A Differentially Private Linear-Time fPTAS for the Minimum Enclosing Ball Problem

Bar Mahpud **Or Sheffet**
Faculty of Engineering
Bar-Ilan University
Israel
{mahpudb, or.sheffet}@biu.ac.il

Abstract

The Minimum Enclosing Ball (MEB) problem is one of the most fundamental problems in clustering, with applications in operations research, statistics and computational geometry. In this works, we give the first differentially private (DP) fPTAS for the Minimum Enclosing Ball problem, improving both on the runtime and the utility bound of the best known DP-PTAS for the problem, of Ghazi et al [18]. Given n points in $\mathbb{R}^d$ that are covered by the ball $B(\theta_{opt}, r_{opt})$, our simple iterative DP-algorithm returns a ball $B(\theta, r)$ where $r \leq (1+\gamma)r_{opt}$ and which leaves at most $\tilde{O}(\frac{\sqrt{d}}{\gamma^2\epsilon})$ points uncovered in $\tilde{O}(n/\gamma^2)$-time. We also give a local-model version of our algorithm, that leaves at most $\tilde{O}(\frac{\sqrt{nd}}{\gamma^2\epsilon})$ points uncovered, improving on the $n^{0.67}$-bound of Nissim and Stemmer [27] (at the expense of other parameters). In addition, we test our algorithm empirically and discuss future open problems.

1 Introduction and Related Work

One of the fundamental problems in clustering is the Minimum Enclosing Ball (MEB) problem, or the 1-Center problem, in which we are given a dataset $P \subset \mathbb{R}^d$ containing n points, and our goal is to find the smallest possible ball $B(\theta_{opt}, r_{opt})$ that contains P. The MEB problem has applications in various areas of operations research, machine learning, statistics and computational geometry: gap tolerant classifiers [10], tuning Support Vector Machine parameters [11] and Support Vector Clustering [4, 5], k-center clustering [9], solving the approximate 1-cylinder problem [9], computation of spatial hierarchies (e.g., sphere trees [22]), and others [16]. The MEB problem is NP-hard to solve exactly, but it can be solved in linear time in constant dimension [25, 17] and has several fully-Polynomial Time Approximation Schemes (fPTAS) [3, 24] that approximate it to any constant $(1+\gamma)$ in time $O(n/\gamma)$.

But in situations where the data is sensitive in nature, such as addresses, locations or descriptive feature-vectors[1] we run the risk that approximating the data's MEB might leak information about a single individual. Differential privacy [14, 13] (DP) alleviates such a concern as it requires that no single individual has a significant effect on the output. Alas, the MEB problem is highly sensitive in nature, since there exist datasets where the addition/removal of a single datum may affect the MEB significantly.

In contrast, it is evident that for any fixed ball $B(\theta, r)$ the number of input points that B contains changes by no more than one with the addition/removal of any single datum. And so, in DP we

[1] Consider a research in a hospital in which one first runs some regression on each patient's data, and then looks for the spread of all regressors of all patients.

Preprint. Under review.

give *bi-criteria* approximations of the MEB: a ball $B(\theta, r)$ that may leave at most a few points of P uncovered and whose radius is comparable to r_{opt}. The work of [28] returns a $O(\sqrt{\log(n)})$-approximation of the MEB while omitting as few as $\tilde{O}(1/\epsilon)$ points from P, and it was later improved to a $O(1)$-approximation [27]. The work of [18] does give a PTAS for the MEB problem, but their $(1+\gamma)$-approximation may leave $\tilde{O}(\sqrt{d}/\epsilon\gamma^3)$ datapoints uncovered[2] and it runs in $n^{O(1/\gamma^2)}$-time where the constant hidden in the big-O notation is huge; as it leverages on multiple tools that take $\exp(d)$-time to construct, such a almost-perfect lattices and list-decodable covers. It should be noted that all of these works actually study the related problem of 1-cluster in which one is given an additional parameter t and seeks to find the smallest MEB of a subset $Q \subset P$ where $|Q| \geq t$. Lastly (as was first commented in [18], Section D.2.1.), a natural way to approximate the MEB problem is through minimizing the convex hinge-loss $L(\theta, x) = \frac{1}{r}\max\{0, \|x-\theta\| - r\}$ but its utility depends on r (as the utility of DP-ERM scales with the Lipfshitz constant of the loss [2]).

By far, one of the most prominent uses of the DP-approximations of the MEB problem lies in range estimation, as $O(1)$-approximations of the MEB can assist in reducing an a-priori large domain to a ball whose radius is proportional to the diameter of P. This helps in reducing the L_2-sensitivity of problems such as the mean and other distance related queries (e.g. PCA). So for example, if we have $\tilde{\Omega}(\frac{\sqrt{d}}{\gamma\epsilon})$ points in a ball of radius $10r_{opt}$ then a DP-approximation of the data's mean using the Gaussian mechanism (see Section 2) returns a point of distance $\leq \gamma r_{opt}$ to the true mean (a technique that is often applied in a Subsample-and-Aggregate framework [26]). This averaging also gives an efficient $(2+\gamma)$-approximation of the MEB. But it is still unknown whether there exists a DP c-approximation of the MEB for $c < 2$ whose runtime is below, say, n^{100}.

Our Contribution and Organization. In this work, we give the first DP-fPTAS for the MEB problem. Our algorithm is very simple and so is its analysis. As input, we assume the algorithm is run after the algorithms of [27] were already run, and as a "starting point" we have both (a) a real number r_0 which is a 4-approximation of r_{opt}, and (b) a 10-approximation of the MEB itself, namely a ball B such that $P \subset B$,[3] which is centered at a point θ_0 satisfying $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$.[4] It is now our goal to refine these parameters to a $(1+\gamma)$-approximation of the MEB. In fact, we can assume that we have a $(1+\gamma)$-approximation of the value of r_{opt}: we simply iterate over all powers: $\frac{r_0}{4}, \frac{r_0}{4}(1+\gamma), \frac{r_0}{4}(1+\gamma)^2, ..., r_0$ where for each guess of r we apply a privacy preserving procedure returning either a point θ satisfying $P \subset B(\theta, r)$ or $\perp$. In our algorithm we simply use a binary-search over these $O(1/\gamma)$ possible values, in order to save on the privacy-budget.

Now, given θ_0 and some radius-guess r, our goal is to shift θ_0 towards θ_{opt}. So, starting from $\theta^0 = \theta_0$, we repeat this simple iterative procedure: we take the mean μ of the points *uncovered by the current* $B(\theta^t, r)$ and update $\theta^{t+1} \leftarrow \theta^t + \frac{\gamma^2}{2}(\mu - \theta^t)$. We argue that, if $r \geq r_{opt}$ then after $T = O(\gamma^{-2}\log(1/\gamma))$-iterations we get θ^T such that $\|\theta^T - \theta_{opt}\| \leq \gamma r_{opt}$ and therefore have that $P \subset B(\theta^T, (1+\gamma)r)$. The reason can be easily seen from Figure 1 — any point $x \in P$ which is uncovered by the current $B(\theta^t, r)$ must be closer to θ_{opt} than to θ^t, and therefore must have a noticeable projection onto the direction $\theta_{opt} - \theta^t$. Thus, in a Perceptron-like style, making a $\Theta(\gamma^2)$-size step towards this x must push us significantly in the $\theta_{opt} - \theta^t$ direction. We thus prove that if the distance of θ^t from θ_{opt} is large, this update step reduces our distance to θ_{opt}. Note that our proof shows that in the non-private case it suffices to take any uncovered point in order to make this progress, or any convex-combination of the uncovered points.

In the private case, rather than using the true mean of the uncovered points in each iteration, we have to use an approximated mean. So we prove that applying our iterative algorithm with a "noisy" mean works just as well, provided that the distance of the true mean of the uncovered points to the noisy mean does not exceed $O(\gamma r)$. This gives a SQ-style algorithm for approximating the MEB of a bounded distribution, a result which may be of interest by itself. After discussing preliminaries in Section 2, we present both the standard (non-noisy) version of our algorithm and its noisy variation in Section 3,

[2] See Lemmas 59 & 60 in [18]

[3] We can always omit the few input points that may reside outside this ball.

[4] We comment that replacing these 4 and 10 constants with any other constants merely changes the constants in our analysis in a very straight-forward way.

Having established that our algorithm works even with an approximation of the mean, all that is left is just to set the parameters of a privacy preserving algorithm accordingly. To that end we work with the notion of zCDP [7] and apply solely the Gaussian mechanism. Seeing as we need to bound the distance between the true mean of the uncovered points and its DP-approximation by $O(\gamma r)$ in each iteration of our algorithm, it follows that the number of uncovered points must be $\Omega(\sqrt{d}/\gamma\epsilon^t)$ where ϵ^t is the privacy budget of the t^{th}-iteration, or else we halt. And due to the composition theorem of DP it suffices to set $\epsilon^t = O(\epsilon/\sqrt{T})$. This leads to a win-win situation: either we find in some iteration a ball that leaves no more than $\tilde{O}(\sqrt{d}/\gamma^2\epsilon)$ points uncovered, or we complete all T iterations and obtain a ball of radius $\leq (1+\gamma)r$ that covers all of P. The full details of this analysis appear in Section 4. We then repeat this analysis but in the local-model, where each user adds Gaussian noise to her own input point. This leads to a similar analysis incurring a $\sqrt{n}$-larger bounds. The algorithm is detailed in Section 5.

While at the topic of local-model DP (LDP) algorithms, it is worth mentioning that the algorithms of [27], which provide us with a good initial "starting point", do have a LDP-variant. Yet the LDP variants of these algorithms may leave as many as $n^{0.67}$ datapoints uncovered. So in Appendix A we give simple differentially private algorithms (in both the curator- and local-models) that obtain such good θ_0 and r_0. Formally, our LDP-algorithm returns a ball $B(\theta_0, r_0)$ s.t. by projecting all points in P onto $B(\theta_0, r)$ we alter no more than $\tilde{O}(\sqrt{d}/\epsilon)$ points and obtain $P' \subset B(\theta_0, r_0)$ where $r_0 \leq 6r_{opt}(P')$. Thus, combining our LDP algorithm for finding a good starting point together with the algorithm of Section 5 we get an overall $(1+\gamma)$-approximation of the MEB in the local model which may omit / alter as many as $\tilde{O}(\sqrt{nd}/\gamma^2\epsilon)$-points. We comment that while this improves on the previously best-known LDP algorithm's bound of $n^{0.67}$, our algorithm's dependency on parameters such as the dimension d or grid-size[5] is worse, and furthermore – that the analysis of [27] (i) relates to the problem of 1-cluster (finding a cluster containing $t \leq n$ many points) and (ii) separates between the required cluster size and the number of omitted points (which is much smaller and only logarithmic in d), two aspects that are not covered in our work.

Lastly, we provide empirical evaluations of our algorithm in Section 6 showing a rather ubiquitous performance across multiple datasets, and discuss open problems in Section 7.

2 Preliminaries

Notation. Given a vector $v \in \mathbb{R}^d$ we denote its L_2-norm as $\|v\|$, and also use $\langle v, u\rangle$ to denote the dot-product between two d-dimensional vectors u and v. A (closed) ball $B(\theta, r)$ is the set of all points $B(\theta, r) = \{x \in \mathbb{R}^d : \|x - \theta\| \leq r\}$. We use $\tilde{O}(\cdot)$ / $\tilde{\Omega}(\cdot)$ to denote big-O / big-Ω dependency up to poly log factors. We comment that in our work we made no effort to optimize constants.

The Gaussian and χ^2_d-Distributions. Given two parameters $\mu \in \mathbb{R}$ and $\sigma^2 > 0$ we denote $\mathcal{N}(\mu, \sigma^2)$ as the Gaussian distribution whose PDF at a point $x \in \mathbb{R}$ is $(2\pi\sigma^2)^{0.5}\exp(-\frac{(x-\mu)^2}{2\sigma^2})$. Standard concentration bounds give that for any $x > 1$ the probability $\Pr_{X\sim\mathcal{N}(\mu,\sigma^2)}[|X - \mu| \geq x\sigma] \leq 2\exp(-x^2/2)$. It is well-known that given two independent random variable $X \sim \mathcal{N}(\mu_1, \sigma_1^2)$ and $Y \sim \mathcal{N}(\mu_2, \sigma_2^2)$ their sum is distributed like a Gaussian $X + Y \sim \mathcal{N}(\mu_1 + \mu_2, \sigma_1^2 + \sigma_2^2)$. We also denote $\mathcal{N}(v, \sigma^2 I_d)$ as the distribution over d-dimensional vectors where each coordinate j is drawn i.i.d. from $\mathcal{N}(v_j, \sigma^2)$. Given $X \sim \mathcal{N}(0, \sigma^2 I_d)$ it is known that $\|X\|^2$ is distributed like a χ^2_d-distribution; and known concentration bounds on the χ^2_d-distribution give that for any $x > 1$ the probability $\Pr_{X\sim\mathcal{N}(0,\sigma^2 I_d)}[\|X\|^2 > \sigma^2(\sqrt{d} + x)^2] \leq \exp(-x^2/2)$.

Differential Privacy. Given a domain $\mathcal{X}$, two multi-sets $P, P' \in \mathcal{X}^n$ are called *neighbors* if they differ on a single entry. An algorithm (alternatively, mechanism) $\mathcal{M}$ is said to be (ϵ, δ)-*differentially private* (DP) [14, 13] if for any two neighboring P, P' and any set S of possible outputs we have: $\Pr[\mathcal{M}(P) \in S] \leq e^\epsilon \Pr[\mathcal{M}(P') \in S] + \delta$.

An algorithm is said to be ρ-zero concentrated differentially privacy (zCDP) [7] if for and two neighboring P and P' and any $\alpha > 1$, the α-Réyni divergence between the output distribution of

[5] It is known [6] that in order to approximate the MEB the input points must lie on some prespecified finite grid.

$\mathcal{M}(P)$ and of $\mathcal{M}(P')$ is upper bounded by $\alpha\rho$, namely

$$\forall\alpha > 1,\ \ \frac{1}{\alpha-1}\log\left(\mathop{\mathbb{E}}_{x\sim\mathcal{M}(P')}\left[\left(\frac{\mathsf{PDF}[\mathcal{M}(P)=x]}{\mathsf{PDF}[\mathcal{M}(P')=x]}\right)^{\alpha}\right]\right) \leq \alpha\rho$$

It is a well-known fact that the composition of two ρ-zCDP mechanisms is 2ρ-zCDP. It is also known that given a function $f : \mathcal{X}^n \to \mathbb{R}^d$ whose L_2-global sensitivity is $\max_{P\sim P'}\|f(P)-f(P')\|_2 \leq G$ then the Gaussian mechanism that returns $f(D)+X$ where $X \sim \mathcal{N}(0, \frac{G^2}{2\rho}I_d)$ is ρ-zCDP. Lastly, it is known that any ρ-zCDP mechanism is (ϵ,δ)-DP for any $\delta < 1$ and $\epsilon = \rho + \sqrt{4\rho\ln(1/\delta)}$. This suggests that given $\epsilon \leq 1$ and $\delta \leq e^{-2}$ it suffices to use a ρ-zCDP mechanis with $\rho \leq \frac{\epsilon^2}{5\ln(1/\delta)}$.

The *Local-Model* of DP: while standard algorithms in DP assume the existence of a trusted curator who has access to the raw data, in the local-model of DP no such curator exists. While the formal definition of the local-model involves the notion of protocols (see [31] for a formal definition), for the context of this work it suffices to say each respondent randomized her own messages so that altogether they preserve ρ-zCDP.

3 A Non-Private fPTAS for the MEB Problem

In this section we give our non-private algorithm. We first analyze it assuming no noise – namely, in each iteration we use the precise mean of the points that do not reside inside the ball $B(\theta^t, r)$. Later, in Section 3.1 we discuss a version of this algorithm in which rather than getting the exact mean, we get a point which is sufficiently close to the mean.

Algorithm 1 Non-Private Minimum Enclosing Ball

```
Input: a set of $n$ points $P \subseteq \mathbb{R}^d$, an approximation parameter $\gamma \in (0,1)$,
  an initial radius $r_0$ s.t. $r_{opt} \leq r_0 \leq 4r_{opt}$, and an initial center $\theta_0$ s.t. $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$.
Set $i_{\min} \leftarrow 0$, $i_{\max} \leftarrow \ln_{1+\gamma}(4)(\approx \frac{4}{\gamma})$, and $\theta^* \leftarrow \theta_0$.
while ($i_{\min} < i_{\max}$) do
    $i_{cur} = \lfloor \frac{i_{\min}+i_{\max}}{2} \rfloor$
    $r_{cur} \leftarrow (1+\gamma)^{i_{cur}} \cdot r_0/4$
    $\theta_{cur} \leftarrow$ MMEB$(P, \gamma, r_{cur}, \theta_0)$
    if $P \subset B(\theta_{cut}, (1+\gamma)r_{cur})$ then
        Set $i_{\max} \leftarrow i_{cur}$, $\theta^* \leftarrow \theta_{cur}$ and $r^* \leftarrow (1+\gamma)r_{cur}$
    else
        $i_{\min} \leftarrow i_{cur} + 1$
return $B(\theta^*, r^*)$
```

Algorithm 2 Margin based Minimum Enclosing Ball (MMEB)

```
Input: a set of $n$ points $P \subseteq \mathbb{R}^d$, an approximation parameter $\gamma \in (0,1)$,
  a candidate radius $r$, and an initial center $\theta_0$ s.t. $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$.
Set $T \leftarrow \frac{4}{\gamma^2}\ln(\frac{100}{\gamma^2})$, and $\theta^0 = \theta_0$.
for $t = 0, 1, 2, \ldots, T-1$ do
    if ($\{x \in P : x \notin B(\theta^t, r)\} = \emptyset$) then return $\theta^t$
    else
        Set $n_w^t \leftarrow |\{x \in P : x \notin B(\theta^t, r)\}|$ and $\mu_w^t \leftarrow \frac{1}{n_w^t}\sum_{x\notin B(\theta^t,r)} x$
        Update $\theta^{t+1} \leftarrow (1-\frac{\gamma^2}{2})\theta^t + \frac{\gamma^2}{2}\mu_w^t$
return $\theta^T$
```

Theorem 3.1. *For any $P \subset \mathbb{R}^d$, denote $B(\theta_{opt}, r_{opt})$ as the MEB of P. Then Algorithm 1 returns a ball $B(\theta, r)$ where $P \subset B(\theta, r)$ and $r \leq (1+3\gamma)r_{opt}$.*

At the core of the proof of Theorem 3.1 lies the following lemma.

Lemma 3.2. Applying Algorithm 2 with any $r \geq r_{opt}$ and any θ_0 where $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$ we obtain a θ where $\|\theta - \theta_{opt}\| \leq \gamma r_{opt}$ in at most T iterations.

Proof of Theorem 3.1. Suppose Lemma 3.2 indeed holds. Then it immediately implies whenever Algorithm 2 is run with $r \geq r_{opt}$ we obtain a point θ where $P \subset B(\theta_{opt}, r_{opt}) \subset B(\theta, (1+\gamma)r_{opt})$. Denote $i^* = \min\{i \in \mathbb{N} : \frac{r_0}{4}(1+\gamma)^i \geq r_{opt}\}$. It is simple to prove inductively that in each iteration of Algorithm 1 we have that $i^* \geq i_{\min}$. Next, call an integer i successful if we obtain for its radius $r_{cur}(i)$ some point θ where $P \subset B(\theta, (1+\gamma)r_{cur}(i))$. Again, it is simple to argue inductively that $i_{\max}$ is always successful. It follows that when the binary search of Algorithm 1 terminates, $i_{\min} = i_{\max}$ and we have a successful i, and so we return a ball of radius $\frac{r_0}{4}(1+\gamma)^{i_{\min}} \cdot (1+\gamma) \leq (1+\gamma)^2 r_{opt} \leq (1+3\gamma)r_{opt}$ which contains all points in P, thus concluding our proof. □

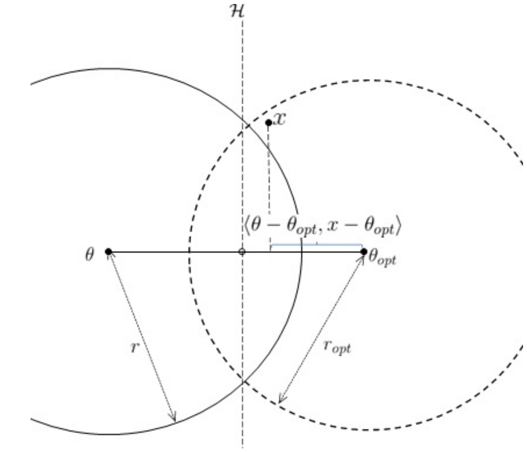


Figure 1: For a point x uncovered by $B(\theta, r)$ where $r \geq r_{opt}$, it must be that x's projection onto the $\overline{\theta\theta_{opt}}$-line is closer to θ_{opt} than to θ.

Thus, all that is left is to prove Lemma 3.2. Its proof, in turn, requires the following claim.

Claim 3.3. Given a set of n points $P \subseteq \mathbb{R}^d$, let $B(\theta_{opt}, r_{opt})$ denote the MEB of P. Let $\theta \in \mathbb{R}^d$ be an arbitrary point, and let r be any real number where $r \geq r_{opt}$. Then for any $x \in P$ s.t. $\|\theta - x\| > r$ it holds that

$$\langle\theta - \theta_{opt}, x - \theta_{opt}\rangle \leq \frac{1}{2}\|\theta - \theta_{opt}\|^2$$

Proof. Let $x \in P$ be a point s.t. $x \notin B(\theta, r)$, as depicted in Figure 1. Let m be the middle point $\frac{\theta+\theta_{opt}}{2}$, and let $\mathcal{H}$ be the hyperplane orthogonal to $\theta - \theta_{opt}$ which passes through m. Denote $\mathcal{H}^+$ as the (open) half-space $\mathcal{H}^+ = \{z \in \mathbb{R}^d : \|z - \theta_{opt}\| < \|z - \theta\|\}$. Therefore $x \in \mathcal{H}^+$ which in turn implies that

$$\langle x - \theta_{opt}, \theta - \theta_{opt}\rangle < \langle m - \theta_{opt}, \theta - \theta_{opt}\rangle = \frac{1}{2}\|\theta - \theta_{opt}\|^2$$ □

We are now ready to prove our main lemma.

Proof of Lemma 3.2. First, we argue that in any iteration t of Algorithm 2 where $\{x \in P : x \notin B(\theta^t, r)\} \neq \emptyset$ it holds that $\|\theta^{t+1} - \theta_{opt}\|^2 \leq (1 - \frac{\gamma^2}{2})\|\theta^t - \theta_{opt}\|^2 + (\frac{\gamma^2}{2})^2 \cdot r_{opt}^2$. That is because by definition

$$\|\theta^{t+1} - \theta_{opt}\|^2 = \left\|\left((1 - \frac{\gamma^2}{2})\theta^t + \frac{\gamma^2}{2}\mu_w^t\right) - \theta_{opt}\right\|^2 = \left\|(1 - \frac{\gamma^2}{2})\left(\theta^t - \theta_{opt}\right) + \frac{\gamma^2}{2}\left(\mu_w^t - \theta_{opt}\right)\right\|^2$$
$$= (1 - \frac{\gamma^2}{2})^2 \cdot \|\theta^t - \theta_{opt}\|^2 + 2\frac{\gamma^2}{2}(1 - \frac{\gamma^2}{2})\langle\theta^t - \theta_{opt}, \mu_{w^t} - \theta_{opt}\rangle + (\frac{\gamma^2}{2})^2 \cdot \|\mu_{w^t} - \theta_{opt}\|^2$$

Claim 3.3 gives that $\langle\theta^t - \theta_{opt}, \mu_w^t - \theta_{opt}\rangle = \frac{1}{n_w^t}\sum_{x \notin B(\theta^t, r)}\langle\theta^t - \theta_{opt}, x - \theta_{opt}\rangle \leq \frac{1}{2}\|\theta^t - \theta_{opt}\|^2$, so

$$\leq (1 - \frac{\gamma^2}{2})^2 \cdot \|\theta^t - \theta_{opt}\|^2 + 2(\frac{\gamma^2}{2} - \frac{\gamma^4}{4}) \cdot \frac{1}{2}\|\theta^t - \theta_{opt}\|^2 + (\frac{\gamma^2}{2})^2 \cdot \|\mu_{w^t} - \theta_{opt}\|^2$$

Lastly note that the ball $B(\theta_{opt}, r_{opt})$ is convex and so

$$\leq (1 - \gamma^2 + \frac{\gamma^4}{4}) \cdot \|\theta^t - \theta_{opt}\|^2 + (\frac{\gamma^2}{2} - \frac{\gamma^4}{4}) \cdot \|\theta^t - \theta_{opt}\|^2 + \frac{\gamma^4}{4} \cdot r_{opt}^2$$
$$\leq (1 - \frac{\gamma^2}{2})\|\theta^t - \theta_{opt}\|^2 + \frac{\gamma^4}{4} \cdot r_{opt}^2 \tag{1}$$

So now, consider any iteration of Algorithm 2 with $r \geq r_{opt}$ and where $\|\theta^t - \theta_{opt}\| \geq \gamma r_{opt}$ and in which we make an update step. Due to Equation (1)

$$\|\theta^{t+1} - \theta_{opt}\|^2 \leq (1 - \frac{\gamma^2}{2})\|\theta^t - \theta_{opt}\|^2 + (\frac{\gamma^2}{2})^2 \cdot r_{opt}^2 \leq (1 - \frac{\gamma^2}{2})\|\theta^t - \theta_{opt}\|^2 + \frac{\gamma^4}{4} \cdot \frac{\|\theta^t - \theta_{opt}\|^2}{\gamma^2}$$
$$= (1 - \frac{\gamma^2}{4})\|\theta^t - \theta_{opt}\|^2 \leq e^{-\frac{\gamma^2}{4}}\|\theta^t - \theta_{opt}\|^2$$

This suggests that after $T = \frac{4}{\gamma^2}\ln(\frac{100}{\gamma^2})$ iterations where $\|\theta^t - \theta_{opt}\| \geq \gamma r_{opt}$ we get that

$$\|\theta^T - \theta_{opt}\|^2 \leq e^{-\frac{T\gamma^2}{4}}\|\theta_0 - \theta_{opt}\|^2 \leq \frac{\gamma^2}{100} \cdot 100 r_{opt}^2 = \gamma^2 r_{opt}^2$$

as required. Now, should it be the case that in some iteration $\|\theta^t - \theta_{opt}\| < \gamma r_{opt}$ and we make an update step. Again, Equation (1) asserts that

$$\|\theta^{t+1} - \theta_{opt}\|^2 \leq (1 - \frac{\gamma^2}{2})\|\theta^t - \theta_{opt}\|^2 + \frac{\gamma^4}{4} \cdot r_{opt}^2 < (1 - \frac{\gamma^2}{2})\gamma^2 r_{opt}^2 + \frac{\gamma^4}{4} \cdot r_{opt}^2 < \gamma^2 r_{opt}^2$$

which suggests that once $\|\theta^t - \theta_{opt}\| < \gamma r_{opt}$ then we have that $\|\theta^\tau - \theta_{opt}\| < \gamma r_{opt}$ for all $\tau \geq t$. □

We comment that non-privately, it is rather simple to obtain a good r_0 and a good starting point θ_0: $r_0 = diam(P)$ which is known to be upper bounded by $2r_{opt}$ and θ_0 can be any $x \in P$ which is within distance r_{opt} from the true center of the MEB of P. Next, we comment that Algorithm 2 runs in time $O(T \cdot n)$ since the averaging of the points in $P \setminus B(\theta^t, r)$ takes $O(n)$-time naïvely. Thus, overall, the runtime of Algorithm 1 is $O(nT\log(1/\gamma)) = O(n\frac{\log^2(1/\gamma)}{\gamma^2})$. Lastly, we comment that in Algorithm 2 we could replace the mean μ_w^t of the uncovered points with any convex combination (even a single $x \notin B(\theta^t, r)$).

3.1 The Noisy/SQ-Version of the fPTAS for the MEB Problem

Now, we consider a scenario where in each iteration t, rather than using the exact mean $\mu_w^t = \frac{\sum_{x \in P \setminus B(\theta^t, r)} x}{|P \setminus B(\theta^t, r)|}$, we obtain an approximated mean $\tilde{\mu}_w^t = \mu_w^t + \Delta^t$ subject to the constraint that $\|\Delta^t\| = O(\gamma r)$. To that end, we modify Algorithm 2 so that our update scale shrinks by a constant factor to $\gamma^2/8$, namely we set $\theta^{t+1} \leftarrow (1 - \frac{\gamma^2}{8})\theta^t + \frac{\gamma^2}{8}\tilde{\mu}_w^t$. We now prove that the revised algorithm still converges to a point close to θ_{opt}.

Lemma 3.4. Applying Algorithm 2 with any $4r_{opt} \geq r \geq r_{opt}$ and any θ_0 where $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$, where in each iteration we use an approximated mean $\tilde{\mu}_w^t = \mu_w^t + \Delta^t$ where $\|\Delta^t\| \leq \frac{\gamma r}{16} \leq \frac{\gamma r_{opt}}{4}$ we obtain a θ where $\|\theta - \theta_{opt}\| \leq \gamma r_{opt}$ in at most $16T = \frac{64}{\gamma^2}\ln(100/\gamma^2)$ iterations.

Proof. First, analogously to Lemma 3.2 we have that in each update step we get

$$\|\theta^{t+1} - \theta_{opt}\|^2 = \left\|\left((1 - \frac{\gamma^2}{8})\theta^t + \frac{\gamma^2}{8}\tilde{\mu}_w^t\right) - \theta_{opt}\right\|^2 = (1 - \frac{\gamma^2}{8})^2 \cdot \|\theta^t - \theta_{opt}\|^2$$
$$+ 2\frac{\gamma^2}{8}(1 - \frac{\gamma^2}{8})\left(\langle\theta^t - \theta_{opt}, \mu_w^t - \theta_{opt}\rangle + \langle\theta^t - \theta_{opt}, \Delta^t\rangle\right) + (\frac{\gamma^2}{8})^2 \cdot \|\mu_w^t - \theta_{opt} + \Delta^t\|^2$$
$$\leq (1 - \frac{\gamma^2}{8})^2 \cdot \|\theta^t - \theta_{opt}\|^2 + 2(\frac{\gamma^2}{8} - \frac{\gamma^4}{64}) \cdot \left(\frac{1}{2}\|\theta^t - \theta_{opt}\|^2 + \|\theta^t - \theta_{opt}\| \cdot \frac{\gamma r_{opt}}{4}\right)$$
$$+ (\frac{\gamma^2}{8})^2 \cdot \left(2\|\mu_w^t - \theta_{opt}\|^2 + 2\frac{\gamma^2 r_{opt}^2}{4^2}\right)$$
$$\leq (1 - \frac{\gamma^2}{8})^2\|\theta^t - \theta_{opt}\|^2 + 2(\frac{\gamma^2}{8} - \frac{\gamma^4}{64}) \cdot \|\theta^t - \theta_{opt}\|\left(\frac{1}{2}\|\theta^t - \theta_{opt}\| + \frac{\gamma r_{opt}}{4}\right) + \frac{3\gamma^4}{64}r_{opt}^2$$

It follows that in each iteration where $\|\theta^t - \theta_{opt}\| \geq \gamma r_{opt}$ we get that

$$\|\theta^{t+1} - \theta_{opt}\|^2 \leq (1 - \frac{2\gamma^2}{8} + \frac{\gamma^4}{64})\|\theta^t - \theta_{opt}\|^2 + 2(\frac{\gamma^2}{8} - \frac{\gamma^4}{64}) \cdot \frac{3}{4}\|\theta - \theta_{opt}\|^2 + \frac{3\gamma^4 r_{opt}^2}{64}$$

$$< (1 - \frac{\gamma^2}{16})\|\theta^t - \theta_{opt}\|^2 + \frac{3\gamma^2}{64}\|\theta^t - \theta_{opt}\|^2 = (1 - \frac{\gamma^2}{64})\|\theta^t - \theta_{opt}\|^2$$

suggesting that after $16T = \frac{64}{\gamma^2}\ln(100/\gamma^2)$ iteration at most it must hold that

$$\|\theta^{16T} - \theta_{opt}\|^2 \leq \exp(-\frac{64}{\gamma^2}\ln(100/\gamma^2) \cdot \frac{\gamma^2}{64})\|\theta_0 - \theta_{opt}\|^2 \leq \frac{\gamma^2}{100} \cdot 100 r_{opt}^2 = \gamma^2 r_{opt}^2$$

As required. Similarly, if at some iteration t it holds that $\|\theta^t - \theta_{opt}\| < \gamma r_{opt}$ then we get that

$$\begin{aligned}\|\theta^{t+1} - \theta_{opt}\|^2 &\leq (1 - \frac{\gamma^2}{8})^2\gamma^2 r_{opt}^2 + 2(\frac{\gamma^2}{8} - \frac{\gamma^4}{64}) \cdot \frac{3}{4}\gamma^2 r_{opt}^2 + \frac{3\gamma^4 r_{opt^2}}{64} \\ &\leq \gamma^2 r_{opt}^2 \left(1 - \frac{2\gamma^2}{8} + \frac{\gamma^4}{64} + \frac{3\gamma^2}{2\cdot 8} - \frac{3\gamma^4}{2\cdot 64} + \frac{3\gamma^2}{64}\right) \leq (1 - \frac{\gamma^2}{64})\gamma^2 r_{opt}^2\end{aligned}$$

suggesting yet again that $\|\theta^\tau - \theta_{opt}\| < \gamma r_{opt}$ for all $\tau \geq t$. □

4 A Differentially Private fPTAS for the MEB Problem

We now turn our attention to the privacy-preserving versions of Algorithms 1 and 2. In this section we give their curator-model ρ-zCDP versions (Algorithms 3 and 4 resp.), whereas in the following section (Section 5) we detail their local-model zCDP versions.

Algorithm 3 Differentially Private Minimum Enclosing Ball (DP-MEB)

Input: a set of n points $P \subseteq \mathbb{R}^d$, an approximation parameter $\gamma \in (0, 1)$, an initial radius r_0 s.t. $r_{opt} \leq r_0 \leq 4r_{opt}$, and an initial center θ_0 s.t. $\|\theta_0 - \theta_{opt}\| \leq 10r_{opt}$, error parameter β and privacy-parameter ρ.

```
Remove any $x \in P$ which doesn't belong to $B(\theta_0, 11r_0)$.
Set $i_{\min} \leftarrow 0$, $i_{\max} \leftarrow \ln_{1+\gamma}(4)(\approx \frac{4}{\gamma})$, and $\theta^* \leftarrow \theta_0$.
Set $B \leftarrow \lceil \log_2 (\ln_{1+\gamma}(4)) \rceil$.
while ($i_{\min} < i_{\max}$) do
    $i_{cur} = \lfloor \frac{i_{\min}+i_{\max}}{2} \rfloor$
    $r_{cur} \leftarrow (1+\gamma)^{i_{cur}} \cdot r_0/4$
    $\theta_{cur} \leftarrow$ DP-MMEB$(P, \gamma, \frac{\beta}{B}, \frac{\rho}{B}, r_{cur}, \theta_0)$
    if ($\theta_{cur} \neq \perp$) then
        Set $i_{\max} \leftarrow i_{cur}$, $\theta^* \leftarrow \theta_{cur}$ and $r^* \leftarrow (1+\gamma)r_{cur}$
    else
        $i_{\min} \leftarrow i_{cur} + 1$
return $B(\theta^*, r^*)$
```

4.1 Privacy Analysis

Lemma 4.1. Algorithm 4 satisfies *ρ-zCDP.*

Proof. At each one of the T iterations of the algorithm, we answer two queries to the input data: a counting query and a summation query. It is known that the L_2*-sensitivity* of a counting query is 1, therefore using the Gaussian mechanism theorem while setting $\sigma^2_{count} = \frac{T+1}{\rho}$ satisfies $\frac{\rho}{2(T+1)}$*-zCDP.* Secondly, we know that all the points are bounded by a ball of radius $11r_0 \leq 44r_{opt} \leq 44r$ around θ_0, hence the summation query has L_2-sensitivity of $\leq 88r$. Thus, by setting $\sigma^2_{sum} = \frac{T(88r)^2}{\rho}$ we have that we answer each summation query using $\frac{\rho}{2T}$-zCDP. Due to sequential composition of zCDP [8], it holds that in all T iteration together we preserve $\left(\rho(1 - \frac{1}{2(T+1)})\right)$-zCDP. Lastly, we apply one last counting query which we answer using the Gaussian mechanism while satisfying $\frac{\rho}{2(T+1)}$-zCDP, thus, overall we are ρ-zCDP. □

Corollary 4.2. Algorithm 3 satisfies *ρ-zCDP.*

Proof. Since Algorithm 3 invokes $B = \lceil \log_2(\log_{1+\gamma}(4)) \rceil$ calls to Algorithm 4 each preserving $\frac{\rho}{B}$-zCDP, Algorithm 3 is ρ-zCDP overall. □

Algorithm 4 DP-Margin based Minimum Enclosing Ball (DP-MMEB)

```
**Input:** a set of $n$ points $P \subseteq \mathbb{R}^d$, an approximation parameter $\gamma \in (0,1)$,
an error parameter $\beta \in (0,1)$, privacy parameter $\rho$,
a candidate raduis $r$, and an initial center $\theta_0$ s.t. $\|\theta_0 - \theta_{opt}\| \le 10r_{opt}$.
Set $\theta^0 \leftarrow \theta_0, T \leftarrow \frac{64}{\gamma^2}\ln(\frac{100}{\gamma^2}), \sigma^2_{count} \leftarrow \frac{T+1}{\rho}$, and $\sigma^2_{sum} \leftarrow \frac{T\cdot(88r)^2}{\rho}$.
for $t = 0, 1, 2, \dots, T-1$ do
    Sample $\Delta_{count} \sim \mathcal{N}(0, \sigma^2_{count})$.
    $\tilde{n^t}_w \leftarrow |\{x \in P : x \notin B(\theta^t, r)\}| + \Delta_{count}$
    if $\left(\tilde{n}^t_w < \frac{16\sqrt{T+1}}{\gamma\sqrt{\rho}}\left(88\sqrt{d} + 110\sqrt{2\ln(4(T+1)/\beta)}\right)\right)$ then
        return $\theta^t$
    Sample $\Delta_{sum} \sim \mathcal{N}(0, \sigma^2_{sum} I_d)$.
    Set $\tilde{\mu}^t_w \leftarrow \frac{1}{\tilde{n}_{w^t}}\left(\sum_{x \notin B(\theta^t, r)} x + \Delta_{sum}\right)$.
    Update $\theta^{t+1} \leftarrow (1 - \frac{\gamma^2}{8})\theta^t + \frac{\gamma^2}{8}\tilde{\mu}_{w^t}$
Sample $\Delta_{count} \sim \mathcal{N}(0, \sigma^2_{count})$.
if $\left( |P \setminus B(\theta^T, (1+\gamma)r)\}| + \Delta_{count} \le \sqrt{\frac{2T\log(4T/\beta)}{\rho}} \right)$ then return $\theta^T$ else return $\perp$
```

4.2 Utility Analysis

Lemma 4.3. W.p. $\ge 1-\beta$, applying Algorithm 4 with $r \ge r_{opt}$ and an initial center θ_0 s.t. $\|\theta_0 - \theta_{opt}\| \le 10r_{opt}$ returns a point θ^t where $|P \setminus B(\theta^t, (1+\gamma)r)| \le \frac{1408\sqrt{Td}+1760\sqrt{2(T+1)\ln(4(T+1)/\beta)}}{\gamma\sqrt{\rho}} + \sqrt{\frac{2(T+1)\log(4(T+1)/\beta)}{\rho}}$.

Proof. Define the events

$$\mathcal{E}_1 := \text{in all } T+1 \text{ draws, } |\Delta_{count}| \le \sqrt{\frac{2(T+1)\log(4(T+1)/\beta)}{\rho}}$$

$$\mathcal{E}_2 := \text{in all } T \text{ draws, } \|\Delta_{sum}\| \le \frac{88r\sqrt{T}}{\sqrt{\rho}}\left(\sqrt{d} + \sqrt{2\ln(4T/\beta)}\right)$$

And standard bounds on the concentration of the Gaussian distribution and the χ^2_d-distribution together with the union-bound assert that $\Pr[\overline{\mathcal{E}_1} \cup \overline{\mathcal{E}_2}] \le \sum_{t=0}^{T} \frac{\beta}{2(T+1)} + \sum_{t=0}^{T-1} \frac{\beta}{2T} \le \beta$. We continue the rest of the proof conditioning on $\mathcal{E}_1 \cap \mathcal{E}_2$ holding.

Thus, under both $\mathcal{E}_1$ and $\mathcal{E}_2$ holding we get

$$\|\tilde{\mu}^t_w - \mu^t_w\| = \|\tilde{\mu}^t_w - \theta^t - (\mu^t_w - \theta^t)\| = \left\| \frac{\sum_{x \notin B(\theta^t,r)}(x-\theta^t) + \Delta_{sum}}{\tilde{n}^t_w} - \frac{\sum_{x\notin B(\theta^t,r)}(x-\theta^t)}{n^t_w} \right\|$$

$$= \left\| \frac{\Delta_{sum}}{\tilde{n}^t_w} - \left(\sum_{x\notin B(\theta^t,r)}(x-\theta^t)\right)\left(\frac{1}{\tilde{n}^t_w} - \frac{1}{n^t_w}\right)\right\| \le \frac{\|\Delta_{sum}\|}{\tilde{n}^t_w} + \|\mu^t_w - \theta^t\|\frac{|\Delta_{count}|}{\tilde{n}^t_w}$$

$$\le \frac{1}{\tilde{n}^t_w\sqrt{\rho}}\left(88r\sqrt{T}\left(\sqrt{d} + \sqrt{2\ln(4T/\beta)}\right) + 22r\sqrt{2(T+1)\log(4(T+1)/\beta)}\right) \le \frac{\gamma r}{16}$$

since $\tilde{n}^t_w \ge 16 \cdot \frac{88\sqrt{Td}+110\sqrt{2(T+1)\ln(4(T+1)/\beta)}}{\gamma\sqrt{\rho}}$ in order for us to make an update.

It thus follows from Lemma 3.4 that if $r \ge r_{opt}$, then one of the two must hold: (i) we do not make an update since $\tilde{n}^t_w$ is below the specified bound, or (ii) we perform all T updates and so $\|\theta^T - \theta_{opt}\| \le \gamma r_{opt}$. In the former case we simply return this θ^t and it must hold under $\mathcal{E}_1$ that

$$|P \setminus B(\theta^t, (1+\gamma)r| \le |P \setminus B(\theta, r)| = n^t_w = \tilde{n}^t_w + \Delta_{count}$$

$$\leq 16 \cdot \frac{88\sqrt{Td} + 110\sqrt{2(T+1)\ln({}^{4(T+1)}/_{\beta})}}{\gamma\sqrt{\rho}} + \sqrt{\frac{2(T+1)\log({}^{4(T+1)}/_{\beta})}{\rho}}$$

and in the latter case $P \subset B(\theta^T, (1+\gamma)r)$ which implies that under $\mathcal{E}_1$ we must pass the last if-condition of Algorithm 4 and return θ^T. Considering the worst of the two cases, the bound of the lemma is proven. □

Corollary 4.4. Given r_0 where $r_{opt} \leq r_0 \leq 4r_{opt}$ and a point θ_0 where $\|\theta_0 - \theta^*\| \leq 10r_{opt}$, w.p. $\geq 1-\beta$ Algorithm 3 is a $O(n \cdot \frac{\log^2(1/\gamma)}{\gamma^2})$-time algorithm that returns a ball $B(\theta^*, r)$ where $r \leq (1+3\gamma)r_{opt}$ and where $|P \setminus B(\theta^*, r^*)| = O(\frac{\left(\sqrt{d}+\sqrt{\log({}^1/_{\gamma\beta})}\right)\log({}^1/_{\gamma})}{\gamma^2\sqrt{\rho}})$.

Proof. The result follows directly from the fact that Algorithm 3 invokes $B = O(\log({}^1/_{\gamma}))$ calls to Algorithm 4, with a privacy budget of $O(\rho/\log({}^1/_{\gamma}))$ each and with a failure probability of $O(\beta/\log({}^1/_{\gamma}))$ each. Plugging those into the bound of Lemma 4.3 together with the fact that $T = O(\gamma^{-2}\log({}^1/_{\gamma}))$ yields the resulting bound. Note that, denoting the "correct" $i^* = \min\{i \geq 0 : \frac{r_0}{4}(1+\gamma)^i \geq r_{opt}\}$, under the event that no invocation of Algorithm 4 fails, each time we execute the binary search with a value of $i_{cur} \geq i^*$ we obtain some $\theta_{cur} \neq \perp$. Due to the nature of the binary search and the fact that upon finding $\theta_{cur} \neq \perp$ we set $i_{\max} = i_{cur}$, it must follows that we return a ball of radius $(1+\gamma)r^* = (1+\gamma) \cdot \frac{r_0}{4} \cdot (1+\gamma)^i$ for some $i \leq i^*$, and so $r^* \leq (1+\gamma)^2 r_{opt} \leq (1+3\gamma)r_{opt}$. Lastly, the runtime of Algorithm 4 is $O(nT)$ making the runtime of Algorithm 3 to be $O(nTB) = O(\frac{n\log^2(1/\gamma)}{\gamma^2})$ as required. □

4.3 Application: Subsample Stable Functions

Much like the work of [18], our work too is applicable as a DP-aggregator in a Subsample-and-Aggregate [26] framework. We say that a point $p \in \mathbb{R}^d$ is (r,β)-*stable* for some function $f : \mathcal{X}^* \to \mathbb{R}^d$ if there exists $m(r,\beta)$ such that for any input $S \subset \mathcal{X}^n$ a random subsample of m entries of S input datapoints returns w.p. $\geq 1-\beta$ a value close to p, namely, $Pr_{S' \subset S, |S|=m}[\|c - f(S')\| \leq r] \geq 1-\beta$.

Theorem 4.5. *Fix $\rho, \gamma, \beta > 0$. There exists some constant $C > 0$ such that the following holds. Suppose $f : \mathcal{X}^* \to \mathbb{R}^d$ is a function that has a (r,β)-stable point. Then, there exists a ρ-zCDP algorithm that takes an input a dataset $S \subset \mathcal{X}^n$ and w.p.$\geq 1-\beta$ returns a $((1+\gamma)r, {}^{\beta}/_{2k})$-stable point provided that $n \geq k \cdot m(r, {}^{\beta}/_{2k})$ for $k = \frac{C\left(\sqrt{d}+\sqrt{\log({}^1/_{\gamma\beta})}\right)\log({}^1/_{\gamma})}{\gamma^2\sqrt{\rho}}$. Furthermore, if finding $f(S')$ for any S' containing $m(r, {}^{\beta}/_{2k})$-many datapoint takes T time, then our algorithm runs in time $O(k\mathsf{T} + k \cdot \frac{\log^2({}^1/_{\gamma})}{\gamma^2})$.*

Proof. The proof simply partitions the n inputs points of S into k disjoint and random subsets $S'_1, S'_2, ..., S'_k$. W.p. $\geq 1-\beta/2$ it holds that $\|f(S'_i) - c\| \leq r$ for every subset S'_i, and then we apply our $(1+\gamma)$ approximation over this dataset of k many points (with a failure probability of $\beta/2$) and returns the resulting center-point. □

This results improves on Theorem 18 of [18] in both the runtime and the required number of subsamples, at the expense of requiring *all* subsamples to be close to the point p rather than just many of the points.

5 A Local-DP fPTAS for the MEB Problem

In this section we give the local-model version of our algorithm. At the core of its utility proof is a lemma analogous to Lemma 4.3, in which we prove that w.h.p. in each iteration t the distance between the true mean of the uncovered points and their LDP-approximation does not exceed ${}^{\gamma r}/_{16}$.

Claim 5.1. Algorithm 5 is a local-model ρ-zCDP.

Proof. The proof is very similar to the proof of Lemma 4.1 — where we apply basically the same accounting, noticing that each $x \in P$ is in charge of randomizing her own data, making this algorithm LDP. □

Algorithm 5 LDP-Margin based Minimum Enclosing Ball (LDP-MMEB)

```
  Input: a set of $n$ points $P \subseteq \mathbb{R}^d$, an approximation parameter $\gamma \in (0,1)$,
   an error parameter $\beta \in (0,1)$, privacy parameter $\rho$,
   a candidate radius $r$, and an initial center $\theta_0$ s.t. $\|\theta_0 - \theta_{opt}\| \le 10r_{opt}$.
Set $\theta^0 \leftarrow \theta_0, T \leftarrow \frac{64}{\gamma^2}\ln(\frac{100}{\gamma^2})$, $\sigma^2_{count} \leftarrow \frac{T+1}{\rho}$, and $\sigma^2_{sum} \leftarrow \frac{T\cdot(88r)^2}{\rho}$.
for $t = 0, 1, 2, \ldots, T-1$ do
    for each $(x \in P)$ do
        Sample $\Delta_{count} \sim \mathcal{N}(0, \sigma^2_{count})$.
        Sample $\Delta_{sum} \sim \mathcal{N}(0, \sigma^2_{sum} I_d)$.
        if $(x \notin B(\theta^t, r))$ then
            Send $Y^t_x = 1 + \Delta_{count}, Z^t_x = x - \theta^t + \Delta_{sum}$
        else Send $Y^t_x = \Delta_{count}, Z^t_x = \Delta_{sum}$
    Set $\tilde{n}^t_w = \sum_{x\in P} Y^t_x$ and $\tilde{v}^t_w = \frac{1}{\tilde{n}^t_w}\sum_{x\in P} Z^t_x$.
    if $(\tilde{n}^t_w < \frac{16\sqrt{n(T+1)}}{\gamma\sqrt{\rho}}\left(88\sqrt{d} + 110\sqrt{2\log({}^{4(T+1)}/_{\beta})}\right))$ then return $\theta^t$
    Update $\theta^{t+1} \leftarrow \theta^t + \frac{\gamma^2}{8}\tilde{v}^t_w$
for each $(x \in P)$ do
    Sample $\Delta_{count} \sim \mathcal{N}(0, \sigma^2_{count})$.
    if $(x \notin B(\theta^T, (1+\gamma)r))$ then
        Send $Y^T_x = 1 + \Delta_{count}$
    else Send $Y^T_x = \Delta_{count}$
Set $n^T_w \leftarrow \sum_x Y_x$
if $\left( n^T_w \le \sqrt{\frac{2n(T+1)\log({}^{4(T+1)}/_{\beta})}{\rho}} \right)$ then return $\theta^T$ else return $\perp$
```

Lemma 5.2. W.p. $\ge 1-\beta$, applying Algorithm 4 with $r \ge r_{opt}$ and an initial center θ_0 s.t. $\|\theta_0 - \theta_{opt}\| \le 10r_{opt}$ returns a point θ^t where $|P \setminus B(\theta^t, (1+\gamma)r)| \le \frac{\sqrt{16n(T+1)}}{\gamma\sqrt{\rho}}\left(88\sqrt{2d} + 110\sqrt{\log({}^{4(T+1)}/_{\beta})}\right) + \sqrt{\frac{2n(T+1)\log({}^{4(T+1)}/_{\beta})}{\rho}}$.

Proof. Analogously to the proof of Lemma 4.3, we use the similar definitions: in each iteration t we denote n^t_w as the true number of datapoints in P outside the ball $n^t_w = |\{x \in P:\ x \notin B(\theta^t, r)\}|$,[6] μ^t_w as their true mean $\mu^t_W = \frac{1}{n^t_w}\sum_{x\notin B(\theta^t,r)} x$, and v^t_w as the difference of the true mean and the current center $v^t_w = \mu^t_w - \theta^t = \frac{1}{n^t_w}\sum_{x\notin B(\theta^t,r)}(x - \theta^t)$. We thus define the events

$$\mathcal{E}_1 := \text{in all } T+1 \text{ iterations, } |\tilde{n}^t_w - n^t_w| \le \sqrt{\frac{2n(T+1)\log({}^{4(T+1)}/_{\beta})}{\rho}}$$

$$\mathcal{E}_2 := \text{in all } T \text{ iterations, } \|\sum_x Z^t_x - n^t_w v^t_w\| \le \frac{88r\sqrt{nT}}{\sqrt{\rho}}\left(\sqrt{d} + \sqrt{2\ln({}^{4T}/_{\beta})}\right)$$

Proving that both $\Pr[\overline{\mathcal{E}}_1] \le \beta/2$ and $\Pr[\overline{\mathcal{E}}_2] \le \beta/2$ is rather straight-forward. In each iteration t it holds that $\sum_x Y^t_x \sim \mathcal{N}(n^t_w, n\sigma^2_{count})$ as the sum on n independent Gaussians, and so we merely apply standard Gaussian concentration bounds together with the union bound over all $T+1$ iterations. Similarly, in teach iteration t it holds that $\sum_x Z^t_x \sim \mathcal{N}(n^t_w(\mu^t_x - \theta^t), n\sigma^2_{sum} I_d)$. So standard bounds on the concentration of the χ^2_d-distribution assert that the L_2-distance between the random draw from such a d-dimensional Gaussian and its mean is $> \sqrt{n\sigma^2_{sum}}(\sqrt{d} + \sqrt{2\ln({}^{4T}/_{\beta})}$ w.p. $< \frac{\beta}{2T}$, after which we apply the union-bound on all T iterations. We continue the rest of the proof conditioning on both $\mathcal{E}_1$ and $\mathcal{E}_2$ holding.

Denoting $\tilde{\mu}^t_w = \tilde{v}^t_w + \theta^t$, we can bound

$$\|\tilde{\mu}^t_w - \mu^t_w\| = \|\tilde{v}^t_w - v^t_w\| = \left\| \frac{\sum_x Z^t_x}{\tilde{n}^t_w} - \frac{\sum_{x\notin B(\theta^t,r)}(x-\theta^t)}{n^t_w} \right\| = \left\| \frac{\sum_x Z^t_x}{\tilde{n}^t_w} - \frac{n^t_w v^t_w}{n^t_w} \right\|$$

[6] Where technically, in the last steps of the algorithm, $n^T_w = |\{x \in P:\ x \notin B(\theta^T, (1+\gamma)r)\}|$.

$$\leq \left\| \frac{\sum_x Z_x^t - n_w^t v_w^t}{\tilde{n}_w^t} - \left(n_w^t v_w^t\right)\left(\frac{1}{\tilde{n}_w^t} - \frac{1}{n_w^t}\right)\right\| \leq \frac{\| \sum_x Z_x^t - n_w^t v_w^t \|}{\tilde{n}_w^t} + \|\mu_w^t - \theta^t\| \frac{|\tilde{n}_w^t - n_w^t|}{\tilde{n}_w^t}$$
$$\leq \frac{1}{\tilde{n}_w^t\sqrt{\rho}} \left(88r\sqrt{nT}\left(\sqrt{d} + \sqrt{2\ln(4T/\beta)}\right) + 22r\sqrt{2n(T+1)\log(4(T+1)/\beta)}\right) \leq \frac{\gamma r}{16}$$

as we only make an update step if $\tilde{n}_w^t \geq \frac{16\sqrt{n(T+1)}}{\gamma\sqrt{\rho}}\left(88\sqrt{d} + 110\sqrt{2\log(4(T+1)/\beta)}\right)$.

Again, we rely on Lemma 3.4 that asserts that under this conditions, if $r \geq r_{opt}$ and we make T updates, then we end up with θ^T which is within distance $\leq \gamma r_{opt}$ of θ_{opt}. We thus get that if we make T updates then $n_w^T = |\{x \in P : x \notin B(\theta^T, (1+\gamma)r)\}| = 0$ thus under $\mathcal{E}_1$ we have that $\tilde{n}_w^T \leq \sqrt{\frac{2n(T+1)\log(4(T+1)/\beta)}{\rho}}$. Alternatively, at some iteration t we halt in which case, under $\mathcal{E}_1$, we have that $n_w^t \leq \tilde{n}_w^t + \sqrt{\frac{2n(T+1)\log(4(T+1)/\beta)}{\rho}}$. □

Corollary 5.3. Algorithm 3 altered so it invokes $B = O(\log(1/\gamma))$ calls to Algorithm 5 (instead of Algorithm 4) is a LDP ρ-zCDP algorithm that returns a ball $B(\theta^*, r^*)$ such that $r^* \leq (1+3\gamma)r_{opt}$ and $|P \setminus B(\theta^*, r^*)| = O(\frac{\sqrt{n}\log(1/\gamma)}{\gamma^2\sqrt{\rho}}\left(\sqrt{d} + \sqrt{\log(1/\gamma\beta)}\right))$.

Proof. The proof follows from using the bound of Lemma 5.2 with $T = O(\gamma^{-2}\log(1/\gamma))$, and with a privacy budget of ρ/B and failure probability of β/B in each invocation of Algorithm 5. □

6 Experiments

In this section we give an experimental evaluation of our algorithm on three synthetic datasets and one real dataset. We emphasize that our experiment should be perceived merely as a proof-of-concept experiment aimed at the possibility of improving the algorithm's analysis, and not a thorough experimentation for a ready-to-deploy code. We briefly explain the experimental setup below.

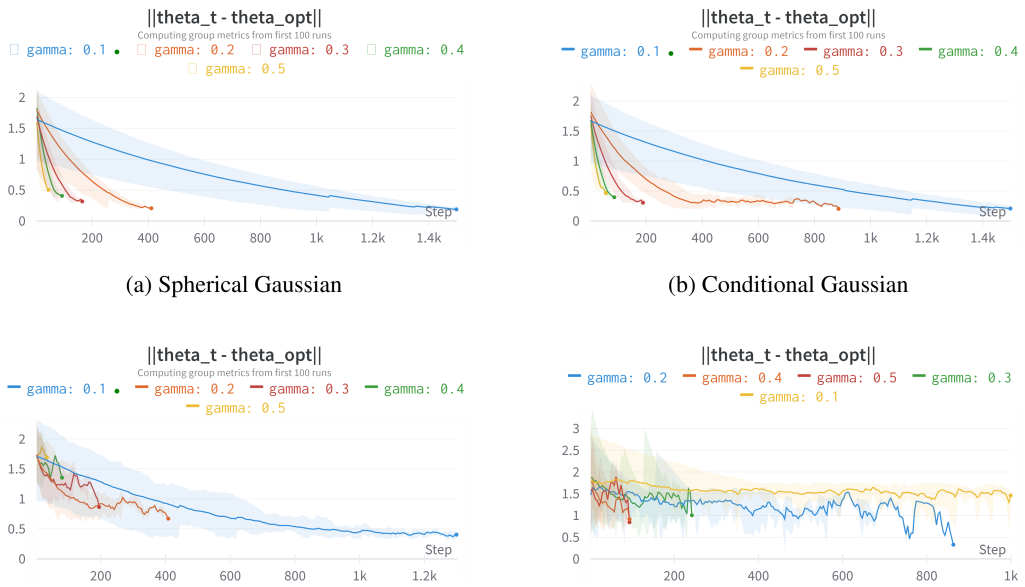
Goal. We set to investigate the performance of our algorithm, and seeing whether the performance is similar across different types of input and across a range of parameters. In addition, we wondered whether in practice our algorithm halts prior to concluding all $T = O(\gamma^{-2}\ln(1/\gamma))$ iterations.

Experiment details. We conducted experiments solely with Algorithm 4, feeding it the true r_{opt} of each given dataset as its r parameter. By default, we used the following set of parameters. Our domain in the synthetic experiments is $[-5,5]^{10}$ (namely, we work in the 10-dimensional space), and our starting point θ_0 is the origin. The default values of our privacy parameter is $\rho = 0.3$, of the approximation constant is 1.2 (namely $\gamma = 0.2$), and of the failure probability is $\beta = e^{0.9} \approx 0.00012$. We set the maximal number of repetitions T just as detailed in Algorithm 4, which depends on γ.

We varied two of the input parameters, ρ and γ, and also the data-type. We ran experiments with $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$ and with $\gamma \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$. Based on the values of ρ and γ we computed $n_0 = \frac{2\sqrt{T}(\sqrt{d}+\sqrt{\ln(T/\beta)}}{\gamma\sqrt{\rho}}$ which we used as our halting parameter. In all experiments involving a synthetic dataset, we set the input size n to be $n = 10n_0$.

We varied also the input type, using 3 synthetically generated datasets and one real-life dataset:

- Spherical Gaussian: we generated samples from a d-dimensional Gaussian $\mathcal{N}(v, I_d)$, where $v \in \mathbb{R}^d$ is a random shift vector. We discarded each point that did not fall in $[-5,5]^{10}$.
- Product Distribution: we generated samples from a d-dimensional Bernoulli distribution with support $\{-1,1\}^d$ with various probabilities for each dimension — where for each coordinate $i \in [10]$ we set $\Pr[x_i = 1] = 2^{-i}$. This creates a "skewed" distribution whose mean is quite far from its 1-center. In order for the 1-center not to coincide with $\theta_0 = \bar{0}$ we shifted this cube randomly in the grid.
- Conditional Gaussian: we repeated the experiment with the spherical Gaussian only this time we conditioned our random draws so that no coordinate lies in the $[0, 0.5]$-interval.



(a) Spherical Gaussian

(b) Conditional Gaussian

(c) Product Distribution

(d) Bar Crawl: Detecting Heavy Drinking

Figure 2: The distance of θ^t to θ_{opt} as a function of t – the iteration number, for $\rho = 0.3$ and $\gamma \in \{0.2, 0.3, 0.4, 0.5\}$. Each curve corresponds to a different γ value. In all experiments the number of iterations until convergence does increase as γ decreases.

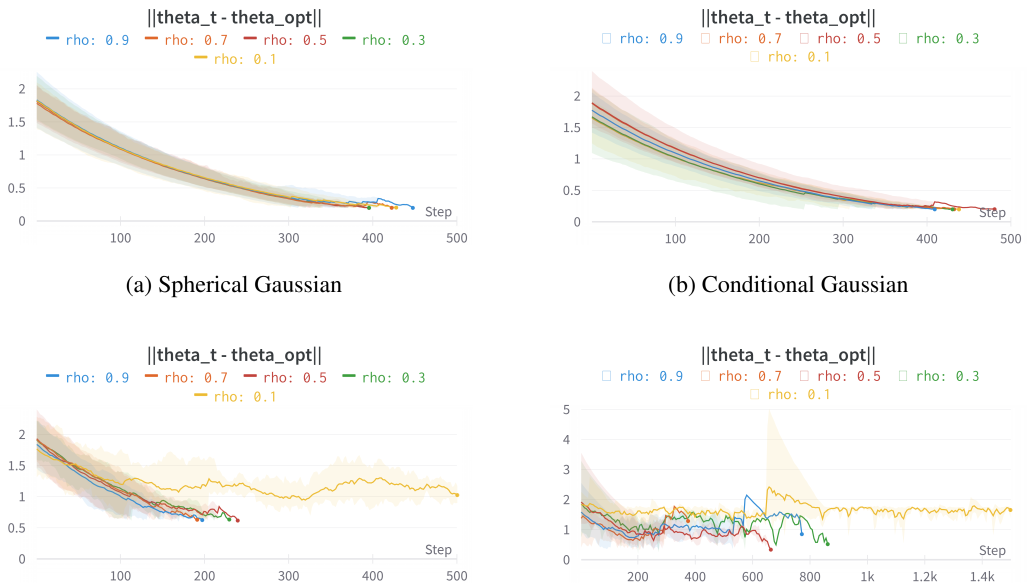
This skews the mean of the distribution to be < 0 in each coordinate, but leaves the 1-center unaltered. Again, we shifted the Gaussian to a random point $v \in [-5, 5]^d$.

- "Bar Crawl: Detecting Heavy Drinking": a dataset taken from the freely available UCI Machine Learning Repository [1] which collected accelerometer data from participants in a college bar crawl [23]. We truncated the data to only its 3 x-, y- and z-coordinates, and dropped any entry outside of $[-1, 1]^3$, and since it has two points $(-1, -1, -1)$ and $(1, 1, 1)$ then its 1-center is the origin (so we shifted the data randomly in the $[-5, 5]^d$ cube). This left us with $n = 12, 921, 593$ points. Note that the data is taken from a very few participants, so our algorithm gives an event-level privacy [15].

We ran our experiments in Python, on a (fairly standard) Intel Core i7 2.80 GHz with 16GB RAM and they run in time that ranged from 15 seconds (for $\gamma = 0.5$) to 2 hours (for $\gamma = 0.1$).

Results. The results are given in Figures 2, 3, where we plotted the distance of θ^t to θ_{opt} for each set of parameters across $t = 10$ repetitions. As evident, we converged to a good approximation of the MEB in almost all settings, with the exceptions of the cases of $\gamma = 0.1$ and $\rho = 0.1$ where a few repetitions failed to converge. We halt the experiment (i) if $\|\theta_t - \theta_{opt}\| \leq \gamma r_{opt}$, or (ii) if there are not enough wrong points, or (iii) if $t > 2500$ indicating that the run isn't converging — for $\gamma = 0.1$ or for $\rho = 0.1$ in "Bar Crawl" (see Figures 2d and 3d) where the size of the data n is simply not big enough. Indeed, the number of iterations until convergence does increase as γ decreases; but, rather surprisingly, varying ρ has a small effect on the halting time. This is somewhat expected as T has no dependency on ρ whereas its dependency on γ is proportional to γ^{-2}, but it is evident that as ρ increases our mean-estimation in each iteration becomes more accurate, so one would expect a faster convergence. Also unexpectedly, our results show that even for datasets whose mean and 1-center aren't close to one another (such as the Conditional Gaussian or Product-Distribution), the number of iterations until convergence remains roughly the same (see for example Figure 2 vs. 3).

Conclusions. Our experiments suggest that indeed our bound T is a worst-case bound, where in all experiments we concluded in about $7 - 50$ times faster than the bound of Algorithm 4. This suggests that perhaps one would be better off if instead of partitioning the privacy budget equally across all T iterations, they devise some sort of adaptive privacy budgeting. (E.g., using $^{3\rho}/_{4}$ budget on the



(a) Spherical Gaussian

(b) Conditional Gaussian

(c) Product Distribution

(d) Bar Crawl: Detecting Heavy Drinking

Figure 3: The distance of θ^t to θ_{opt} as a function of t – the iteration number, for $\gamma = 0.2$ and $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Each curve corresponds to a different ρ value. In all experiments varying ρ has a small effect on the halting time. W comment that in Figure3c even the experiment with $\rho = 0.1$ converged, in iteration $t = 1736$.

first $T/4$ iterations and then the remaining $^{\rho}/_{4}$ budget on the latter $^{3T}/_{4}$ iterations.) Such adaptive budgeting is simple when using zCDP, as it does not require "privacy odometers" [29].

7 Discussion and Open Problems

This work is the first to give a DP-fPATS for the MEB problem, in both the curator- and the local-model, and it leads to numerous open problems. The first is the question of improving the utility guarantee. Specifically, the number of points our algorithm may omit from P has a dependency of $\tilde{O}(\gamma^{-2})$ in the approximation factor, where this dependency follows from the fact that in each of our $T = \tilde{O}(\gamma^{-2})$ iterations we require an approximation of $O(\gamma r_{opt})$ to the true mean of the uncovered points. Thus finding either an iterative algorithm which makes $\ll T$ iterations or a variant of SVT that will allow the privacy budget to scale like $O(\log(T))$ will reduce this dependency to only $\tilde{O}(\gamma^{-1})$. Alternatively, it is intriguing whether there exists a lower-bound for any zCDP PTAS of the MEB problem proving a polynomial dependency on γ. (The best we were able to prove is via packing argument [20, 7] using a grid of $O((^1/_\gamma)^d)$ many points, leading to a $d\log(^1/_\gamma)$ bound.)

A different open problem lies on the the application of this DP-MEB approximation to the task of DP-clustering, and in particular — on improving on the works of [21, 30, 12] for "stable" k-median/means clustering. One can presumably combine our technique with the LSH-based approach used in [27] to cover a subset of points lying close together, however — it is unclear to us what is the effect of using only some of each cluster's "core" on the approximated MEB we return and on the k-means/median cost. More importantly, it does not seem that for the k-means problem our MEB approximation yields a better cost than the simple baseline of DP-averaging each cluster's core (after first finding a $O(1)$-MEB approximation, as discussed in the introduction). But it is possible that our work can be a building block in a first PTAS for the k-center problem in low-dimensions, a setting in which the k-center problem has a non-private PTAS [19].

References

[1] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[2] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *FOCS*, 2014.

[3] Mihai Batdoiu and Kenneth L. Clarkson. Smaller core-sets for balls. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 801–802, 2004.

[4] Asa Ben-Hur, David Horn, Hava T. Siegelmann, and Vladimir Vapnik. Support vector clustering. *J. Mach. Learn. Res.*, 2:125–137, mar 2002.

[5] Yaroslav Bulatov, Sachin R. Jambawalikar, Piyush Kumar, and Saurabh Sethia. Hand recognition using geometric classifiers. In *ICBA*, 2004.

[6] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *FOCS*, pages 634–649. IEEE Computer Society, 2015.

[7] Mark Bun and Thomas Steinke. Concentrated differential privacy: Simplifications, extensions, and lower bounds. In *TCC*, volume 9985 of *Lecture Notes in Computer Science*, pages 635–658, 2016.

[8] Mark Bun, Thomas Steinke, and Jonathan Ullman. Make up your mind: The price of online queries in differential privacy, 2016.

[9] Mihai Bundefineddoiu, Sariel Har-Peled, and Piotr Indyk. Approximate clustering via core-sets. In *Proceedings of the Thiry-Fourth Annual ACM Symposium on Theory of Computing*, STOC '02, page 250–257, New York, NY, USA, 2002. Association for Computing Machinery.

[10] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, Jun 1998.

[11] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine Learning*, 46(1):131–159, Jan 2002.

[12] Edith Cohen, Haim Kaplan, Yishay Mansour, Uri Stemmer, and Eliad Tsfadia. Differentially-private clustering of easy instances. In *ICML*, volume 139, pages 2049–2059. PMLR, 2021.

[13] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In *EUROCRYPT*, 2006.

[14] Cynthia Dwork, Frank Mcsherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, 2006.

[15] Cynthia Dwork, Moni Naor, Toniann Pitassi, and Guy N. Rothblum. Differential privacy under continual observation. In *STOC*, pages 715–724. ACM, 2010.

[16] D. Jack Elzinga and Donald W. Hearn. The minimum covering sphere problem. *Management Science*, 19(1):96–104, 2022/01/04/ 1972. Full publication date: Sep., 1972.

[17] David Eppstein and Jeff Erickson. Iterated nearest neighbors and finding minimal polytopes. *Discret. Comput. Geom.*, 11:321–350, 1994.

[18] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. Differentially private clustering: Tight approximation ratios. In *NeurIPS*, 2020.

[19] Sariel Har-peled. *Geometric Approximation Algorithms*. American Mathematical Society, 2011.

[20] Moritz Hardt and Kunal Talwar. On the geometry of differential privacy. In *STOC*, pages 705–714. ACM, 2010.

[21] Zhiyi Huang and Jinyan Liu. Optimal differentially private algorithms for k-means clustering. In *Proceedings of the 37th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, Houston, TX, USA, June 10-15, 2018*, pages 395–408. ACM, 2018.

[22] Philip M. Hubbard. Approximating polyhedra with spheres for time-critical collision detection. *ACM Trans. Graph.*, 15(3):179–210, jul 1996.

[23] Jackson A. Killian, Kevin M. Passino, Arnab Nandi, Danielle R. Madden, and John D. Clapp. Learning to detect heavy drinking episodes using smartphone accelerometer data. In *Proceedings of the 4th International Workshop on Knowledge Discovery in Healthcare Data co-located with the 28th International Joint Conference on Artificial Intelligence, KDH@IJCAI 2019, Macao, China, August 10th, 2019*, volume 2429 of *CEUR Workshop Proceedings*, pages 35–42. CEUR-WS.org, 2019. Dataset available freely on archive.ics.uci.edu/ml/datasets/Bar+Crawl%3A+Detecting+Heavy+Drinking.

[24] Piyush Kumar, Joseph S. B. Mitchell, E. Alper Yildirim, and E. Alper Yıldırım. Computing core-sets and approximate smallest enclosing hyperspheres in high dimensions. In *ALENEX), Lecture Notes Comput. Sci*, pages 45–55, 2003.

[25] Nimrod Megiddo. The weighted euclidean 1-center problem. *Math. Oper. Res.*, 8(4):498–504, 1983.

[26] K. Nissim, S. Raskhodnikova, and A. Smith. Smooth sensitivity and sampling in private data analysis. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing*, pages 75–84. ACM, 2007. Full version in: http://www.cse.psu.edu/~asmith/pubs/NRS07.

[27] Kobbi Nissim and Uri Stemmer. Clustering algorithms for the centralized and local models. *ArXiv*, abs/1707.04766, 2018.

[28] Kobbi Nissim, Uri Stemmer, and Salil Vadhan. Locating a small cluster privately. *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, Jun 2016.

[29] Ryan M. Rogers, Salil P. Vadhan, Aaron Roth, and Jonathan R. Ullman. Privacy odometers and filters: Pay-as-you-go composition. In *NIPS*, pages 1921–1929, 2016.

[30] Moshe Shechner, Or Sheffet, and Uri Stemmer. Private k-means clustering with stability assumptions. In Silvia Chiappa and Roberto Calandra, editors, *AISTATS*, volume 108, pages 2518–2528. PMLR, 2020.

[31] Salil Vadhan. *The Complexity of Differential Privacy*, pages 347–450. Springer, Yehuda Lindell, ed., 2017.

A Finding an Initial Good Center

In this section we give, for completeness, the ρ-zCDP version of the algorithms for approximating P's optimal radius up to a constant factor and finding some θ_0 which is sufficiently close to the center of P's MEB. The algorithm itself is ridiculously simple, and have appeared before implicitly. We bring it here for two reasons: (a) completeness and (b) in its LDP-version, this algorithm's utility depends solely on $\sqrt{n}$. Thus, combining this algorithm with the Algorithm 5 of Section 5, we obtain a LDP-fPTAS for the MEB problem who's utility depends on $\sqrt{n}$ rather than the $n^{0.67}$-bound of [27] (at the expense of worse dependency on other parameters). This gives a clear improvement on previous algorithms for approximating the MEB problem when $n \to \infty$. Our algorithm requires a starting point θ_0 which is $R_{\max}$ away from all points in P (namely, $P \subset B(\theta_0, R_{\max})$), and a lower bound $r_{\min}$ on r_{opt}; and its overall utility bounds depends on $\log(R_{\max}/r_{\min})$. In a standard setting, where $P \subset [-B, B]^d$ and where all points lie on some grid $\mathcal{G}^d$ whose step-size is τ, we can set θ_0 as the origin and set $R_{\max} = B\sqrt{d}$ and $r_{\min} = \tau/2$, resulting in $O(\log({}^{Bd}/_{\tau}))$-dependency. In the specific case where $r_{opt} = 0$ and all datapoints in P lie on the exact same grid point we can just return the closest grid point to the resulting θ once it get to a radius of $r = r_{min} = \tau/2$.

Algorithm 6 Noisy Average and Radius (GoodCenter)

```
Input: a set of $n$ points $P$ and parameters $\theta_0$, $R_{\max}$ and $r_{\min}$, such that $P \subset B(\theta_0, R_{\max})$ and $r_{opt} \geq r_{\min}$. Failure parameter $\beta \in (0,1)$, privacy parameter $\rho$.
Set $T \leftarrow \lceil \log_2(R_{\max}/r_{\min}) \rceil + 1$, $X \leftarrow \sqrt{\frac{2T\ln({}^{4T}/_{\beta})}{\rho}}$
Set $\sigma^2_{count} \leftarrow \frac{T}{\rho}$, $\sigma^2_{sum} \leftarrow \frac{T}{\rho}$.
Init $P^0 \leftarrow P$, $\theta^0 \leftarrow \theta_0$, $n_{cur} \leftarrow n$ and $r_{cur} \leftarrow R_{\max}$.
for ($t = 0, 1, 2, ..., T-1$) do
    $P^t \leftarrow P^t \cap B(\theta^t, r_{cur})$.
    $\Delta_{sum} \sim \mathcal{N}(0, 4r^2_{cur}\sigma^2_{sum}I_d)$
    $\tilde{\mu}^t \leftarrow (\sum_{x \in P^t} x + \Delta_{sum})/n_{cur}$
    $\Delta_{count} \leftarrow \mathcal{N}(0, \sigma^2_{count})$
    if ($|P^t \setminus B(\tilde{\mu}^t, \frac{1}{2}r_{cur})| + \Delta_{count} \geq X$) then return $B(\theta^t, r_{cur})$
    Update: $r_{cur} \leftarrow \frac{1}{2}r_{cur}$, $n_{cur} \leftarrow n_{cur} - 2X$, $\theta^{t+1} \leftarrow \tilde{\mu}^t$.
return $B(\theta^T, r_{cur})$
```

Theorem A.1. *Algorithm 6 is ρ-zCDP.*

Proof. The proof follows immediately from the fact that the L_2-global sensitivity of a count query is 1, and that the L_2-global sensitivity of a sum of datapoints in a ball of radius r_{cur} is at most $2r_{cur}$. The rest of the proof relies on the composition of $2T$ queries, each answered with a "budget" of $\frac{\rho}{2T}$-zCDP. □

Theorem A.2. *W.p. $\geq 1-\beta$, given a set of points P of size n where $n \geq \max\{16T\sqrt{\frac{2T\ln(4T/\beta)}{\rho}}, 16\sqrt{\frac{T}{\rho}}(\sqrt{d}+\sqrt{2\ln(4T/\beta)})\}$, Algorithm 6 returns a ball $B(\theta^*, r^*)$ where (i) the set $P' = P \cap B(\theta^*, r^*)$ contains at least $n - \sqrt{\frac{8T^3\ln(4T/\beta)}{\rho}}$, and (ii) denoting $B(\theta(P'), r_{opt}(P'))$ as the MEB of P', we have that $r^* \leq 6r_{opt}$.*

Proof. Let $\mathcal{E}$ be the event where for any of the $\leq T$ draws of the Δ_{sum} and Δ_{count} it holds that

$$|\Delta_{count}| \leq \sqrt{\frac{2T\ln(4T/\beta)}{\rho}} \qquad \text{and} \qquad \|\Delta_{sum}\| \leq 2r_{cur}\sqrt{\frac{T}{\rho}}(\sqrt{d}+\sqrt{2\ln(4T/\beta)})$$

where again, standard union bound and Gaussian / χ^2-distribution concentration bounds give that $\Pr[\overline{\mathcal{E}}] \leq \beta$. So we continue the proof under the assumption that $\mathcal{E}$ holds.

In this case, in any iteration it must hold that $|P \setminus B(\mu^t, \frac{1}{2}r_{cur})| \leq 2X = \sqrt{\frac{8T\ln(4T/\beta)}{\rho}}$. It follows that all in all we remove in the process of Algorithm 6 at most $2XT$ points, and since $n \geq 16XT$ we have that in any iteration t it always holds that $n \geq |P^t| \geq n - 2Xt = n_{cur} \geq \frac{7n}{8} \geq 14XT$. Denoting in any iteration t the true mean of the points (remaining) in P^t as $\mu_t = \frac{1}{|P^t|}\sum_{x\in P^t} x$, and the center of the MED of P^t as θ_t, it follows that

$$\begin{aligned}
\|\tilde{\mu}^t - \mu^t\| &= \|\tilde{\mu}^t - \theta_t - (\mu^t - \theta_t)\| = \left\| \frac{\Delta_{sum} + \sum_{x\in P^t}(x-\theta_t)}{n_{cur}} - \frac{\sum_{x\in P^t}(x-\theta_t)}{|P^t|} \right\| \\
&\leq \left\| \frac{\Delta_{sum}}{n_{cur}} \right\| + \left\| \frac{\left(\sum_{x\in P^t}(x-\theta_t)\right)(|P^t| - n_{cur})}{|P^t| n_{cur}} \right\| \leq \frac{8\|\Delta_{sum}\|}{7n} + \|\mu^t - \theta_t\| \frac{2XT}{n_{cur}} \\
&\leq \frac{8\cdot 2r_{cur}\sqrt{\frac{T}{\rho}}(\sqrt{d}+\sqrt{2\ln(4T/\beta)})}{7n} + \frac{r_{opt}(P^t)}{7} \leq \frac{r_{cur} + r_{opt}(P^t)}{7}
\end{aligned}$$

Since we assume $n \geq 16\sqrt{\frac{T}{\rho}}(\sqrt{d}+\sqrt{2\ln(4T/\beta)})$. Moreover, since $\|\mu^t - \theta_t\| \leq r_{opt}(P^t)$ it follows that $\|\tilde{\mu}^t - \theta_t\| \leq \frac{r_{cur}+8r_{opt}(P^t)}{7}$. Now, as long as $r_{cur} \geq 6r_{opt}(P^t)$ we have that

$$\frac{r_{cur}}{2} \geq \frac{r_{cur}}{7} + \frac{5r_{cur}}{14} \geq \frac{r_{cur}}{7} + \frac{30r_{opt}(P^t)}{14} \geq r_{opt}(P^t) + \frac{r_{cur} + 8r_{opt}(P^t)}{7} \geq r_{opt}(P^t) + \|\tilde{\mu}^t - \theta_t\|$$

thus $B(\theta_t, r_{opt}(P^t)) \subset B(\tilde{\mu}^t, \frac{r_{cur}}{2})$ which implies that $|P^t \setminus B(\mu^t, \frac{1}{2}r_{cur})| = 0$, and so under $\mathcal{E}$ we continue to the next iteration.

And so, when we halt it must hold that r_{cur} (which is the r^* we return) must satisfy that $r_{cur} < 6r_{opt}(P^t)$. □

Corollary A.3. Algorithm 6 is a ρ-zCDP algorithm that, given n points on a grid $\mathcal{G} \subset [-B, B]^d$ of side-step τ where $n = \Omega(\sqrt{\frac{\log(Bd/\tau)}{\rho}}(\sqrt{d} + \sqrt{\log(Bd/\tau\beta)}))$ returns w.p. $\geq 1-\beta$ a ball $B(\theta^*, r^*)$ where for $P' = P \setminus B(\theta^*, r^*)$ it holds that both $n - |P'| = O(\frac{\log(Bd/\tau)}{\sqrt{\rho}}\sqrt{\log(Bd/\tau\beta)}))$ and that w.r.t to $B(\theta_{opt}, r_{opt})$ which is the true MEB of P' we have that $\|\theta^* - \theta_{opt}\| \leq 6r_{opt}(P')$.

A.1 A Local-DP Version of Finding an Initial Good Center

Algorithm 7 LDP Noisy Average and Radius (LDP-GoodCenter)

```
**Input:** a set of $n$ points $P$ and some parameter $R_{\max}, \theta_0$ and $r_{\min}$, such that $P \subset B(\theta_0, R_{\max})$ and $r_{opt} \geq r_{\min}$. Failure parameter $\beta \in (0,1)$, privacy parameter $\rho$.
Set $T \leftarrow \lceil \log_2(R_{\max}/r_{\min}) \rceil + 1$, $X \leftarrow \sqrt{\frac{2nT\ln(4T/\beta)}{\rho}}$
$\sigma^2_{count} \leftarrow \frac{T}{\rho}$, $\sigma^2_{sum} \leftarrow \frac{T}{\rho}$.
Init $\theta^0 \leftarrow \theta_0$, and $r_{cur} \leftarrow R_{\max}$.
for $(t = 0, 1, 2, ..., T-1)$ do
    Denote $\Pi^t$ as the projection onto $B(\theta^t, r_{cur})$.
    for each $x \in P$ do
        Send $Y_x \sim \mathcal{N}(\Pi^t(x), 4r^2_{cur}\sigma^2_{sum} I_d)$
    $\tilde{\mu}^t \leftarrow \frac{1}{n}\sum_x Y_x$
    for each $x \in P$ do
        if $(x \notin B(\tilde{\mu}^t, \frac{1}{2}r_{cur}))$ then
            Send $Z_x \sim \mathcal{N}(1, \sigma^2_{count})$
        else Send $Z_x \sim \mathcal{N}(0, \sigma^2_{count})$
    if $(\sum_x Z_x \geq X)$ then return $B(\theta^t, r_{cur})$
    Update: $r_{cur} \leftarrow \frac{1}{2}r_{cur}$, $\theta^{t+1} \leftarrow \tilde{\mu}^t$.
return $B(\theta^T, r_{cur})$
```

Theorem A.4. *Algorithm 7 is a LDP algorithm in which each user maintains ρ-zCDP. Forthermore, w.p. $\geq 1-\beta$, given a set of point P of size n where $n \geq \max\{16T\sqrt{\frac{2nT\ln(4T/\beta)}{\rho}}, 16\sqrt{\frac{nT}{\rho}}(\sqrt{d} + \sqrt{2\ln(4T/\beta)})\}$, Algorithm 7 returns a ball $B(\theta^*, r^*)$ where the set $P' = \{\Pi_{B(\theta^*,r^*)}(x) : x \in P\}$ contains no more than $2T\sqrt{\frac{2T\ln(4T/\beta)}{\rho}}$ points for which $x \neq \Pi_{B(\theta^*,r^*)}(x)$; and denoting $B(\theta(P'), r_{opt}(P'))$ as the MEB of P', it holds that $\|\theta^* - \theta(P')\| \leq 8r*$.*

The proof of Theorem A.4 is completely analogous to the proof of Theorems A.1 and A.2 using the fact that in each iteration t of the algorithm

$$\sum_x Y_x \sim \mathcal{N}\left(\sum_x \Pi^t(x),\ \ 4nr^2_{cur}\sigma^2_{sum}I_d\right)$$
$$\sum_x Z_x \sim \mathcal{N}\left(|\{x \in P :\ x \notin B(\tilde{\mu}^t, r_{cur}/2)\}|,\ \ n\sigma^2_{count}\right)$$

Corollary A.5. Algorithm 7 is a ρ-zCDP algorithm in the local-model that, given n points on a grid $\mathcal{G} \subset [-B,B]^d$ of side-step τ where $n = \Omega(\frac{\log(Bd/\tau)}{\rho}(\sqrt{d} + \sqrt{\log(Bd/\tau\beta)})^2)$ returns w.p. $\geq 1-\beta$ a ball $B(\theta^*, r^*)$ where for the set $P' = \{\Pi_{B(\theta^*,r^*)}(x) :\ x \in P\}$ it holds that at most $O(\frac{\sqrt{n}\cdot\log(Bd/\tau)}{\sqrt{\rho}}\sqrt{\log(Bd/\tau\beta)})$ points are shifted in the projection (and the rest remain as they are in P) and that w.r.t to $B(\theta_{opt}, r_{opt})$ which is the true MEB of P' we have that $\|\theta^* - \theta_{opt}\| \leq 6r^*$.

Note that comparing Corollary A.5 with the approximation of [27], we have that they may omit $O(n^{0.67}\log(n/\tau))$-many points whereas we may omit only $\sqrt{n}\log^{3/2}(d/\tau)$ points. But, of course, they deal with a bounding ball for t points out of giving n, whereas we deal with the MEB problem.