How to Find a Point in the Convex Hull Privately

Haim Kaplan
School of Computer Science, Tel Aviv University, Israel
Google, Tel Aviv, Israel
haimk@tau.ac.il

Micha Sharir
School of Computer Science, Tel Aviv University, Israel
michas@tau.ac.il

Uri Stemmer
Department of Computer Science, Ben-Gurion University, Beer Sheva, Israel
Google, Tel Aviv, Israel
u@uri.co.il

Abstract

We study the question of how to compute a point in the convex hull of an input set S of n points in $\mathbb{R}^d$ in a differentially private manner. This question, which is trivial without privacy requirements, turns out to be quite deep when imposing differential privacy. In particular, it is known that the input points must reside on a fixed *finite* subset $G \subseteq \mathbb{R}^d$, and furthermore, the size of S must grow with the size of G. Previous works [1, 2, 3, 4, 5, 11] focused on understanding how n needs to grow with $|G|$, and showed that $n = O\left(d^{2.5} \cdot 8^{\log^* |G|}\right)$ suffices (so n does not have to grow significantly with $|G|$). However, the available constructions exhibit running time at least $|G|^{d^2}$, where typically $|G| = X^d$ for some (large) discretization parameter X, so the running time is in fact $\Omega(X^{d^3})$.

In this paper we give a differentially private algorithm that runs in $O(n^d)$ time, assuming that $n = \Omega(d^4 \log X)$. To get this result we study and exploit some structural properties of the Tukey levels (the regions $D_{\geq k}$ consisting of points whose Tukey depth is at least k, for $k = 0, 1, \dots$). In particular, we derive lower bounds on their volumes for point sets S in general position, and develop a rather subtle mechanism for handling point sets S in degenerate position (where the deep Tukey regions have zero volume). A naive approach to the construction of the Tukey regions requires $n^{O(d^2)}$ time. To reduce the cost to $O(n^d)$, we use an approximation scheme for estimating the volumes of the Tukey regions (within their affine spans in case of degeneracy), and for sampling a point from such a region, a scheme that is based on the volume estimation framework of Lovász and Vempala [14] and of Cousins and Vempala [7]. Making this framework differentially private raises a set of technical challenges that we address.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Theory of computation → Randomness, geometry and discrete structures; Security and privacy → Formal methods and theory of security

Keywords and phrases Differential privacy, Tukey depth, Convex hull

Digital Object Identifier 10.4230/LIPIcs.SoCG.2020.52

Related Version A full version of the paper is available at `http://arxiv.org/abs/2003.13192`.

Funding *Haim Kaplan*: Partially supported by ISF grant 1595/19 and grant 1367/2016 from the German-Israeli Science Foundation (GIF)
Micha Sharir: Partially supported by ISF Grant 260/18, by grant 1367/2016 from the German-Israeli Science Foundation (GIF), and by Blavatnik Research Fund in Computer Science at Tel Aviv University.
Uri Stemmer: Partially supported by ISF grant 1871/19.

Acknowledgements We thank Santosh Vempala for many helpful discussions.

1 Introduction

We often would like to analyze data while protecting the privacy of the individuals that contributed to it. At first glance, one might hope to ensure privacy by simply deleting all names and ID numbers from the data. However, such anonymization schemes are proven time and again to violate privacy. This gave rise of a theoretically-rigorous line of work that has placed private data analysis on firm foundations, centered around a mathematical definition for privacy known as *differential privacy* [8].

Consider a database S containing personal information of individuals. Informally, an algorithm operating on such a database is said to preserve differential privacy if its outcome distribution is (almost) insensitive to any arbitrary change to the data of one individual in the database. Intuitively, this means that an observer looking at the outcome of the algorithm (almost) cannot distinguish between whether Alice's information is x or y (or whether Alice's information is present in the database at all) because in any case it would have (almost) no effect on the outcome distribution of the algorithm.

▶ **Definition 1.1** (Dwork et al. [8]). *Two databases (multisets) S and S' are called* neighboring *if they differ in a single entry. That is, $S = S_0 \cup \{x\}$ and $S' = S_0 \cup \{y\}$ for some items x and y. A randomized algorithm A is (ε, δ)-*differentially private *if for every two neighboring databases S, S' and for any event T we have*

$$\boldsymbol{Pr}[A(S) \in T] \leq e^{\varepsilon} \cdot \boldsymbol{Pr}[A(S') \in T] + \delta.$$

When $\delta = 0$ this notion is referred to as pure *differential privacy, and when $\delta > 0$ it is referred to as* approximate *differential privacy.*

▶ Remark 1.2. Typically, ε is set to be a small constant, say $\varepsilon = 0.1$, and δ is set to be a small function of the database size $|S|$ (much smaller than $1/|S|$). Note that to satisfy the definition (in any meaningful way) algorithm A must be randomized.

Differential privacy is increasingly accepted as a standard for rigorous treatment of privacy. However, even though the field has witnessed an explosion of research in the recent years, much remains unknown and answers to fundamental questions are still missing. In this work we study one such fundamental question, already studied in [1, 2, 3, 4, 5, 11]: Given a database containing points in $\mathbb{R}^d$ (where every point is assumed to be the information of one individual), how can we *privately* identify a point in the *convex hull* of the input points? This question, which is trivial without privacy requirements, turns out to be quite deep when imposing differential privacy. In particular, Bun et al. [5] showed that in order to be able to solve it, we must assume that the input points reside on a fixed *finite* subset $G \subseteq \mathbb{R}^d$, and furthermore, the number of input points must grow with the size of G.

The Private Interior Point (PIP) Problem.
Let $\beta, \varepsilon, \delta, X$ be positive parameters where $\beta, \varepsilon, \delta$ are small and X is a large integer. Let $G \subseteq [0,1]^d$ be a finite uniform grid with side steps $1/X$ (so $|G| = (X+1)^d$). Design an algorithm A such that for some $n \in \mathbb{N}$ (as small as possible as a function of $\beta, \varepsilon, \delta, X$) we have

1. **Utility:** For every database S containing at least n points from G it holds that $A(S)$ returns a point in the convex hull of S with probability at least $1 - \beta$. (The outcome of A does not have to be in G.)
2. **Privacy:** For every pair of neighboring databases S, S', each containing at least n points from G, and for any event T, we have $\mathbf{Pr}[A(S) \in T] \leq e^{\varepsilon} \cdot \mathbf{Pr}[A(S') \in T] + \delta$.

The parameter n is referred to as *the sample complexity* of the algorithm. It is the smallest number of points on which we are guaranteed to succeed (not to be confused with the actual size of the input). The PIP problem is very natural on its own. Furthermore, as was observed in [2], an algorithm for solving the PIP problem can be used as a building block in other applications with differential privacy, such as learning halfspaces and linear regression. Previous works [1, 2, 3, 4, 5, 11] have focused on the task of minimizing the sample complexity n while ignoring the runtime of the algorithm. In this work we seek an efficient algorithm for the PIP problem, that still keeps the sample complexity n "reasonably small" (where "reasonably small" will be made precise after we introduce some additional notation).

1.1 Previous Work

Several papers studied the PIP problem for $d = 1$. In particular, three different constructions with sample complexity $2^{O(\log^* |G|)}$ were presented in [3, 4, 5] (for $d = 1$). Recently, Kaplan et al. [11] presented a new construction with sample complexity $O((\log^* |G|)^{1.5})$ (again, for $d = 1$). Bun et al. [5] gave a lower bound showing that every differentially private algorithm for this task must have sample complexity $\Omega(\log^* |G|)$. Beimel et al. [2] incorporated a dependency in d to this lower bound, and showed that every differentially private algorithm for the PIP problem must use at least $n = \Omega(d + \log^* |G|)$ input points.

For the case of *pure* differential privacy (i.e., $\delta = 0$), a lower bound of $n = \Omega(\log X)$ on the sample complexity follows from the results of Beimel et al. [1]. This lower bound is tight, as an algorithm with sample complexity $n = O(\log X)$ (for $d = 1$) can be obtained using a generic tool in the literature of differential privacy, called the *exponential mechanism* [16]. We sketch this application of the exponential mechanism here. (For a precise presentation of the exponential mechanism see Section 2.1.) Let $G = \{0, \frac{1}{X}, \frac{2}{X}, \dots, 1\}$ be our (1-dimensional) grid within the interval $[0, 1]$, and let S be a multiset containing n points from G. The algorithm is as follows.

1. For every $y \in G$ define the *score* $q_S(y) = \min\{|\{x \in S \mid x \geq y\}|, \ |\{x \in S \mid x \leq y\}|\}$.
2. Output $y \in G$ with probability proportional to $e^{\varepsilon \cdot q_S(y)}$.

Intuitively, this algorithm satisfies differential privacy because changing one element of S changes the score $q_S(y)$ by at most ± 1, and thus changes the probabilities with which we sample elements by roughly an e^{ε} factor. As for the utility analysis, observe that $\exists y \in G$ with $q_S(y) \geq \frac{n}{2}$, and the probability of picking this point is (at least) proportional to $e^{\varepsilon n/2}$. As this probability increases exponentially with n, by setting n to be big enough we can ensure that points y' outside of the convex hull (those with $q_S(y') = 0$) get picked with very low probability.

Beimel et al. [2] observed that this algorithm extends to higher dimensions by replacing $q_S(y)$ with the Tukey depth $\mathtt{td}_S(y)$ of the point y with respect to the input set S (the *Tukey depth* of a point y is the minimal number of points that need to be removed from S to ensure that y is *not* in the convex hull of the remaining input points). However, even though there must exist a point $y \in \mathbb{R}^d$ with high Tukey depth (at least $n/(d+1)$; see [15]), the finite grid $G \subseteq \mathbb{R}^d$ might fail to contain such a point. Hence, Beimel et al. [2] first *refined* the grid G into a grid G' that contains a point with high Tukey depth, and then randomly picked a point y from G' with probability proportional to $e^{\varepsilon \cdot \mathtt{td}_S(y)}$. To compute the probabilities with which grid points are sampled, the algorithm in [2] explicitly computes the Tukey depth of every point in G', which, because of the way in which G' is defined, results in running time of at least $\Omega(|G|^{d^2}) = \Omega(X^{d^3})$ and sample complexity $n = O(d^3 \log |G|) = O(d^4 \log X)$. Beimel et al. then presented an improvement of this algorithm with reduced sample complexity of $n = O(d^{2.5} \cdot 8^{\log^* |G|})$, but the running time remained $\Omega(|G|^{d^2}) = \Omega(X^{d^3})$.

1.2 Our Construction

We give an approximate differentially private algorithm for the private interior point problem that runs in $O(n^d)$ time,[1] and succeeds with high probability when the size of its input is $\Omega(\frac{d^4}{\varepsilon}\log\frac{X}{\delta})$. Our algorithm is obtained by carefully implementing the exponential mechanism and reducing its running time from $\Omega(|G|^{d^2}) = \Omega(X^{d^3})$ to $O(n^d)$. We now give an informal overview of this result.

To avoid the need to extend the grid and to calculate the Tukey depth of each point in the extended grid, we sample our output directly from $[0,1]^d$. To compute the probabilities with which we sample a point from $[0,1]^d$ we compute, for each k in an appropriate range, the *volume* of the Tukey region of depth k, which we denote as D_k. (That is, D_k is the region in $[0,1]^d$ containing all points with Tukey depth exactly k.) We then sample a value $k \in [n]$ with probability proportional to $\mathrm{Vol}(D_k)\cdot e^{\varepsilon k}$, and then sample a random point uniformly from D_k.

Observe that this strategy picks a point with Tukey depth k with probability proportional to $\mathrm{Vol}(D_k)\cdot e^{\varepsilon k}$. Hence, if for a "large enough" value of k (say $k \geq \frac{n}{cd}$ for a suitable absolute constant $c > 1$) we have that $\mathrm{Vol}(D_k)$ is "large enough", then a point with Tukey depth k is picked with high probability. However, if $\mathrm{Vol}(D_k) = 0$ (or too small) then a point with Tukey depth k is picked with probability zero (or with too small a probability). Therefore, to apply this strategy, we derive a lower bound on the volume of every non-degenerate Tukey region, showing that if the volume is non-zero, then it is at least $\Omega\left(1/X^{d^3}\right)$.

There are two issues here. The first issue is that the best bound we know on the complexity of a Tukey region is $O(n^{(d-1)\lfloor d/2\rfloor})$, so we cannot compute these regions explicitly (in the worst-case) in time $O(n^d)$ (which is our target runtime complexity). We avoid the need to compute the Tukey regions explicitly by using an approximation scheme for estimating the volume of each region and for sampling a point from such a region, a scheme that is based on the volume estimation framework of Lovász and Vempala [14] and of Cousins and Vempala [7]. The second issue is that it might be the case that all Tukey regions for large values of k are degenerate, i.e., have volume 0, in which case this strategy might fail to identify a point in the convex hull of the input points.

Handling degeneracies. We show that if the Tukey regions of high depth are degenerate, then many of the input points must lie in a lower-dimensional affine subspace. This can be used to handle degenerate inputs S as follows. We first (privately) check whether there exists an affine proper subspace that contains many points of S. If we find such a subspace f, we recursively continue the procedure within f, with respect to $S \cap f$. Otherwise, if no such subspace exists, then it must be the case that the Tukey regions of high depth are full-dimensional (with respect to the subspace into which we have recursed so far), so we can apply our algorithm for the non-degenerate case and obtain a point that lies, with high probability, in the convex hull of the surviving subset of S, and thus of the full set S.

We remark that it is easy to construct algorithms with running time polynomial in the input size n, when n grows exponentially in d. (In that case one can solve the problem using a reduction to the 1-dimensional case.) In this work we aim to reduce the running time while keeping the sample complexity n at most polynomial in d and in $\log|G|$.

[1] When we use O-notation for time complexity we hide logarithmic factors in X, $1/\varepsilon$, $1/\delta$, and polynomial factors in d. We assume operations on real numbers in $O(1)$ time (the so-called real RAM model).

2 Preliminaries

We assume that our input set S consists of n points that lie in the intersection of a grid G with the cube $[0,1]^d$ in $\mathbb{R}^d$. We assume that G is of side length $1/X$ for a given accuracy integer parameter X, so it partitions the cube into X^d cells.

2.1 The exponential mechanism

Let G^* denote the set of all finite databases over a grid G, and let F be a finite set. Given a database $S \in G^*$, a *quality* (or *scoring*) function $q : G^* \times F \to \mathbb{N}$ assigns a number $q(S,f)$ to each element $(S,f) \in G^* \times F$, identified as the "quality" of f with respect to S. We say that the function q has *sensitivity* Δ if for all neighboring databases S and S' and for all $f \in F$ we have $|q(S,f) - q(S',f)| \le \Delta$.

The *exponential mechanism* of McSherry and Talwar [16] privately identifies an element $f \in F$ with large quality $q(S,f)$. Specifically, it chooses an element $f \in F$ randomly, with probability proportional to $\exp(\varepsilon \cdot q(S,f)/(2\Delta))$. The privacy and utility of the mechanism are:

▶ **Theorem 2.1** (McSherry and Talwar [16]). *The exponential mechanism is $(\varepsilon, 0)$-differentially private. Let q be a quality function with sensitivity Δ. Fix a database $S \in G^*$ and let $\mathrm{OPT} = \max_{f\in F}\{q(S,f)\}$. For any $\beta \in (0,1)$, with probability at least $(1-\beta)$, the exponential mechanism outputs a solution f with quality $q(S,f) \ge \mathrm{OPT} - \frac{2\Delta}{\varepsilon}\ln\frac{|F|}{\beta}$.*

2.2 Tukey depth

The *Tukey depth* [18] $\mathsf{td}_S(q)$ of a point q with respect to S is the minimum number of points of S we need to remove to make q lie outside the convex hull of the remaining subset. Equivalently, $\mathsf{td}_S(q)$ is the smallest number of points of S that lie in a closed halfspace containing q. We will write $\mathsf{td}(q)$ for $\mathsf{td}_S(q)$ when the set S is clear from the context. It easily follows from Helly's theorem that there is always a point of Tukey depth at least $n/(d+1)$ (see, e.g., [15]). We denote the largest Tukey depth of a point by $\mathsf{td}_{\max}(S)$ (the maximum Tukey depth is always at most $n/2$).

We define the regions $D_{\ge k}(S) = \left\{q \in [0,1]^d \mid \mathsf{td}_S(q \ge k\right\}$ and $D_k(S) = D_{\ge k}(S) \setminus D_{\ge k+1}(S)$ for $k = 0, \ldots, \mathsf{td}_{\max}(S)$. Note that $D_{\ge 1}$ is the convex hull of S and that $D_{\ge 0} = [0,1]^d$. It is easy to show that $D_{\ge k}$ is convex; $D_{\ge k}$ is in fact the intersection of all (closed) halfspaces containing at least $n-k+1$ points of S; see [17]. It is easy to show that all this is true also when S is degenerate. See Figure 1 for an illustration. The following lemma is easy to establish.

▶ **Lemma 2.2.** *If $D_{\ge k}$ is of dimension d (we refer to such a region as non-degenerate) then $C_k = \partial D_{\ge k}(S)$ is a convex polytope, each of whose facets is contained in a simplex σ spanned by d points of S, such that one of the open halfspaces bounded by the hyperplane supporting σ contains exactly $k-1$ points of S.*

3 The case of general position

As already said, we apply the exponential mechanism for privately identifying a point in $[0,1]^d$ with (hopefully) large Tukey depth with respect to the input set S. This satisfies (pure) differential privacy since the sensitivity of the Tukey depth is 1. In this section we show that when the input points are in general position, then this application of the exponential mechanism succeeds (with high probability) in identifying a point that has positive Tukey depth, that is, a point inside the convex hull of S.

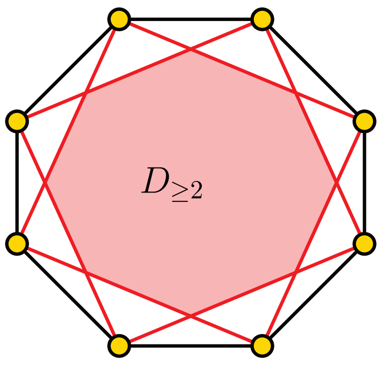

Figure 1 The Tukey layers $D_{\geq 2}$ and $D_{\geq 1}$.

To implement the exponential mechanism (i.e., to sample a point from $[0,1]^d$ appropriately), we need to compute the Tukey regions $D_{\geq k}$ and their volumes. In this section we compute these regions explicitly in $O\left(n^{1+(d-1)\lfloor d/2 \rfloor}\right)$ time. In Section 5 we will show that the cost can be reduced to $O(n^d)$.

Computing $D_{\geq k}$. We pass to the dual space, and construct the arrangement $\mathcal{A}(S^*)$ of the hyperplanes dual to the points of S. A point h^* dual to a hyperplane h supporting $D_{\geq k}$ has at least $n-k+1$ dual hypeplanes passing below h^* or incident to h^*, or, alternatively, passing above h^* or incident to h^*. Furthermore, if we move h^* slightly down in the first case (resp., up in the second case), the number of hypeplanes below (resp., above) it becomes smaller than $n-k+1$.

When h^* is a vertex of $\mathcal{A}(S^*)$ we refer to it as k-*critical*, or simply as *critical*. If $D_{\geq k}$ is non-degenerate then, by Lemma 2.2, each hyperplane h that supports a facet of $D_{\geq k}$ must be spanned by d affinely independent points of S. That is, all these hyperplanes are dual to k-critical vertices of $\mathcal{A}(S^*)$.

We compute all critical dual vertices that have at least $n-k+1$ dual hyperplanes passing below them or incident to them, or those with at least $n-k+1$ dual hyperplanes passing above them or incident to them. The intersection of the appropriate primal halfspaces that are bounded by the hyperplanes corresponding to these dual vertices is $D_{\geq k}$. This gives an algorithm for constructing $D_{\geq k}$ when it is non-degenerate. Otherwise $D_{\geq k}$ is degenerate and its volume is 0, and we need additional techniques, detailed in the next subsection, to handle such situations.

We compute the volume of each non-degenerate $D_{\geq k}$, for $k = 1, \ldots, \mathsf{td}_{\max}(S)$. We do that in brute force, by computing and triangulating $D_{\geq k}$ and adding up the volumes of the simplices in this triangulation. Then we subtract the volume of $D_{\geq k+1}$ from the volume of $D_{\geq k}$ to get the volume of D_k.

The sampling mechanism. We assign to each D_k the weight $e^{\varepsilon k/2}$ and sample a region D_k, for $k = 0, \ldots, \mathsf{td}_{\max}(S)$, with probability

$$\mu_k = \frac{e^{\varepsilon k/2}\mathrm{Vol}(D_k)}{\sum_{j\geq 0} e^{\varepsilon j/2}\mathrm{Vol}(D_j)},$$

where $\mathrm{Vol}(D_k)$ denotes the volume of D_k. Then we sample a point uniformly at random from D_k. We do this in brute force by computing D_k, triangulating it, computing the volume of each simplex, drawing a simplex from this triangulation with probability proportional to its volume, and then drawing a point uniformly at random from the chosen simplex.[2]

This is an instance of the exponential mechanism in which the score (namely the Tukey depth) has sensitivity 1, i.e., $|\mathsf{td}_S(q) - \mathsf{td}_{S'}(q)| \leq 1$ for any point $q \in [0,1]^d$, when S and S' differ by only one element. It thus follows from the properties of the exponential mechanism (Theorem 2.1) that this procedure is (purely) ε-differentially private.

Complexity. Computing the dual arrangement $\mathcal{A}(S^*)$ takes $O(n^d)$ time [10]. Assume that $D_{\geq k}$ is non-degenerate and let M_k denote the number of hyperplanes defining $D_{\geq k}$ (i.e., the hyperplanes supporting its facets). It takes $O(M_k^{\lfloor d/2 \rfloor})$ time to construct $D_{\geq k}$, as the intersection of M_k halfspaces, which is a dual version of constructing the convex hull (see [6]). Within the same asymptotic bound we can triangulate $D_{\geq k}$ and compute its volume. We obviously have $M_k = O(n^d)$, but the number can be somewhat reduced. The following lemma is known.

▶ **Lemma 3.1** (Proposition 3 in [13]). *The number of halfspaces needed to construct $D_{\geq k}$ is $O(n^{d-1})$.*

Proof. Fix a $(d-1)$-tuple σ of points of S, and consider all the relevant (closed) halfspaces, each of which is bounded by a hyperplane that is spanned by σ and another point of S, and contains at least $n-k+1$ points of S. It is easy to check that, as long as the intersection of these halfspaces is full-dimensional, it is equal to the intersection of just two of them. ◀

Summing up, we get that computing the volume of all the non-degenerate regions $D_{\geq k}$, for $k = 1, \ldots, \mathsf{td}_{\max}(S)$, takes $O\left(\sum_{k \geq 1} M_k^{\lfloor d/2 \rfloor}\right) = O\left(n^{1+(d-1)\lfloor d/2 \rfloor}\right)$ time.

Utility. We continue to assume that $D_{\geq k}$ is non-degenerate, and give a lower bound on its volume. By Lemma 2.2, such a $D_{\geq k}$ is the intersection of halfspaces, each bounded by a hyperplane that is spanned by d points of S. Denote by H the set of these hyperplanes. To obtain the lower bound, it suffices to consider the case where $D_{\geq k}$ is a simplex, each of whose vertices is the intersection point of d hyperplanes of H.

The equation of a hyperplane h that passes through d points, $a_1, \ldots, a_d$, of S is

$$\begin{vmatrix} 1 & x_1 & \cdots & x_d \\ 1 & a_{1,1} & \cdots & a_{1,d} \\ & & \vdots & \\ 1 & a_{d,1} & \cdots & a_{d,d} \end{vmatrix} = 0,$$

where $a_i = (a_{i,1}, \ldots, a_{i,d})$, for $i = 1, \ldots, d$. The coefficients of the x_i's in the equation of h are $d \times d$ subdeterminants of this determinant, where each determinant has one column of 1's, and $d-1$ other columns, each of whose entries is some $a_{i,j}$, which is a rational of the form m/X, with $0 \leq m \leq X$ (the same holds for the 1's, with $m = X$). The value of such a determinat (coefficient) is a rational number with denominator X^d. By Hadamard's

[2] A simple way to implement the last step is to draw uniformly and independently d points from $[0,1]$, compute the lengths $\lambda_1, \ldots, \lambda_{d+1}$ of the intervals into which they partition $[0,1]$, and return $\sum_{i=1}^{d+1} \lambda_i v_i$, where $v_1, \ldots, v_{d+1}$ are the vertices of the simplex.

inequality, its absolute value is at most the product of the Euclidean norms of its rows, which is at most $d^{d/2}$. That is, the numerator of the determinant is an integer of absolute value at most $d^{d/2}X^d$. The free term is a similar sub-determinant, but all its entries are the $a_{i,j}$'s, so it too is a rational with denominator X^d, and with numerator of absolute value at most $d^{d/2}X^d$. Multiplying by X^d, all the coefficients become integers of absolute value at most $d^{d/2}X^d$.

Each vertex of any region D_k of Tukey depth k, for any k, is a solution of a linear system of d hyperplane equations of the above form. It is therefore a rational number whose denominator, by Cramer's rule, is the determinant of all non-free coefficients of the d hyperplanes. This is an integer whose absolute value, again by Hadamard's inequality, is at most

$$\left(\sqrt{d}d^{d/2}X^d\right)^d \le d^{d(d+1)/2}X^{d^2}.$$

Since the free terms are also integers, we conclude that the coordinates of the intersection point are rationals with a common integral denominator of absolute value at most $d^{d(d+1)/2}X^{d^2}$.

We can finally obtain a lower bound for the nonzero volume of a simplex spanned by any $d+1$ linearly independent intersection points $v_1, \ldots, v_{d+1}$. This volume is

$$\frac{1}{d!}\begin{vmatrix} 1 & v_{1,1} & \cdots & v_{1,d} \\ 1 & v_{2,1} & \cdots & v_{2,d} \\ & & \vdots & \\ 1 & v_{d+1,1} & \cdots & v_{d+1,d} \end{vmatrix},$$

where again $v_i = (v_{i,1}, \ldots, v_{i,d})$, for $i = 1, \ldots, d+1$. Note that all the entries in any fixed row have the same denominator. The volume is therefore a rational number whose denominator is $d!$ times the product of these denominators, which is thus an integer with absolute value at most

$$d! \cdot \left(d^{d(d+1)/2}X^{d^2}\right)^d \le (dX)^{d^3}$$

(for $d \ge 2$). That is, we get the following lemma.

▶ **Lemma 3.2.** *If the volume of $D_{\ge k}$ is not zero then it is at least $1/(dX)^{d^3}$.*

Assume that the volume of $D_{\ge k}$ is not zero for $k = k_0 := n/(4d)$. Since the score of a point outside the convex hull is zero and the volume of $D_{\ge 0}$ is at most 1, we get that the probability to sample a point outside of the convex hull is at most

$$\frac{1}{e^{\varepsilon k_0}\mathrm{Vol}(D_{k_0})} \le \frac{(dX)^{d^3}}{e^{\varepsilon n/(4d)}}.$$

This inequality leads to the following theorem, which summarizes the utility that our instance of the exponential mechanism provides.

▶ **Theorem 3.3.** *If $n \ge \frac{4d^4\log(dX)}{\varepsilon} + \frac{4d}{\varepsilon}\log\frac{1}{\beta}$ and $D_{\ge n/4d}$ has non-zero volume then the exponential mechanism, implemented as above, returns a point in the convex hull with probability at least $1-\beta$, in $O\left(n^{1+(d-1)\lfloor d/2\rfloor}\right)$ time.*

4 Handling degeneracies

In general we have no guarantee that $D_{\geq n/4d}$ has non-zero volume. In this section we show how to overcome this and compute (with high probability) a point in the convex hull of any input. We rely on the following lemma, which shows that if $D_{\geq k}$ has zero volume then many points of S are in a lower-dimensional affine subspace.

▶ **Lemma 4.1.** *If $D_{\geq k}$ spans an affine subspace f of dimension j then*

$$|S \cap f| \geq n - (d-j+1)(k-1).$$

Proof. Recall that $D_{\geq k}$ is the intersection of all closed halfspaces h that contain at least $n-k+1$ points of S. Note that a halfspace that bounds $D_{\geq k}$ and whose bounding hyperplane properly crosses f, defines a proper halfspace within f, and, by assumption, the intersection of these halfspaces has positive relative volume. This means that the intersection of these halfspaces in $\mathbb{R}^d$ has positive volume too, and thus cannot confine $D_{\geq k}$ to f. To get this confinement, there must exist (at least) $d-j+1$ halfspaces in the above collection, whose intersection is f. Hence the union of their complements is $\mathbb{R}^d \setminus f$. Since this union contains at most $(d-j+1)(k-1)$ points of S, the claim follows. ◀

In what follows, to simplify the expressions that we manipulate, we use the weaker lower bound $n-(d-j+1)k$. In order for the lemma to be meaningful, we want k to be significantly smaller than the centerpoint bound $n/(d+1)$, so we set, as above, $k = n/(4d)$.

We use Lemma 4.1 to handle degenerate inputs S, using the following high-level approach. We first (privately) check whether there exists an affine proper subspace that contains many points of S. If we find such a subspace f, we recursively continue the procedure within f, with respect to $S \cap f$. Lemma 4.1 then implies that we do not lose too many points when we recurse within f (that is, $|S \cap f|$ is large), using our choice $k = n/(4d)$. Otherwise, if no such subspace exists, Lemma 4.1 implies that $D_{\geq k}$ is full-dimensional (with respect to the subspace into which we have recursed so far), so we can apply the exponential mechanism, as implemented in Section 3, and obtain a point that lies, with high probability, in the convex hull of the surviving subset of S, and thus of the full set S. We refer to this application of the exponential mechanism in the appropriate affine subspace as the *base case* of the recursion.

The points of $S \cap f$ are not on a standard grid within f. (They lie of course in the standard uniform grid G of side length $1/X$ within the full-dimensional cube, but $G \cap f$ is not a standard grid and in general has a different, coarser resolution.) We overcome this issue by noting that there always exist j coordinates, which, without loss of generality, we assume to be $x_1, \ldots, x_j$, such that f can be expressed in parametric form by these coordinates. We then project f (and $S \cap f$) onto the $x_1x_2\cdots x_j$-coordinate subspace f'. We recurse within f', where the projected points of $S \cap f$ do lie in a standard grid (a cross-section of G), and then lift the output point x_0', which lies, with high probability, in $\mathrm{conv}(S_0')$, back to a point $x' \in f$. It is straightforward to verify that if x_0' is in the convex hull of the projected points then x' is in the convex hull of $S \cap f$.

4.1 Finding an affine subspace with many points privately

For every affine subspace f, of dimension $0 \leq j \leq d-1$, spanned by some subset of (at least) $j+1$ points of G, we denote by $c(f)$ the number of points of S that f contains, and refer to it as the *size* of f.

We start by computing $c(f)$ for every subspace f spanned by points of S, as follows. We construct the (potentially degenerate) arrangement $\mathcal{A}(S^*)$ of the set S^* of the hyperplanes dual to the points of S. During this construction, we also compute the multiplicity of each flat in this arrangement, namely, the number of dual hyperplanes that contain it. Each intersection flat of the hyperplanes is dual to an affine subspace f defined by the corresponding subset of the primal points of S (that it contains), and $c(f)$ is the number of dual hyperplanes containing the flat. In other words, as a standard byproduct of the construction of $\mathcal{A}(S^*)$, we can compute the sizes of all the affine subspaces that are spanned by points of S, in $O(n^d)$ overall time.

We define

$$M_i = M_i(S) = \max\{c(f) \mid \ f \text{ is spanned by a subset of } S \text{ and is of dimension } \textit{at most } i\},$$

and compute $M_i' = M_i + Y_i$, where Y_i is a random variable drawn (independently for each i) from a Laplace distribution with parameter $b := \frac{1}{\varepsilon}$ centered at the origin. (That is, the probability density function of Y_i is $\frac{1}{2b}e^{-|x|/b} = \frac{\varepsilon}{2}e^{-\varepsilon|x|}$.)

Our algorithm now uses a given confidence parameter $\beta \in (0,1)$ and proceeds as follows. If for every $j = 0, \ldots, d-1$

$$M_j' \le n - (d-j+1)k - \frac{1}{\varepsilon}\log\frac{2}{\beta}, \tag{1}$$

we apply the base case. Otherwise, set j to be the smallest index such that

$$M_j' > n - (d-j+1)k - \frac{1}{\varepsilon}\log\frac{2}{\beta}. \tag{2}$$

Having fixed j, we find (privately) a subspace f of dimension j that contains a large number of points of S. To do so, let Z_j be the collection of all j-dimensional subspaces that are spanned by $j+1$ affinely independent points of the grid G (not necessarily of S). We apply the exponential mechanism to pick an element of Z_j, by assigning a *score* $s(f)$ to each subspace of Z_j, which we set to be

$$s(f) = \max\{0, c(f) - M_{j-1}\} \ ,$$

if $j \ge 1$, and $s(f) = c(f)$ if $j = 0$. Note that by its definition, $s(f)$ is zero if f is not spanned by points of S. Indeed, in this case the $c(f)$ points contained in f span some subspace of dimension $\ell \le j-1$ and therefore M_{j-1} must be at least as large as $c(f)$. We will shortly argue that $s(f)$ has sensitivity at most 2 (Lemma 4.4), and thus conclude that this step preserves the differential privacy of the procedure.

We would like to apply the exponential mechanism as stated above in time proportional to the number of subspaces of non-zero score, because this number depends only on n (albeit being exponential in d) and not on (the much larger) X. However, to normalize the scores to probabilities, we need to know the number of elements of Z_j with zero score, or alternatively to obtain the total number of subspaces spanned by $j+1$ points of G (that is, the size of Z_j).

We do not have a simple expression for $|Z_j|$ (although this is a quantity that can be computed, for each j, independently of S, once and for all), but clearly $|Z_j| \le X^{d(j+1)}$. We thus augment Z_j (only for the purpose of analysis) with $X^{d(j+1)} - |Z_j|$ "dummy" subspaces, and denote the augmented set by Z_j', whose cardinality is now exactly $X^{d(j+1)}$. We draw a subspace f from Z_j' using the exponential mechanism. To do so we need to compute, for each score $s \ge 0$, the number N_s of elements of Z_j' that have score s, give each such element weight $e^{\varepsilon s/4}$, choose the index s with probability proportional to $N_s e^{\varepsilon s/4}$, and then choose uniformly a subspace from those with score s. It is easy to check that this is indeed an implementation of the exponential mechanism as described in Section 2.1.

If the drawing procedure decides to pick a subspace that is not spanned by points of S, or more precisely decides to pick a subspace of score 0, we stop the whole procedure, with a failure. If the selected score is positive, the subspace to be picked is spanned by $j+1$ points of S, and those subspaces are available (from the dual arrangement construction outlined above). We thus obtain a selected subspace f (by randomly choosing an element from the available list of these subspaces), and apply the algorithm recursively within f, restricting the input to $S \cap f$. (Strictly speaking, as noted above, we apply the algorithm to a projection of f onto a suitable j-dimensional coordinate subspace.)

It follows that we can implement the exponential mechanism on all subspaces in Z'_j in time proportional to the number of subspaces spanned by points of S, which is $O(n^d)$, and therefore the running time of this subspace selection procedure (in each recursive call) is $O(n^d)$.

4.1.1 Privacy analysis

▶ **Lemma 4.2.** *Let $S_1 = S_0 \cup \{x\}$ and $S_2 = S_0 \cup \{y\}$ be two neighboring data sets. Then, for every $i = 0, \dots, d-1$, we have $|M_i(S_1) - M_i(S_2)| \le 1$.*

Proof. Let f be a subspace of dimension at most i that is spanned by points of S_1 and contains $M_i(S_1)$ such points. If f does not contain x then f is also a candidate for $M_i(S_2)$, so in this case $M_i(S_2) \ge M_i(S_1)$. If f does contain x then $S_1 \cap f \setminus \{x\} \subseteq S_0$ spans a subspace f' of f which is of dimension at most i, so it is a candidate for $M_i(S_2)$. Since it does not contain x (and may contain y) we have in this case that $M_i(S_2) \ge M_i(S_1) - 1$. Therefore we can conclude that in anycase $M_i(S_2) \ge M_i(S_1) - 1$. A symmetric argument shows that $M_i(S_1) \ge M_i(S_2) - 1$. Combining these two inequalities the lemma follows. ◀

▶ **Lemma 4.3.** *The value of each M'_i, for $i = 0, \dots, d-1$, is ε-differentially private.*

Proof. This follows from standard arguments in differential privacy (e.g., see [9, 19]), since, by Lemma 4.2, M_i is of *sensitivity* 1 (in the sense shown in the lemma). ◀

Since we choose j by comparing each of the M'_j's to a value which is the same for neighboring data sets S_1 and S_2 (which have the same cardinality n), Lemma 4.3 implies that the choice of the dimension j is differentially private.

The next step is to choose the actual subspace f in which to recurse. The following lemma implies that this step too is differentially private.

▶ **Lemma 4.4.** *Let S_1 and S_2 be as in Lemma 4.2. Then, for each $j = 0, \dots, d-1$ and for every subspace $f \in Z'_j$, we have $|s_{S_1}(f) - s_{S_2}(f)| \le 2$.*

Proof. Fix j and $f \in Z'_j$. Clearly, $|c_{S_1}(f) - c_{S_2}(f)| \le 1$, and, by Lemma 4.2, M_{j-1} is also of *sensitivity* 1, and the claim follows. ◀

▶ **Lemma 4.5.** *The subspace-selection procedure described in this section (with all its recursive calls) is $2d^2\varepsilon$-differentially private.*

Proof. By Lemma 4.3 the computation of each M'_i is ε-differentially private, and by Lemma 4.4 the exponential mechanism on the scores $s(f)$ is also ε-differentially private. Since we compute at most d values M'_i at each step, and we recurse at most d times, the claim follows by composition [9, 19]. ◀

▶ Remark 4.6. We can save a factor of d in Lemma 4.5 by using a framework called *the sparse vector technique*, see e.g., [9].

4.1.2 Utility analysis

The following lemma is the key for our utility analysis.

▶ **Lemma 4.7.** *Let $\beta \in (0,1)$ be a given parameter. For $k = \frac{n}{4d}$ and for every $j = 0, \dots, d-1$ the following properties hold.*

(i) *If $M_j \geq n - (d-j+1)k$ then, with probability at least $1-\beta$,*

$$M_j' \geq n - (d-j+1)k - \frac{1}{\varepsilon}\log\frac{2}{\beta}.$$

(ii) *On the other hand, if $M_j \leq n - (d-j+1)k - \frac{2}{\varepsilon}\log\frac{2}{\beta}$, then, with probability at least $1-\beta$,*

$$M_j' \leq n - (d-j+1)k - \frac{1}{\varepsilon}\log\frac{2}{\beta}.$$

Proof. (i) follows since the probability of the Laplace noise Y_j to be smaller than $-\frac{1}{\varepsilon}\log\frac{2}{\beta}$ is at most β, and (ii) follows since the probability of Y_j to be larger than $\frac{1}{\varepsilon}\log\frac{2}{\beta}$ is also at most β. ◀

We say that (the present recursive step of) our algorithm *fails* if one of the complements of the events specified in Lemma 4.7 happens, that is, the step fails if for some j, either (i) $M_j \geq n-(d-j+1)k$ and $M_j' < n-(d-j+1)k-\frac{1}{\varepsilon}\log\frac{2}{\beta}$, or (ii) $M_j \leq n-(d-j+1)k-\frac{2}{\varepsilon}\log\frac{2}{\beta}$ and $M_j' > n-(d-j+1)k-\frac{1}{\varepsilon}\log\frac{2}{\beta}$. Otherwise we say that (this step of) our algorithm *succeeds*.[3]

It follows from Lemma 4.1 that if the algorithm succeeds and applies the base case then $D_{\geq k}$ is full dimensional. Furthermore, if the algorithm succeeds and decides to recurse on a subspace of dimension j (according to the rule in (1) and (2)) then, for every $\ell < j$, $M_\ell \leq n-(d-\ell+1)k$ and $M_j \geq n-(d-j+1)k-\frac{2}{\varepsilon}\log\frac{2}{\beta}$. The following lemma is an easy consequence of this reasoning.

▶ **Lemma 4.8.** *If the algorithm succeeds, with dimension j, and applies the exponential mechanism to pick a j-dimensional subspace, then there exists a j-dimensional subspace f with score $s(f) = M_j - M_{j-1} \geq k - \frac{2}{\varepsilon}\log\frac{2}{\beta}$. Furthermore, if $k \geq \frac{8d^2}{\varepsilon}\log X + \frac{8}{\varepsilon}\log\frac{1}{\beta}$ then, with probability at least $1-\beta$, the exponential mechanism picks a subspace f with $s(f) \geq M_j - M_{j-1} - \frac{k}{2} \geq \frac{k}{2} - \frac{2}{\varepsilon}\log\frac{2}{\beta}$.*

Proof. The first part of the lemma follows from the definition of success, as just argued. For the second part notice that, since $|Z_j'| \leq X^{d^2}$, the probability of drawing a subspace $f' \in Z_j'$ of score smaller than $M_j - M_{j-1} - \frac{k}{2}$ is at most

$$X^{d^2} \cdot \frac{e^{\varepsilon(M_j - M_{j-1} - k/2)/4}}{e^{\varepsilon(M_j - M_{j-1})/4}} = X^{d^2} \cdot e^{-\varepsilon k/8}\ .$$

This expression is at most β if $k \geq \frac{8d^2}{\varepsilon}\log X + \frac{8}{\varepsilon}\log\frac{1}{\beta}$. ◀

[3] Note that there is a “grey zone” of values of M_j between these two bounds, in which the step always succeeds.

If follows that if our algorithm succeeds and recurses in a subspace f of dimension j then, with probability at least $1-\beta$,

$$\begin{aligned} c(f) &\geq M_{j-1} + s(f) \geq M_j - \frac{k}{2} \\ &\geq n - (d-j+1)k - \frac{2}{\varepsilon}\log\frac{1}{\beta} - \frac{k}{2} \geq n - \left(d-j+\frac{3}{2}\right)k - \frac{2}{\varepsilon}\log\frac{1}{\beta}. \end{aligned}$$

That is, when we recurse in f of dimension j we lose at most $\left(d-j+\frac{3}{2}\right)k + \frac{2}{\varepsilon}\log\frac{1}{\beta}$ points. Denote by $d_0 = d, d_1, \ldots, d_t$ the sequence of dimensions into which the procedure recurses (reaching the base case at dimension $d_t \geq 0$). Hence, keeping k fixed throughout the recursion, at the r-th recursive step we lose at most $\left(d_r - d_{r+1} + \frac{3}{2}\right)k + \frac{2}{\varepsilon}\log\frac{1}{\beta}$ points. Summing up these losses over $r = 0, \ldots, t-1$, the total loss is at most

$$(d_0 - d_t)k + \frac{3}{2}kt + \frac{2t}{\varepsilon}\log\frac{1}{\beta} \leq \frac{5d}{2}\cdot k + \frac{2d}{\varepsilon}\log\frac{1}{\beta}.$$

Substituting $k = \frac{n}{4d}$, we get that the total number of points that we loose is at most $\frac{2n}{3}$ if $n = \Omega\left(\frac{d}{\varepsilon}\log\frac{1}{\beta}\right)$, with a sufficiently large constant of proportionality.

Notice that we keep k fixed throughout the recursion and n may decrease. Consequently, if n' is the number of points in some recursive call in some dimension $\ell < d$, then $n' \geq \frac{n}{3}$ and therefore $k = \frac{n}{4d} \leq \frac{3n'}{4d}$ which is still smaller than the centerpoint guarantee of $\frac{n'}{\ell+1}$.

As described, our subspace-selection procedure (with all its recursive calls) is $2d^2\varepsilon$-differentially private. Dividing ε by $2d^2$ we get that our subspace-selection procedure is ε-differentially private, and that the total number of points we lose is much smaller than n if $n = \Omega\left(\frac{d^3}{\varepsilon}\log\frac{1}{\beta}\right)$.

Recall Section 3, where we showed that we need $n = \Omega\left(\frac{d^4 \log dX}{\varepsilon}\right)$ for the (ε-differentially private) base case to work correctly. (Recall also that the base case is actually applied in a suitable projection of the terminal subspace onto some coordinate-frame subspace of the same dimension, and that the above lower bound on n suffices for any such lower-dimensional instance too.)

The following theorem summarizes the result of this section.

▶ **Theorem 4.9.** *If $n = \Omega\left(\frac{d^4 \log dX}{\varepsilon} + \frac{d^3 \log\frac{1}{\beta}}{\varepsilon}\right)$, our algorithm (including all recursive calls and the base case) is ε-differentially private, runs in $O\left(n^{1+(d-1)\lfloor d/2\rfloor}\right)$ time, and finds a point of depth at least $k = \frac{n}{4d}$ with probability at least $1-2d^2\beta$.*

Proof. The privacy statement follows by composition, using Lemma 4.5 and the privacy properties of the exponential mechanism. The confidence bound follows since the probability of failure of the recursive call in a subspace of dimension j is at most $(j+1)\beta$. The running time of the algorithm is dominated by the running time of the exponential mechanism that we perform at the bottom (base case) of the recursion. ◀

5 An $O(n^d)$ algorithm via volume estimation

The upper bound on the running time in Theorem 4.9 is dominated by the running time of the base case, in which we compute all the regions $D_{\geq \ell}$ explicitly, which takes $n^{O(d^2)}$ time. To reduce this cost, we use instead a mechanism that (a) estimates the volume of D_k to a sufficiently small relative error, and (b) samples a random point "almost" uniformly from D_k.

We show how to accomplish (a) and (b) using the volume estimation mechanisms of Lovász and Vempala [14] and later of Cousins and Vempala [7]. We also show how to use these procedures to implement approximately the exponential mechanism described in Section 3. These algorithms are Monte Carlo, so they fail with some probability, and when they fail we may lose our ε-differential privacy guarantee. As a result, the modified algorithm will not be purely differentially private, as the one in Section 3, and we will only be able to guarantee that it is (ε, δ)-differentially private, for any prescribed $\delta > 0$. The following theorem is our main result. We prove it in the full version of this paper [12].

▶ **Theorem 5.1.** *Given $n = \Omega\left(\frac{d^4 \log \frac{dX}{\delta}}{\varepsilon}\right)$ points, our algorithm (including all recursive calls and the base case) is (ε, δ)-differentially private, runs in $O\left(n^d\right)$ time, and finds a point of depth at least $k = \frac{n}{4d}$ with probability at least $1 - \delta$.*

6 Conclusions

We gave an $O(n^d)$-time algorithm for privately computing a point in the convex hull of $\Omega(d^4 \log X)$ points with coordinates that are multiples of $1/X$ in $[0, 1]$. Even though this gives a huge improvement of what was previously known and requires some nontrivial technical effort, and sophisticated sampling and volume estimation tools, this running time is still not satisfactory for large values of d. The main hurdle in improving it further is the nonexistence of efficient algorithms for computing Tukey depths and Tukey levels.

The main question that we leave open is whether there exists a differentially private algorithm for this task which is polynomial in n and d? (when the input size, n, is still polynomial in $\log X$ and d).

References

1 Amos Beimel, Shiva Prasad Kasiviswanathan, and Kobbi Nissim. Bounds on the sample complexity for private learning and private data release. In *TCC*, volume 5978 of *LNCS*, pages 437–454. Springer, 2010.

2 Amos Beimel, Shay Moran, Kobbi Nissim, and Uri Stemmer. Private center points and learning of halfspaces. In *Conference on Learning Theory (COLT)*, pages 269–282, 2019.

3 Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. In *APPROX-RANDOM*, volume 8096 of *LNCS*, pages 363–378. Springer, 2013.

4 Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated cdp. In *50th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 74–86, 2018.

5 Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 634–649, 2015.

6 Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10:377–409, 1993.

7 Ben Cousins and Santosh S. Vempala. Gaussian cooling and $O^*(n^3)$ algorithms for volume and gaussian volume. *SIAM J. Comput.*, 47(3):1237–1273, 2018.

8 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *TCC*, volume 3876 of *LNCS*, pages 265–284. Springer, 2006.

9 Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4), 2014.

10 Herbert Edelsbrunner, Raimund Seidel, and Micha Sharir. On the zone theorem for hyperplane arrangements. *SIAM J. Comput.*, 22(2):418–429, 1993.

11 Haim Kaplan, Katrina Ligett, Yishay Mansour, Moni Naor, and Uri Stemmer. Privately learning thresholds: Closing the exponential gap. *CoRR*, 2019. `arXiv:1911.10137`.

12 Haim Kaplan, Micha Sharir, and Uri Stemmer. How to find a point in the convex hull privately, 2020. `arXiv:2003.13192`.

13 Xiaohui Liu, Karl Mosler, and Pavlo Mozharovskyi. Fast computation of Tukey trimmed regions and median in dimension p $>$ 2. *J. of Comput. and Graph. Stat.*, 28(3):682–697, 2019.

14 László Lovász and Santosh S. Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *J. Comput. Syst. Sci.*, 72(2):392–417, 2006.

15 Jiří Matousek. *Lectures on Discrete Geometry.* Springer-Verlag, Berlin, Heidelberg, 2002.

16 Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 94–103, 2007.

17 Peter J. Rousseeuw and Ida Ruts. Constructing the bivariate Tukey median. *Statistica Sinica*, 8(3):827–839, 1998.

18 John W. Tukey. Mathematics and the picturing of data. In *Proc. of the International Congress of Mathematicians*, volume 2, page 523–531, 1975.

19 Salil Vadhan. The complexity of differential privacy. In Yehuda Lindell, editor, *Tutorials on the Foundations of Cryptography: Dedicated to Oded Goldreich*, pages 347–450. Springer, 2017.