $\gamma_0 = 1$. As a consequence, we have $\gamma_i > 1 + \frac{i}{2}$. In addition, since $\lambda_i \leq \frac{1}{1 + \epsilon}$, that is, $\gamma_i \leq \frac{1}{1 - \frac{1}{1 + \epsilon}}$, we have

$$i \leq \frac{2}{\epsilon - \xi - \epsilon \xi} = \frac{2}{(1 - \frac{\epsilon}{1 + \epsilon})\epsilon}.$$

Consequently, we obtain the theorem.
8 k-Center Clustering with Outliers

Let \( \gamma \in (0, 1) \). Given a set \( P \) of \( n \) points in \( \mathbb{R}^d \), the problem of \( k \)-center clustering with outliers is to find \( k \) balls to cover \( (1 - \gamma)n \) points, and the maximum radius of the balls is minimized (w.l.o.g., we can assume that the \( k \) balls have the same radius). Given an instance \((P, \gamma)\), let \( \{C_1, \ldots, C_k\} \) be the \( k \) clusters forming \( P_{opt} \) (the subset of \( P \) yielding the optimal solution), and \( r_{opt} \) be the optimal radius; that is, each \( C_j \) is covered by an individual ball with radius \( r_{opt} \). Similar to Section 4, we first introduce a linear time algorithm, and then show how to modify it to be sub-linear time by using Lemma 5 and 6.

**Linear time algorithm.** Our algorithm in Section 4.1 can be generalized to be a linear time bi-criteria algorithm for the problem of \( k \)-center clustering with outliers, if \( k \) is assumed to be a constant. Our idea is as follows. In Algorithm 1, we maintain a set \( T \) as the core-set of \( P_{opt} \); here, we instead maintain \( k \) sets \( T_1, T_2, \ldots, T_k \) as the core-sets of \( C_1, C_2, \ldots, C_k \), respectively. Consequently, each \( T_j \) for \( 1 \leq j \leq k \) has an approximate MEB center \( o'_j \) in the \( i \)-th round of Step 3, and we let \( O_i = \{o'_1, \ldots, o'_k\} \). Initially, \( O_0 \) and \( T_j \) for \( 1 \leq j \leq k \) are all empty; we randomly select a point \( p \in P \), and with probability \( 1 - \gamma \), \( p \in P_{opt} \) (w.l.o.g., we assume \( p \in C_1 \) and add it to \( T_1 \); thus \( O_1 = \{p\} \) after this step). We let \( Q \) be the set of the farthest \( t = (1 + \delta)\gamma n \) points to \( O_i \), and \( l_i \) be the \((t + 1)\)-th largest distance from \( P \) to \( O_i \) (the distance from a point \( p \in P \) to \( O_i \) is \( \min_{1 \leq j \leq k} \|p - o'_j\| \)). Then, we randomly select a point \( q \in Q \), and with probability \( \frac{\gamma}{t + 1} \) \( q \in P_{opt} \) (as (22) in Lemma 3). For ease of presentation, we assume that \( q \in P_{opt} \) happens and we have an “oracle” to guess which optimal cluster \( q \) belongs to, say \( q \in C_j \); then, we add \( q \) to \( T_j \) and update the approximate MEB center of \( T_j \). Since each optimal cluster \( C_j \) for \( 1 \leq j \leq k \) has the core-set with size \( \frac{2}{\gamma} + 1 \) \((\text{by setting } s = \frac{2}{\gamma} \text{ in Theorem 1})\), adding at most \( k(\frac{2}{\gamma} + 1) \) points, the distance \( l_i \) will be smaller than \((1 + \delta)r_{opt} \). Consequently, a \((1 + \epsilon, 1 + \delta)\)-approximation solution is obtained when \( i \geq k(\frac{2}{\gamma} + 1) \). *Note* that some “small” clusters could be missing from the above random sampling based approach and therefore \( |O_i| \) could be less than \( k \); however, it can be guaranteed that the total number of missing outliers is at most \( \delta \gamma n \), i.e., a \((1 + \epsilon, 1 + \delta)\)-approximation is always guaranteed (otherwise, the ratio \( \frac{|P_{opt} \cap Q|}{|Q|} > \frac{\delta}{1 + \delta} \) and we can continue to sample a point from \( P_{opt} \) so as to update \( O_i \)).

To remove the oracle for guessing the cluster containing \( q \), we can enumerate all the possible \( k \) cases; since we add \( k(\frac{2}{\gamma} + 1) \) points to \( T_1, T_2, \ldots, T_k \), it generates \( k(\frac{2}{\gamma} + 1) = 2k \log k(\frac{2}{\gamma} + 1) \) solutions in total, and at least one yields a \((1 + \epsilon, 1 + \delta)\)-approximation with probability \((1 - \gamma)(\frac{\delta}{1 + \delta})^{k(\frac{2}{\gamma} + 1)} \) (by the same manner for proving Theorem 2).

**Theorem 4.** Let \((P, \gamma)\) be an instance of \( k\)-center clustering with outliers. Given two parameters \( \epsilon, \delta \in (0, 1) \), there exists an algorithm that outputs a \((1 + \epsilon, 1 + \delta)\)-approximation with probability \((1 - \gamma)(\frac{\delta}{1 + \delta})^{k(\frac{2}{\gamma} + 1)} \).

The running time is \( O(k \log k(\frac{2}{\gamma} + 1)n + \frac{1}{\gamma}d) \).

If one repeatedly runs the algorithm \( O(\frac{1}{1 - \gamma}(1 + \frac{1}{\gamma})^{k(\frac{2}{\gamma} + 1)}) \) times, with constant probability, the algorithm outputs a \((1 + \epsilon, 1 + \delta)\)-approximation solution.

Similar to our discussion on the running time for MEB with outliers in Section 4.1 Bádoiu et al. [13] also achieved a linear time bi-criteria approximation for the \( k\)-center clustering with outliers problem (see Section 4 in their paper). However, the hidden constant of their running time is exponential in \( (\frac{k}{\mu})^{O(1)} \) (where \( \mu \) is defined in [13], and should be \( \delta \gamma \) to ensure a \((1 + \epsilon, 1 + \delta)\)-approximation) that is much larger than \( k \log k(\frac{2}{\gamma} + 1) \) in Theorem 4.

**Sub-linear time algorithm.** The linear time algorithm can be further improved to be sub-linear time; the idea is similar to that for designing sub-linear time algorithm for MEB with outliers in Section 4.2. First, we follow Definition 3 and define the shape set \( X \), where each \( x \in X \) is union of \( k \) balls in the space; the center \( c(x) \) should be the set of its \( k \) ball centers, say \( c(x) = \{o'_1, o'_2, \ldots, o'_k\} \), and the size \( s(x) \) is the radius, i.e., \( x = \bigcup_{j=1}^{k} B(o'_j, s(x)) \). Obviously, if \( x \) is a feasible solution for the instance \((P, \gamma, \frac{P \cap \bigcup_{j=1}^{k} B(o'_j, s(x))}{\delta}) \) should be at least \((1 - \gamma)n \). Also, define the distance function \( f(c(x), p) = \min_{1 \leq j \leq k} \|p - o'_j\| \). It is easy to verify that the shape set \( X \) satisfies Property 1 and 3 defined in Section 3. From Lemma 5, we know that it is possible to obtain a point in \( P_{opt} \cap Q \) with probability \((1 - \eta_1)(\frac{\delta}{1 + \delta})^{k(\frac{2}{\gamma} + 1)} \). Further, we can estimate the value \( l_i \) and select the best candidate solution based on Lemma 6. Overall, we have the following theorem.

**Theorem 5.** Let \((P, \gamma)\) be an instance of \( k\)-center clustering with outliers. Given the parameters \( \epsilon, \delta, \eta_1, \eta_2 \in (0, 1) \), there exists an algorithm that outputs a \((1 + \epsilon, 1 + O(\delta))\)-approximation with probability \((1 - \gamma)(1 - \eta_1)(1 - \eta_2)(\frac{\delta}{1 + \delta})^{k(\frac{2}{\gamma} + 1)} \). The running time is \( O(k \log k(\frac{2}{\gamma} + 1)(\frac{2}{\gamma} + 1)n + \frac{1}{\gamma}d) \).
If one repeatedly runs the algorithm $N = O\left(\frac{1}{\epsilon^2}(\frac{1}{\eta_1}(3 + \frac{3}{\epsilon}))^{k(\frac{2}{1}+1)}\right)$ times with setting $\eta_2 = O\left(\frac{1}{2^{\eta_1}+\epsilon+\frac{1}{\epsilon^{\eta_1}}}\right)$, with constant probability, the algorithm outputs a $(1 + \epsilon, 1 + O(\delta))$-approximation solution.

9 Flat Fitting with Outliers

Let $j$ be a fixed integer between 0 and $d$. Given a $j$-dimensional flat $F$ and a point $p \in \mathbb{R}^d$, we define their distance, $dist(F, p)$, to be the Euclidean distance from $p$ to its projection onto $F$. Let $P$ be a set of $n$ points in $\mathbb{R}^d$. The problem of flat fitting is to find the $j$-dimensional flat $F$ that minimizes $\max_{p \in P} dist(F, p)$. It is easy to see that the MEB problem is the case $j = 0$ of the flat fitting problem. Moreover, given a parameter $\gamma \in (0, 1)$, the flat fitting with outliers problem is to find a subset $P' \subset P$ with size $(1 - \gamma)n$ such that $\max_{p \in P'} dist(F, p)$ is minimized. Similar to MEB with outliers, we also use $P_{\text{opt}}$ to denote the optimal subset. Before presenting our algorithms for flat fitting with outliers, we first introduce the linear time algorithm from Har-Peled and Varadarajan [42] for the vanilla version (without outliers).

We start from the case $j = 1$, i.e., the flat $F$ is a line in the space. Roughly speaking, their algorithm is an iterative procedure to update the solution round by round, until it is close enough to the optimal line $l_{\text{opt}}$. There are two parts in the algorithm. (1) It picks an arbitrary point $p_\Delta \in P$ and let $q_\Delta$ be the farthest point of $P$ from $p_\Delta$: it can be proved that the line passing through $p_\Delta$ and $q_\Delta$, denoted as $l_\Delta$, is a good initial solution that yields a 4-approximation with respect to the objective function. (2) In each of the following rounds, the algorithm updates the solution from $l_{i-1}$ to $l_i$ where $i \geq 1$ is the current number of rounds: let $p_i$ be the farthest point of $P$ from $l_{i-1}$ and let $h_i$ denote the 2-dimensional flat spanned by $p_i$ and $l_{i-1}$; then the algorithm computes a set of $O\left(\frac{1}{\epsilon}|\log \frac{1}{\epsilon}|\right)$ lines on $h_i$, and picks one of them as $l_i$ via an “oracle”. They proved that the improvement from $l_{i-1}$ to $l_i$ is significant enough; thus, after running $\nu = O\left(\frac{1}{\epsilon}|\log \frac{1}{\epsilon}|\right)$ rounds, it is able to achieve a $(1 + \epsilon)$-approximation. To remove the “oracle”, the algorithm can enumerate all the $O\left(\frac{1}{\epsilon}|\log \frac{1}{\epsilon}|\right)$ lines on $h_i$, and thus the total running time is $O\left(2^{\left|\log \frac{1}{\epsilon}\right|} j n \right)$.

Linear time algorithm. Now we consider to adapt the above algorithm to the case with outliers, where in fact the idea is similar to the idea proposed in Section 4.1 for MEB with outliers. For simplicity, we still use the same notations as above. Consider the part (1) first. If we randomly pick a point $p_\Delta$ from $P$, with probability $1 - \gamma$, it belongs to $P_{\text{opt}}$; further, we randomly pick a point, denoted as $q_\Delta$, from the set of $(1 + \delta_0)\gamma n$ farthest points of $P$ from $p_\Delta$, where the value of $\delta_0$ will be determined below. Obviously, with probability $\frac{\delta_0}{1 + \delta_0}, q_\Delta \in P_{\text{opt}}$. Denote by $P_0 = \{p \in P_{\text{opt}} | ||p - p_\Delta|| \leq ||q_\Delta - p_\Delta||\}$. Then we have the following lemma.

Lemma 7. Denote by $l_0$ the line passing through $p_\Delta$ and $q_\Delta$. Then, with probability $(1 - \gamma)(\frac{\delta_0}{1 + \delta_0})$,

$$\max_{p \in P_0} dist(l_0, p) \leq 4 \max_{p \in P_0} dist(l_{\text{opt}}, p) \leq 4 \max_{p \in P_{\text{opt}}} dist(l_{\text{opt}}, p).$$

Also, the size of $P_0$ is at least $(1 - (1 + \delta_0)\gamma)n$.

It is straightforward to obtain the size of $P_0$. The inequality (42) directly comes from the aforementioned result of [42], as long as $p_\Delta$ and $q_\Delta \in P_{\text{opt}}$. So we can use the line $l_0$ as our initial solution. Then, we can apply the same random sampling idea to select the point $p_i$ in the $i$-th round. Namely, we randomly pick a point as $p_i$ from the set of $(1 + \delta_2)\gamma n$ farthest points of $P$ from $l_i$. Moreover, we need to shrink the set $P_{i-1}$ to $P_i = \{p \in P_{i-1} | dist(l_{i-1}, p) \leq dist(l_{i-1}, p_i)\}$. Similar to Lemma 7, we can show that the improvement from $l_{i-1}$ to $l_i$ is significant enough with probability $(1 - \gamma)(\frac{\delta_0}{1 + \delta_0})^{i+1}$, and the size of $P_i$ is at least $(1 - (1 + (i + 1)\delta_0)\gamma)n$. After running $\nu$ rounds, we obtain the line $l_{\nu}$ such that $\max_{p \in P_{\nu}} dist(l_{\nu}, p) \leq (1 + \epsilon)\max_{p \in P_{\text{opt}}} dist(l_{\text{opt}}, p)$, and $|P_{\nu}| \geq (1 - (1 + (\nu + 1)\delta_0)\gamma)n$. So if we set $\delta_0 = \frac{4}{\epsilon^{\gamma_0}}$ with a given $\delta \in (0, 1)$, the line $l_{\nu}$ will be a bi-criteria $(1 + \epsilon, 1 + \delta)$-approximation of the instance $(P, \gamma)$. By using the idea in [42], we can extend the result to the case $j > 1$ with $\nu = \frac{\nu_0(\gamma)}{\epsilon^{\gamma_0}} |\log \frac{1}{\epsilon}|$. We refer the reader to [42] for more details.

Theorem 6. Let $(P, \gamma)$ be an instance of $j$-dimensional flat fitting with outliers. Given two parameters $\epsilon, \delta \in (0, 1)$, there exists an algorithm that outputs a $(1 + \epsilon, 1 + \delta)$-approximation with probability $(1 - \gamma)(\frac{1}{\epsilon})^{g(j, \epsilon)}$ where $g(j, \epsilon) = \text{poly}(\epsilon^{O(j^2)}, \frac{1}{\epsilon^2})$. The running time is $O\left(2^{\left|\log \frac{1}{\epsilon}\right|} j n \right)$ where $g'(j, \epsilon) = \text{poly}(\epsilon^{O(j^2)}, \frac{1}{\epsilon^2})$.

If one repeatedly runs the algorithm $2^{\left|\log \frac{1}{\epsilon}\right|}$ times, with constant probability, the algorithm outputs a $(1 + \epsilon, 1 + \delta)$-approximation solution.
Sub-linear time algorithm. We can view the flat fitting with outliers problem as an MEX with outliers problem. Let $r \geq 0$ and $F$ be a $j$-dimensional flat. Then we can define a $j$-dimensional “slab” $SL(F, r) = \{ p \in \mathbb{R}^d \mid \text{dist}(F, p) \leq r \}$, where its “center” and “size” are $F$ and $r$ respectively (e.g., a ball is a 0-dimensional slab); the distance function $f(F, p) = \text{dist}(F, p)$. It is easy to see that the shape set of slabs satisfies Property $\text{P1}$ and $\text{P2}$ defined in Section 6. Furthermore, finding the optimal flat is equivalent to finding the smallest slab covering $(1 - \gamma)n$ points of $P$. Therefore, by using Lemma $5$ and $6$ we achieve the following theorem.

Theorem 7. Let $(P, \gamma)$ be an instance of $j$-dimensional flat fitting with outliers. Given the parameters $\epsilon, \delta, \eta_1, \eta_2 \in (0, 1)$, there exists an algorithm that outputs a $(1 + \epsilon, 1 + O(\delta))$-approximation with probability $(1 - \gamma)\left(1 - \eta_1\right)(1 - \eta_2)\frac{\left(3 + \frac{3}{2}\right)^{g(j, \epsilon)}}{\delta^2}$ where $g(j, \epsilon) = \text{poly}((\epsilon^j)^{O(\epsilon^j)^{1/\epsilon}}, \frac{1}{\eta_1}, \frac{1}{\eta_2}, \frac{1}{\delta})$. The running time is $O(2^{(\epsilon, \delta, \gamma)n})$ where $g(j, \epsilon) = \text{poly}((\epsilon^j)^{O(\epsilon^j)^{1/\epsilon}}, \frac{1}{\eta_1}, \frac{1}{\eta_2}, \frac{1}{\delta})$. If one repeatedly runs the algorithm $N = O\left(\frac{1}{\mathcal{E}/\epsilon} \left(\frac{1}{\eta_1^{n}} (3 + \frac{3}{2})^{g(j, \epsilon)}\right)\right)$ times with setting $\eta_2 = O(\frac{1}{2^{(\epsilon, \delta, \gamma)n}})$, with constant probability, the algorithm outputs a $(1 + \epsilon, 1 + O(\delta))$-approximation solution.

10 Support Vector Machine with Outliers

In practice, datasets often contain outliers. The separating margin of SVM could be considerably deteriorated by outliers. As mentioned in [27], most of existing techniques [67,72] for SVM outliers removal are numerical approaches (e.g., adding some penalty item to the objective function), and only can guarantee local optimimums. Ding and Xu [27] modeled SVM with outliers as a combinatorial optimization problem and provided an algorithm called “Random Gradient Descent Tree”. We focus on one-class SVM with outliers first, and explain the extension for two-class SVM with outliers at the end of this section.

10.1 One-class SVM with Outliers

Below is the definition of the one-class SVM with outliers problem proposed in [27].

Definition 6 (One-class SVM with Outliers). Given a set $P$ of $n$ points in $\mathbb{R}^d$ and a small parameter $\gamma \in (0, 1)$, the one-class SVM with outliers problem is to find a subset $P' \subset P$ with size $(1 - \gamma)n$ and a hyperplane $H$ separating the origin $o$ and $P'$, such that the distance between $o$ and $H$ is maximized.

Linear time algorithm. We briefly overview the algorithm of [27]. They also considered the “bicriteria approximation” with two small parameters $\epsilon, \delta \in (0, 1)$: a hyperplane $H$ separates the origin $o$ and a subset $P' \subset P$ with size $(1 - (1 + \delta)\gamma)n$, where the distance between $o$ and $H$ is at least $(1 - \epsilon)$ of the optimum. The idea of [27] is based on the fact that the SVM (without outliers) problem is equivalent to the polynoma distance problem in computational geometry [33].

Let $o$ be the origin and $P$ be a given set of points in $\mathbb{R}^d$. The polytope distance problem is to find a point $q$ inside the convex hull of $P$ so that the distance $|q - o|$ is minimized.

For an instance $P$ of one-class SVM, it can be proved that the vector $q_{opt} - o$, if $q_{opt}$ is the optimal solution for the polytopes distance between $o$ and $P$, is the normal vector of the optimal hyperplane. We refer the reader to [27,33] for more details. The polytope distance problem can be efficiently solved by Gilbert Algorithm [32,33]. For completeness, we present it in Algorithm 3.

Similar to the core-set construction method of MEB in Section 2.1, the algorithm also greedily improves the current solution by selecting some point $p_i$ in each iteration. Let $\rho$ be the polytope distance between $o$ and $P$, $D = \max_{q \in P} |p - q|$, and $E = \frac{D^2}{\eta^2}$. Given $\epsilon \in (0, 1)$, it has been proved that a $(1 - \epsilon)$-approximation of one-class SVM (i.e., a separating margin with the width at least $(1 - \epsilon)$ of the optimum) can be achieved by running Algorithm 3 at most $2\left[2E/\epsilon\right]$ steps [20,33]. To handle outliers, the algorithm of [27] follows the similar intuition of Section 4.1: it replaces the step of greedily selecting the point $p_i$ by randomly sampling a point from a set $Q$, which contains the $(1 + \delta)\gamma n$ points having the smallest projection distances (i.e., the values of the function $|p|/|p|$) in Step 2(a) of Algorithm 3. To achieve a $(1 - \epsilon, 1 + \delta)$-approximation with constant success probability, the algorithm takes $O\left(\frac{D^2}{\epsilon \rho^2 n d}\right)$ time, where $z = O\left(\frac{D^2}{\epsilon \rho^2 n d}\right)$.

Sub-linear time algorithm. We define $X$ to be the set of all the closed half-spaces not covering the origin $o$ in $\mathbb{R}^d$; for each $x \in X$, let $H_x$ be the hyperplane enclosing $x$ and let $h_x$ be the projection of $o$ on $H_x$ (see Figure 5). We suppose that the given instance $(P, \gamma)$ has feasible solution. That is, there
Given two point sets $P_1$ and $P_2$ in $\mathbb{R}^d$ and two small parameters $\gamma_1, \gamma_2 \in (0, 1)$, the two-class SVM with outliers problem is to find two subsets $P'_1 \subset P_1$ and $P'_2 \subset P_2$ with $|P'_1| = (1 - \gamma_1)|P_1|$ and $|P'_2| = (1 - \gamma_2)|P_2|$, and a margin separating $P'_1$ and $P'_2$, such that the width of the margin is maximized.

Fig. 5: An illustration for $\mathcal{H}_x$ and $h_x$. 

**Algorithm 3** Gilbert Algorithm [27, 34]

**Input:** A point-set $P$ in $\mathbb{R}^d$, and $N \in \mathbb{Z}^+$.

**Output:** $v_i$ as an approximate solution of the polytope distance between the origin and $P$.

1. Initialize $i = 1$ and $v_1$ to be the closest point in $P$ to the origin $o$.
2. Iteratively perform the following steps until $i = N$.
   
   (a) Find the point $p_i \in P$ whose orthogonal projection on the supporting line of segment $\overline{ov_i}$ has the closest distance to $o$ (called the projection distance of $p_i$), i.e., $p_i = \arg \min_{p \in P} \langle \frac{p - o}{||p - o||} \rangle$, where $\langle p, v_i \rangle$ is the inner product of $p$ and $v_i$ (see Figure 4).
   
   (b) Let $v_{i+1}$ be the point on segment $\overline{v_i o}$ closest to the origin $o$; update $i = i + 1$.

![Fig. 4: An illustration of step 2 in Algorithm 3](image)

exists at least one half-space $x \in \mathcal{X}$ that the hyperplane $\mathcal{H}_x$ separates the origin $o$ and a subset $P'$ with size $(1 - \gamma)n$. We define the center $c(x) = \frac{h_x}{||h_x||}$; since the MEX with outlier problem in Definition 4 is a minimization problem, we design the size function $s(x) = \frac{1}{||h_x||}$. Obviously, a $(1 - \epsilon)$-approximation of the SVM with outliers problem is equivalent to a $\frac{1}{1-\epsilon}$-approximation with respect to the size function $s(x)$. We also define the distance function $f(c(x), p) = -\langle p, \frac{h_x}{||h_x||} \rangle$. It is easy to verify that the shape set $\mathcal{X}$ satisfies Property 1, 2, and 3.

Recall that Algorithm 3 selects the point $p_i = \arg \min_{p \in P} \{ \frac{p - o}{||p - o||} \}$ in each iteration. Actually, the vector $\frac{p - o}{||p - o||}$ can be viewed as a shape center and $p_i$ is the farthest point to $\frac{p - o}{||p - o||}$ based on the distance function $f(c(x), p)$. Moreover, the set $Q$ mentioned in the previous linear time algorithm actually is the set of the farthest $(1 + \delta)\gamma n$ points from $P$ to $\frac{p - o}{||p - o||}$. Consequently, we can apply Lemma 5 to sample a point from $P_{\text{opt}} \cap Q$, and apply Lemma 6 to estimate the value of $l_i$ for each candidate solution $\frac{v_i}{||v_i||}$. Overall, we can improve the running time of the algorithm of [27] to be independent of $n$.

**Theorem 8.** Let $(P, \gamma)$ be an instance of SVM with outliers. Given the parameters $\epsilon, \delta, \eta_1, \eta_2 \in (0, 1)$, there exists an algorithm that outputs a $(1 - \epsilon, 1 + O(\delta))$-approximation with probability $(1 - \gamma)(1 - \eta_1)(1 - \eta_2)(\frac{\delta}{3(1 + \delta)})^2$ where $z = O(\frac{D^3}{\epsilon^5})$. The running time is $O(\frac{D^3}{\epsilon^5} + \epsilon^2)$.

If one repeatedly runs the algorithm $N = O\left(\frac{1}{1 - \gamma}(1 - \eta_1)\frac{3 \epsilon}{\epsilon(1 + \epsilon)}\right)$ times with setting $\eta_2 = O\left(\frac{1}{\epsilon N}\right)$, with constant probability, the algorithm outputs a $(1 - \epsilon, 1 + O(\delta))$-approximation solution.

**10.2 Two-class SVM with Outliers**

Below is the definition of the two-class SVM with outliers problem proposed in [27].

**Definition 7 (Two-class SVM with Outliers).** Given two point sets $P_1$ and $P_2$ in $\mathbb{R}^d$ and two small parameters $\gamma_1, \gamma_2 \in (0, 1)$, the two-class SVM with outliers problem is to find two subsets $P'_1 \subset P_1$ and $P'_2 \subset P_2$ with $|P'_1| = (1 - \gamma_1)|P_1|$ and $|P'_2| = (1 - \gamma_2)|P_2|$, and a margin separating $P'_1$ and $P'_2$, such that the width of the margin is maximized.
We use $P_1^{\text{opt}}$ and $P_2^{\text{opt}}$, where $|P_1^{\text{opt}}| = (1 - \gamma_1)|P_1|$ and $|P_2^{\text{opt}}| = (1 - \gamma_2)|P_2|$, to denote the subsets of $P_1$ and $P_2$ which are separated by the optimal margin. The ordinary two-class SVM (without outliers) problem is equivalent to computing the polytope distance between the origin $o$ and $\mathcal{M}(P_1, P_2)$, where $\mathcal{M}(P_1, P_2)$ is the Minkowski difference of $P_1$ and $P_2$ [33]. Note that it is not necessary to compute the set $\mathcal{M}(P_1, P_2)$ explicitly. Instead, Algorithm 5 only needs to select one point from $\mathcal{M}(P_1, P_2)$ in each iteration, and overall the running time is still linear in the input size. To deal with two-class SVM with outliers, Ding and Xu [27] slightly modified their algorithm for the case of one-class. In each iteration, it considers two subsets $Q_1 \subset P_1$ and $Q_2 \subset P_2$, which respectively consist of points having the $(1 + \delta)\gamma_1|P_1|$ smallest projection distances among all points in $P_1$ and the $(1 + \delta)\gamma_2|P_2|$ largest projection distances among all points in $P_2$ on the vector $v_i$; then, the algorithm randomly selects two points $p_1^i \in Q_1$ and $p_2^i \in Q_2$, and their difference vector $p_2^i - p_1^i$ will serve as the role of $p_i$ in Step 2(a) of Algorithm 3 to update the current solution $v_i$. This approach yields a $(1 - \epsilon, 1 + \delta)$-approximation in linear time.

Fig. 6: An illustration for two-class SVM. The distances from $o$ to $\mathcal{H}^\perp$ and $\mathcal{H}^\top$ are $s^\perp$ and $s^\top$, respectively. The hyperplanes $\tilde{\mathcal{H}}^\perp$ and $\tilde{\mathcal{H}}^\top$ are the estimations of $\mathcal{H}^\perp$ and $\mathcal{H}^\top$, and the distances from $o$ to them are $\tilde{s}^\perp$ and $\tilde{s}^\top$ respectively.

To improve the algorithm to be sub-linear, we need several modifications on our previous idea for the case of one-class. First, we change the distance function to be:

$$f(p, c) = \begin{cases} 
-\langle p, \frac{h}{||h||} \rangle & \text{if } p \in P_1; \\
\langle p, \frac{h}{||h||} \rangle & \text{if } p \in P_2.
\end{cases}$$

By using this new distance function, we can apply Lemma 5 twiceto obtain the points $p_1^i \in Q_1 \cap P_1^{\text{opt}}$ and $p_2^i \in Q_2 \cap P_2^{\text{opt}}$ separately in sub-linear time. Given a vector (i.e., candidate center) $\frac{v}{||v||}$, assume $\mathcal{H}^\perp$ and $\mathcal{H}^\top$ are the parallel hyperplanes orthogonal to $\frac{v}{||v||}$ that the margin formed by them separates $P_1^i$ and $P_2^i$, where $P_1^i \subset P_1$ and $P_2^i \subset P_2$ with $|P_1^i| = (1 - \gamma_1)|P_1|$ and $|P_2^i| = (1 - \gamma_2)|P_2|$. Without loss of generality, we assume that the origin $o$ is inside the margin. Suppose that the distances from $o$ to $\mathcal{H}^\perp$ and $\mathcal{H}^\top$ are $s^\perp$ and $s^\top$, respectively. Then, we obtain two shapes (closed half-spaces) $x^\perp = (-\frac{v}{||v||}, \frac{1}{s^\perp})$ and $x^\top = (\frac{v}{||v||}, \frac{1}{s^\top})$ with $P_1^i \subset x^\perp$ and $P_2^i \subset x^\top$. Consequently, we can apply Lemma 6 twice to obtain two values $\frac{1}{s^\perp} \leq \frac{1}{s^\perp}$ and $\frac{1}{s^\top} \leq \frac{1}{s^\top}$ with $|P_1 \setminus x(\frac{v}{||v||}, \frac{1}{s^\perp})| \leq (1 + O(\delta))\gamma_1|P_1|$ and $|P_2 \setminus x(\frac{v}{||v||}, \frac{1}{s^\top})| \leq (1 + O(\delta))\gamma_2|P_2|$. Therefore, we can use the value $\tilde{s}^\perp + \tilde{s}^\top$ as an estimation of $s^\perp + s^\top$. See Figure 6 for an illustration. Overall, we can achieve a $(1 - \epsilon, 1 + O(\delta))$-approximation in sub-linear time.