Near Optimal Distributed Learning of Halfspaces with Two Parties

Mark Braverman MBRAVERM@GMAIL.COM
Princeton University

Gillat Kol GILLAT.KOL@GMAIL.COM
Princeton University

Shay Moran SHAYMORAN1@GMAIL.COM
Technion and Google Research

Raghuvansh R. Saxena RRSAXENA@PRINCETON.EDU
Princeton University

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

Distributed learning protocols are designed to train on distributed data without gathering it all on a single centralized machine, thus contributing to the efficiency of the system and enhancing its privacy. We study a central problem in distributed learning, called *distributed learning of halfspaces*: let $U \subseteq \mathbb{R}^d$ be a known domain of size n and let $h : \mathbb{R}^d \to \mathbb{R}$ be an unknown target affine function.[1] A set of *examples* $\{(u, b)\}$ is distributed between several parties, where $u \in U$ is a point and $b = \mathsf{sign}(h(u)) \in \{\pm 1\}$ is its label. The parties' goal is to agree on a classifier $f : U \to \{\pm 1\}$ such that $f(u) = b$ for every input example (u, b).

We design a protocol for the distributed halfspace learning problem in the two-party setting, communicating only $\tilde{O}(d \log n)$ bits. To this end, we introduce a new tool called *halfspace containers*, that is closely related to *bracketing numbers* in statistics and to *hyperplane cuttings* in discrete geometry, and allows for a compressed approximate representation of every halfspace. We complement our upper bound result by an almost matching $\tilde{\Omega}(d \log n)$ lower bound on the communication complexity of any such protocol.

Since the distributed halfspace learning problem is closely related to the *convex set disjointness* problem in communication complexity and the problem of *distributed linear programming* in distributed optimization, we also derive upper and lower bounds of $\tilde{O}(d^2 \log n)$ and $\tilde{\Omega}(d \log n)$ on the communication complexity of both of these basic problems.

Keywords: Distributed Learning, Communication Complexity

1. Introduction

Modern applications of machine learning often involve data obtained from several different sources. For example, in healthcare related applications, data is collected from hospitals and labs in remote locations. Another host of examples involves algorithms that are trained on personal data (*e.g.*, a music recommendation app which is trained on preferences made by numerous users). The collection of data in these applications may be costly or even infeasible due to its sheer size.

1. In practice, the domain U is defined implicitly by the representation of d-dimensional vectors which is used in the protocol.

Furthermore, data collection may be problematic from a privacy perspective in contexts where it contains sensitive information (*e.g.*, patients' data, financial data, personal data on smartphones).

Such applications raise the need for algorithms that are able to train on *distributed* data without gathering it all on a single centralized machine. Consequently, tech companies invest significant efforts in developing suitable technologies; one notable example is Google's *Federated Learning* project (Konečný et al., 2016) which attempts to train a centralized model on data which is distributed on different clients (say mobile phones). The high-level approach here is that each client sends *compressed updates* to a centralized model.

1.1. Distributed Learning of Halfspaces

This work focuses on the problem of *distributed learning of halfspaces* (*a.k.a linear classifiers*). Linear classifiers form the backbone of many popular learning algorithms and are very well studied in the centralized setting: they date back to the seminal *Perceptron Algorithm* from the 50's (Rosenblatt, 1958), and form the basis of more modern algorithms such as kernel machines and neural nets.

In the distributed setting, (*improper*) learning of halfspaces refers to the following task: a set of *examples* is distributed between several parties. Each example consists of a pair (u, b), where $u \in U$ is a feature vector[2], $b = \mathsf{sign}(h(u)) \in \{\pm 1\}$ is the label, and $h : \mathbb{R}^d \to \mathbb{R}$ is the (unknown) target linear function. The parties' goal is to agree[3] on a classifier $f : U \to \{\pm 1\}$ such that $f(u) = b$ for every input example (u, b), while minimizing the amount of communication[4]. In practice, the domain U is often defined implicitly by the encoding used to represent and transmit d-dimensional vectors. (So, if each entry is encoded using B bits in binary-representation then U is a grid of size $n = 2^{d \cdot B}$.) For this reason, it is usually the case that the dimension d is much smaller than the domain size n. This problem received considerable attention both in theory and practice, see, *e.g.*, Jian-Pei Zhang et al. (2005); Navia-Vazquez et al. (2006); Chang et al. (2007); McDonald et al. (2010); Forero et al. (2010); Daumé III et al. (2012); Balcan et al. (2012); Kane et al. (2019).

We give (a nearly) tight bound of $\tilde{\Theta}(d \log n)$ on the communication complexity of this problem in the two-party setting.

Theorem 1 (Informal) *Let $d, n \geq 2$ and let $U \subseteq \mathbb{R}^d$ be a domain with n points. The communication complexity of distributed learning of halfspaces in the two-party setting is $\tilde{\Theta}(d \log n)$.*

Although this result concerns the two-party setting, we note that the ideas and techniques used in its proof can be extended to more complex distributed models. We focus here on the two-party setting as it already requires substantial new ideas and provides a clean platform to demonstrate them. Our result also addresses the case where the parties are required to have zero misclassification error,

2. In this context, it may be natural to think of the domain U as a grid, a discretized manifold, or any other domain that arises naturally from euclidean representations of data.
3. The requirement that the parties *agree on a consistent output hypothesis* is important, as, for instance, it helps rule out efficient "memorization-based" protocols where, *e.g.*, each party simply memorizes its samples and outputs a hypothesis consistent with the samples. This is because the party that memorizes its samples needs to send them to the other party so that they agree on the same output hypothesis. In fact, our protocol in this paper satisfies an even stronger property: the output hypothesis can be deduced from the transcript of the protocol. It can be shown that this implies strong generalization bounds.
4. Note that f may not be consistent with any halfspace, as we consider improper learning.

i.e., they want to output a hypothesis that correctly classifies all the points. Getting results for the non-zero error case is also interesting. We defer generalizations for future work.

We view the $\tilde{O}(d \log n)$ upper bound as the main technical contribution of this paper. The protocol achieving it is *deterministic* and exploits geometric tools we design for this purpose. Furthermore, this protocol can be implemented *efficiently* (*i.e.*, the internal computations take $\mathrm{poly}(n, d)$ time). See Appendix B.4 for a more detailed discussion. The upper bound improves upon a previous bound of $O(d \log^2 n)$ by Daumé III et al. (2012) and Balcan et al. (2012) which rely on distributed implementations of boosting algorithms.

The lower bound (which is tight up-to a "$\log d$" term) also applies to *randomized* protocols and uses a geometric embedding as well as a communication complexity-style direct sum argument. It improves upon a previous lower bound of $\Omega(d + \log n)$ by Kane et al. (2019). (We note however that the bound by Kane et al. (2019) applies in a more general distributed model, see Section 1.4 for more details.)

1.2. Convex Set Disjointness and Distributed LP Feasibility

To obtain the bounds in Theorem 1, we study the related *Convex Set Disjointness* problem, introduced in this context by Kane et al. (2019). Convex Set Disjointness is a communication problem over a set $U \subseteq \mathbb{R}^d$, in which two parties, Alice and Bob, hold input sets $X, Y \subseteq U$. The parties' mutual goal is to decide whether $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$, where $\mathrm{conv}(\cdot)$ denotes the convex hull operator. We denote this problem CSD_U and will mostly be interested in the case where $n = |U|$ satisfies $n \gg d$.

CSD is equivalent to the problem of *distributed Linear Programming feasibility* in distributed optimization. *Linear Programming* (LP) is one of the most basic primitives in optimization. In the associated decision problem, called *LP feasibility*, the goal is to determine whether a system of linear inequalities (also called constraints) is satisfiable. In distributed LP feasibility, the constraints are partitioned between several parties. Indeed, LP feasibility is equivalent to CSD, albeit in a dual formulation where constraints and points are interchanged: disjointness of the convex hulls amounts to the existence of a separating hyperplane, which, from a dual perspective, corresponds to a point that satisfies all of the constraints.

We improve the known bounds on the communication complexity of CSD and LP feasibility in the two-party setting.

Theorem 2 (Informal) *Let $d, n \geq 2$ and let $U \subseteq \mathbb{R}^d$ be a domain with n points. Then, the communication complexity of CSD_U (and therefore also of the corresponding two-party LP feasibility problem) is at most $\tilde{O}(d^2 \log n)$ and at least $\tilde{\Omega}(d \log n)$.*

Our CSD upper bound is achieved by a deterministic protocol and improves upon two previous bounds by Vempala et al. (2020): the first gives an $O(d^3 \log^2 n)$ upper bound for general sets U and the other gives an $O(d^2 \log^2 d \log n)$ bound for some natural special cases (see Section 1.4 for a more detailed comparison). Our lower bound applies even when the protocol is randomized and improves upon Vempala et al. (2020) who derive a similar lower bound of $\Omega(d \log n)$ in the deterministic setting and a lower bound of $\Omega(\log n)$ in the randomized setting.

Settling the communication complexity of CSD (and therefore also of LP feasibility) is an outstanding open problem[5].

Connection to Distributed Learning. As mentioned above, we obtain our bounds for distributed learning by studying the CSD problem. We note that our CSD protocol has the additional property that when the convex hulls are disjoint, it generates a "certificate" for this fact (*i.e.*, a separating function f). It is easy to see that any such protocol also solves the learning problem, where one party only gets positive examples and the other only gets negative examples. We show that the mixed case can also be handled.

While our CSD protocol can be used to solve the learning problem, the communication of this protocol is off by a factor of d from our stated optimal result for the learning problem. To obtain the optimal result, we consider a restricted version of CSD, called PromiseCSD, where we are *promised* that if the convex hulls of X and Y intersect, then X and Y intersect. For this restricted problem we are able to construct an optimal $\tilde{O}(d \log n)$ protocol. Observe that any certificate producing PromiseCSD protocol also solves the learning problem, as in the learning problem we are promised that there exists a separating hyperplane, and thus that the inputs are disjoint.

In the opposite direction, it is not hard to see that any learning algorithm can be used to decide PromiseCSD (see Section 2.4). We also mention that a *proper*[6] learning protocol can be used to decide CSD. Therefore, obtaining the upper bound of Theorem 1 with a proper learner will improve the upper bound of Theorem 2 and show that the $\tilde{\Omega}(d \log n)$ lower bound is essentially tight.

Connections to classical communication complexity problems. The CSD problem can be seen as a geometric interpolation between two central problems in communication complexity, namely, *Set Disjointness* (when $d \geq n-1$), and *Greater-Than* (when $d = 1$). Indeed, if $d \geq n-1$ then one can pick the n points in $U \subseteq \mathbb{R}^d$ to be affinely independent, which implies that $X \cap Y = \emptyset \iff \mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$. Therefore, in this case the communication complexity of CSD_U is the same as that of Set Disjointness, which is $\Theta(n)$ (Kalyanasundaram and Schintger, 1992). In the other extreme, if $d = 1$ then U is a set of n points on the real line and CSD_U boils down to comparing the two extreme points in Alice's input with the two extreme points in Bob's input (see Figure 1). Thus, the case of $d = 1$ is equivalent to the Greater-Than problem on $\log n$ bits, whose communication complexity is $\Theta(\log n)$ in the deterministic setting and $\Theta(\log \log n)$ in the randomized setting (with constant error) (Feige et al., 1994; Viola, 2013).

1.3. Geometric Tools

In this paper, we take a geometric approach when designing our protocols. This is widely different than the techniques used by previous works, who mainly focused on implementing distributed versions of known learning algorithms. In this section, we describe some of the geometric tools that go into our constructions.

Our protocol for $\mathsf{PromiseCSD}_U$ uses the observation that if X and Y can be separated by a hyperplane, then one of these sets is contained in a halfspace that contains at most $n/2$ points from U. The protocol asks Alice and Bob to check if their input lies in such a halfspace, and if so send this halfspace to the other party. The parties then ignore all the points outside the halfspace

5. Settling this problem may also shed light on the communication complexity of distributed versions of related problems studied in the literature (*i.e.*, super-linear lower bounds for different versions of distributed linear programming, max-flow, and perfect matching).
6. That is, the function f output by the learning algorithm is consistent with a hyperplane.

Figure 1: Convex Set Disjointness in 1D: the convex hull of Alice's input (blue points) is disjoint from the convex hull of Bob's input (red points) if and only if $\mathbf{x_{right}} < \mathbf{y_{left}}$ or $\mathbf{y_{right}} < \mathbf{x_{left}}$. Thus, this case amounts to deciding (2 instances of) the *Greater-Than* problem on $\log n$ bits. The disjointness condition does not hold for the depicted scenario, and indeed, the convex hulls intersect.

and reduce the size of U by half. The protocol repeats this "divide and conquer" process $\log n$ times, and since $O(d \log n)$ bits are required to specify a hyperplane (up-to equivalence), the overall communication is $O(d \log^2 n)$.

To get the optimal $\mathsf{PromiseCSD}_U$ protocol, removing the extra $\log n$ factor, we design a new tool called *halfspace containers*, that is also of independent interest (see an additional application below). Roughly speaking, it allows us to get a compressed approximate representation of each halfspace H. More formally, for $U \subseteq \mathbb{R}^d$ denote by $\mathsf{HS}(U) = \{H \cap U : H \text{ is a halfspace in } \mathbb{R}^d\}$ the family of all halfspaces restricted to U.

Theorem 3 (Halfspace Containers) *Let $U \subseteq \mathbb{R}^d$ and $\varepsilon > 0$. Then, there exists a family $\mathcal{C} \subseteq 2^U$ of size $(d/\varepsilon)^{O(d)}$ such that*

$$(\forall H \in \mathsf{HS}(U))(\exists C \in \mathcal{C}) \ : \ H \subseteq C \ \text{ and } \ |C \setminus H| \leq \varepsilon |U|.$$

The set $\mathcal{C}$ in the above theorem is called a family of ε-*containers* for $\mathsf{HS}(U)$. Each container $C \in \mathcal{C}$ in our construction has a rather simple form: it is the complement of an intersection of $\leq d$ halfspaces. Additionally, our proof of Theorem 3 is constructive and implies a (randomized) algorithm which, given an input halfspace H, finds in $\mathrm{poly}(n, d, 1/\varepsilon)$ time an ε-container $C \in \mathcal{C}$ for it. (See Appendix B.4 for a more detailed discussion).

Theorem 3 extends to arbitrary distributions: for every probability measure P on $\mathbb{R}^d$ there is a family $\mathcal{C}$ of subsets of $\mathbb{R}^d$ whose size is $(d/\varepsilon)^{O(d)}$ such that for every halfspace H there is $C \in \mathcal{C}$ such that $H \subseteq C$ and $P(C \setminus H) \leq \varepsilon$. To see this, note that Theorem 3 handles the case when P is finitely supported, and use the fact that each distribution can be approximated (in an appropriate sense) by finitely supported distributions.

We also mention that a weaker object (that is insufficient for our purposes), obtained by removing the requirement that $H \subseteq C$, and demanding that $|H \Delta C| \leq \varepsilon |U|$ is called an ε-*cover* for $\mathsf{HS}(U)$. A classical result by Haussler implies that $\mathsf{HS}(U)$ (and, in fact, any family with VC dimension d) has an ε-cover of size roughly $(1/\varepsilon)^d$ (Haussler, 1995). An interesting topic for future study is whether our container construction can be improved to match Haussler's parameters and whether other natural VC classes have small containers (see discussion in Appendix B.1). We remark that matching Haussler's bound with respect to container will tighten our bounds by removing the $\log d$ gap between the upper and lower bounds.

Bracketing Numbers in Statistics. Theorem 3 can be used to get a distribution-free bound on the *bracketing number* of halfspaces. Bracketing is a technique for deriving uniform laws of large numbers for empirical processes (see, *e.g.*, Adams and Nobel (2010)). Let $\mathcal{H}$ be a family of events

in a probability space $(\mathcal{X}, \Omega, P)$. A pair of events $F, G \subseteq \mathcal{X}$ is called an ε-bracket for $H \in \mathcal{H}$ if $F \subseteq H \subseteq G$ and $P(G \setminus F) \leq \varepsilon$. A family of events $\mathcal{B}$ is an ε-bracketing for $\mathcal{H}$ if every $H \in \mathcal{H}$ admits an ε-bracket in $\mathcal{B}$. The ε-bracketing number of $\mathcal{H}$ (*a.k.a*, bracketing entropy) is defined as $\log|\mathcal{B}|$ for the smallest possible ε-bracketing $\mathcal{B}$.

The key property is that if $|\mathcal{B}|$ is small then it implies a uniform law of large numbers with respect to all events in $\mathcal{H}$: let $x_1 \dots x_n \sim P$ be independent samples. A union bound over $\mathcal{B}$ implies that for a sufficiently large n, $P_n(B) \approx P(B)$ simultaneously for all $B \in \mathcal{B}$, where $P_n(B) = \frac{|\{i:\, x_i \in B\}|}{n}$. Consequently, by bracketing it follows that also $P_n(H) \approx P(H)$ for all $H \in \mathcal{H}$. Note that this reasoning applies even when $\mathcal{H}$ is infinite (the crucial property is that $\mathcal{B}$ is small, which allows for a union bound).

Brackets are typically used to prove distribution-dependent laws of large numbers. For example, they imply a uniform law of large numbers for the family of convex sets in $\mathbb{R}^2$ when the underlying distribution is uniform over $[0,1]^2$ (see page 22-24 in Pollard (1984)). Note that the family of convex sets has an infinite VC dimension and thus one must use a distribution dependent argument. Another notable advantage of brackets over other techniques such as VC dimension, Rademacher complexity, covering numbers, and others is that brackets hold even when the random sample $x_1 \dots x_n \sim P$ is not independent. As such, brackets are applicable to more general ergodic processes (Adams and Nobel, 2010). For further reading, see the books (van der Vaart et al., 1996; Dudley, 1999).

Theorem 3 implies a distribution-free bound of $O(d \log(d/\varepsilon))$ on the *bracketing number* of halfspaces in $\mathbb{R}^d$ as follows: given a halfspace H, construct an ε-bracket F, G for H by taking $F \supseteq H$ to be an $\frac{\varepsilon}{2}$-container for H and $G \subseteq H$ to be the complement of an $\frac{\varepsilon}{2}$-container for the complement of H (we use here the fact that a complement of a halfspace is a halfspace). Prior to our work, it was known that halfspaces admit finite bracketing numbers, albeit in a distribution dependent manner (*i.e.*, the bound on $|\mathcal{B}|$ depends on the distribution P (Adams and Nobel, 2010)).

Dual Carathéodory. A key ingredient in our construction of the family of containers is a *dual variant of Carathéodory's theorem* which we prove.

Let $\mathcal{Q} \subseteq \mathbb{R}^d$ be a polytope. There are two natural ways of representing $\mathcal{Q}$: (i) as the convex hull of its vertices, (ii) as an intersection of halfspaces. *Carathéodory's Theorem*, a fundamental statement in convex geometry (Carathéodory, 1907), implies that if $\mathcal{Q}$ is the convex hull of a few *vertices* then it can be covered by a few simplices. Quantitatively, if $\mathcal{Q}$ has n vertices then it can be covered by at most n^{d+1} sets of the form $\mathrm{conv}(\{x_0, \dots, x_d\})$, where the x_i's are vertices of $\mathcal{Q}$.

How many subsimplices are needed in order to cover a polytope $\mathcal{Q}$ defined as the intersection of n halfspaces? A bound of $n^{d(d+1)}$ follows by the previous bound[7]. We prove the following theorem that achieves a quadratic improvement in the exponent:

Theorem 4 (Dual Carathéodory) *Let $\mathcal{Q} \subseteq \mathbb{R}^d$ be a polytope obtained as the intersection of n halfspaces. Then, $\mathcal{Q}$ can be covered using at most n^d subsimplices of the form $\mathrm{conv}(\{x_0, \dots, x_d\})$, where the x_i's are vertices of $\mathcal{Q}$.*

1.4. Related Work

Lovász and Saks (1993) studied a variant of CSD where the goal is to decide whether the convex hulls intersect in a point from U. This variant exhibits a very different behaviour, even in

7. The number of vertices in $\mathcal{Q}$ is at most n^d, as every vertex is defined by d hyperplanes.

dimension $d = 2$: indeed, if U is in convex position[8] (say n points on the unit circle), then this becomes equivalent to the classical Set Disjointness problem whose communication complexity is $\Theta(n)$, whereas for our CSD, any planar instance $U \subseteq \mathbb{R}^2$ can be decided using $O(\log n)$ bits.

Variants of CSD were also considered by several works in distributed machine learning and distributed optimization (see, *e.g.*, Balcan et al. (2012); Daumé III et al. (2012); Chen et al. (2016); Kane et al. (2019); Vempala et al. (2020)). Other variants in which the number of rounds is bounded arise in space lower bounds for learning linear classifiers in streaming models (Dagan et al., 2019).

Kane et al. (2019) studied CSD in a more general communication model where the input domain $U = \mathbb{R}^d$ is the entire (infinite) space, but the parties can transmit points from their input sets for a unit cost of communication. They establish a $\tilde{O}(d^3 \log n)$ upper bound and a $\tilde{\Omega}(d + \log n)$ lower bound on the number of transmitted points/bits when the input subsets are of size n and the dimension is d. These bounds translate[9] to upper and lower bounds of $\tilde{O}(d^3 \log^2 n)$ and $\tilde{\Omega}(d + \log n)$ in the setting considered in this paper. We mention that Kane et al. (2019), as well as some of the other previous papers (*e.g.*, Balcan et al. (2012)), consider a more general set of concept classes and also study the *agnostic* (non-realizable) setting, where the labeling may not be completely consistent with any hypothesis in the class.

Recently, Vempala et al. (2020) published a thorough study of the communication complexity of various optimization problems including LP feasibility, which, as explained above, is equivalent to CSD. The main difference is that Vempala et al. (2020) do not consider arbitrary domains U, and focus on the domain U which is implied by the standard representation of d dimensional vectors. Thus, U in this case is a grid of size n, say $[n^{1/d}]^d$.

They derive a lower bound of $\Omega(\log n)$ in the randomized setting (Theorem 9.2) and of $\Omega(d \log n)$ in the deterministic setting (Theorem 3.6), as well as several upper bounds. Their best upper bound of $O(d^2 \log^2 d \log n)$ (Theorem 11.3) is based on an implementation of the *Center of Gravity* algorithm. This matches (up to an extra "$\log d$" factor) the upper bound of $O(d^2 \log d \log n)$ given in this work, but does not apply to arbitrary domains[10] U. The paper suggests a second protocol based on Clarkson (1995)*'s Algorithm* that communicates $O(d^3 \log^2 n)$ bits (matching the bound of Kane et al. (2019)) and whose analysis does extend to arbitrary domains U (Theorem 10.1).

Cutting Hyperplanes and Containers. There is an intimate relationship between containers and the notion of *cuttings* from discrete geometry. Given n hyperplanes in $\mathbb{R}^d$, an ε-cutting is a partition of $\mathbb{R}^d$ into simplices with disjoint interiors such that the interior of each simplex intersects at most $\varepsilon \cdot n$ of the hyperplanes. Cuttings provide a divide-and-conquer approach which is used to solve a variety of geometric problems (Yao and Yao, 1985; Clarkson, 1987; Chazelle, 1993; Matouvsek, 2002).

Cuttings can also be used to prove Theorem 3. The construction is based on a duality argument which views the n points in U as hyperplanes in a dual space. Then, one constructs an ε-cutting for these n hyperplanes, and given a halfspace H in the primal space, a container for it is implied

8. A set U is in convex position if $u \notin \text{conv}(U \setminus \{u\})$ for all $u \in U$.
9. The extra $\log n$ factor in the upper bound is because transmitting $u \in U$ requires $\log|U| = \log n$ bits.
10. Specifically, their analysis exploits the assumed grid structure of U: their bound on the number of iterations of the protocol uses bounds on determinants of matrices with entries from $[n^{1/d}]$. In fact, already in the one-dimensional case, if the domain $U \subseteq \mathbb{R}$ consists of n points which form a geometric progression (say $U = \{1, 2, 4, \ldots, 2^n\}$), then the Center of Gravity protocol can transmit up to $\Omega(n)$ bits, which is exponentially larger than the $O(\log n)$ optimal deterministic protocol, and double exponentially larger than the $O(\log \log n)$ optimal randomized protocol.

by the simplex in the constructed ε-cutting which contains the point dual to H. (So the number of containers is equal to the number of simplices in the ε-cutting.) The analysis is based on point-hyperplane duality which preserves the "above/below" relations (see, *e.g.*, de Berg et al. (2008)).

An advantage of the construction given here is that it can be implemented in $\mathrm{poly}(n, d)$ time (see Appendix B.4). This is in contrast to cuttings which take $\exp(d)$ time to construct. On the other hand, the cutting-based construction yields a faster, $O(d \log n)$ time, routine for finding a container given an input halfspace query. Thus, the cutting construction is appealing in contexts where many halfspace queries are made (and then the cost of preprocessing is amortized over the many queries). This is less suitable in the context of this paper since in our protocols the parties query a single halfspace per a constructed family of containers and construct many different families of containers (one family of containers per round in the protocol).

Another difference is that in the communication-complexity application considered in this paper $\varepsilon = 1/4$ is a constant and we wish to minimize the dependence on d. However, in typical applications of cuttings ε is the main parameter which is assumed to be small and d is treated as a constant.

It will be interesting to further explore the relationship between cuttings, containers, and brackets.

2. Proofs Overview

In this section we overview the proofs and highlight some of the more technical arguments. We defer the full proof to the Appendix. We begin by overviewing the proof of Theorem 3, the halfspace container theorem, which forms the crux of our communication protocols. Then, we sketch our CSD protocol in Section 2.2 and our CSD lower bound in Section 2.3, thus proving Theorem 2. Finally, in Section 2.4, we use our constructions for CSD and outline the upper and lower bound proofs for the distributed halfspace learning problem, thus proving Theorem 1.

2.1. Halfspace Containers (Theorem 3)

Let $U \subseteq \mathbb{R}^d$ be a domain with n points. We want to show that for every $\varepsilon > 0$, there is a collection $\mathcal{C}$ of (roughly) $(d/\varepsilon)^d$ sets called *containers*, such that for every halfspace H there is a container $C \in \mathcal{C}$ such that $H \subseteq C$ and $C \setminus H$ contains at most $\varepsilon \cdot n$ points from U. It will be more convenient to prove the following equivalent statement in which H and C switch roles: there is a collection $\mathcal{C}$ of (roughly) $(d/\varepsilon)^d$ sets, such that for every halfspace H there is $C \in \mathcal{C}$ such that $C \subseteq H$ and $H \setminus C$ contains at most $\varepsilon \cdot n$ points from U. Indeed, these statements are equivalent, because the complement of a halfspace is a halfspace, and so taking the complements of all sets in a family $\mathcal{C}$ with the above property yields the desired family of containers.

A natural way of constructing a set C that is contained in H but "approximates" it well, is to take $C = \cap H_i$, where the intersection is over all halfspaces H_i that are *equivalent* to H with respect to U. That is, halfspaces H_i such that $H_i \cap U = H \cap U$. However, the set $\mathcal{C}$ of all such C's is too large, as each C may be the intersection of many halfspaces H_i. We next show that we can use sets of the form $C = \cap_{i \le d+1} H'_i$ that are obtained as the intersection of only $d+1$ halfspaces H'_i, where each halfspace H'_i is "roughly equivalent" to H.

Constructing an ε-net. The first step in the construction is to pick a "small" $V \subseteq U$ which forms an ε-net to sets of the form $H_0 \setminus (\cap_{i \le d+1} H_i)$, where the H_i's are halfspaces. That is, V satisfies

that for every set B of the form $B = H_0 \setminus (\cap_{i \leq d+1} H_i)$, if B contains at least $\varepsilon \cdot n$ points from U then $B \cap V \neq \emptyset$. By standard arguments from VC theory, a random subset $V \subseteq U$ of size roughly d^2/ε will satisfy this property.

Once we have such an ε-net V, the idea is to associate with any given halfspace H a set of $d+1$ halfspaces $H_1, \ldots, H_{d+1}$ which are induced by V such that

(i) $\cap_{i \leq d+1} H_i \subseteq H$, and

(ii) $H \setminus (\cap_{i \leq d+1} H_i)$ does not contain any point from V.

Since V is an ε-net, property (ii) implies that $H \setminus (\cap_{i \leq d+1} H_i)$ contains at most $\varepsilon \cdot n$ points from U, as needed. The ε-net V will effectively represent our set U from now on, although $|V|$ is significantly smaller.

The Auxiliary Dual Polytope. To derive the halfspaces $H_1, \ldots, H_{d+1}$ which satisfy the above properties (i) and (ii), we consider the *dual space* in which each halfspace $\{x \in \mathbb{R}^d : \boldsymbol{a} \cdot x \leq b\}$ is associated with the $d+1$ dimensional vector $(\boldsymbol{a}, b) \in \mathbb{R}^{d+1}$.

Consider the set $\mathcal{P} = \mathcal{P}(H)$ of all halfspaces that are *equivalent* to H with respect to the ε-net V. That is, $\mathcal{P} \subseteq \mathbb{R}^{d+1}$ contains representations of all halfspaces H' such that $H' \cap V = H \cap V$ (we stress that such halfspaces can have a different intersection with the domain U). Note that $\mathcal{P}$ is a convex set which is defined[11] by $|V|$ linear inequalities, as each $v \in V$ corresponds to a linear inequality posing that $v \in H \iff v \in H'$. For an illustration, see Figure 3.

Now, by Carathéodory Theorem, there are $d+2$ vertices of $\mathcal{P}$ such that the vector associated with H is in their convex hull. By the definition of $\mathcal{P}$, these $d+2$ vertices correspond to halfspaces $H_i \subseteq \mathbb{R}^d$ such that $H_i \cap V = H \cap V$. We claim that these H_i's satisfy the above properties (i) and (ii). Indeed, since H is in their convex hull, it can be shown that $\cap_i H_i \subseteq H$, which amounts to (i), and since the H_i's are in $\mathcal{P}$, we have that $H_i \cap V = H \cap V$ for every i, which implies (ii).

An Inferior Bound. Let us now see how to get an inferior bound of $|V|^{O(d^2)} = (d/\varepsilon)^{O(d^2)}$ on the size of $\mathcal{C}$, by counting the number of possible $d+2$ tuples $H_1, \ldots, H_{d+1}$. How many polytopes $\mathcal{P}(H)$ are there, counting over all possible halfspaces H? The constraints defining the polytope $\mathcal{P}(H)$ are determined by the intersection $V \cap H$. Therefore, since there are $O(|V|^d)$ distinct intersections of V with halfspaces, we get that there are $O(|V|^d)$ such polytopes $\mathcal{P}(H)$. Now, given a fixed $\mathcal{P}(H)$, how many vertices does it have? $\mathcal{P}(H)$ is defined by $|V|$ constraints and therefore has at most $|V|^{d+1}$ vertices (each vertex is determined by $d+1$ constraints).

If we consider all possible $d+2$ tuples of vertices of $\mathcal{P}(H)$ as possible halfspaces $H_1, \ldots, H_{d+1}$ which have H in their convex hull, the number of such tuples is bounded by $|V|^{(d+1)(d+2)}$. The bound we get for $|\mathcal{C}|$ is $O(|V|^d) \cdot |V|^{(d+1)(d+2)} = (d/\varepsilon)^{O(d^2)}$.

To remove the extra d factor from the exponent, we exploit the *Dual Carathéodory* Theorem (Theorem 4), which asserts that if a polytope $\mathcal{Q} \subseteq \mathbb{R}^d$ is defined by n linear inequalities, then it can be covered by n^d subsimplices. In our context, $\mathcal{Q} = \mathcal{P}(H)$, the number of constraints n is $|V| \approx d^2/\varepsilon$, and the dimension is $d+1$. Thus, the theorem enables us to find a collection of just $|V|^{O(d)}$ tuples of $d+2$ vertices $H_1, \ldots, H_{d+1}$ such that every point in $\mathcal{P}(H)$, including H itself, is in the convex hull of one of these tuples.

11. In the complete proof we will define $\mathcal{P}$ with $O(d)$ more constraints in order to ensure boundedness.

Proof of Dual Carathéodory. We prove the theorem in a constructive manner, using a process from computational geometry called *Bottom Vertex Triangulation* (Clarkson, 1988; Goodman and O'Rourke, 2004). In a nutshell, given a point $a \in \mathcal{Q}$, we use the Bottom Vertex Triangulation process to encode the names of $d+1$ vertices of $\mathcal{Q}$ whose convex hull contains a. The encoding is a sequence of d linear inequalities, each being one of the n linear inequalities that define $\mathcal{Q}$. This implies that the polytope can be covered using at most n^d subsimplices, corresponding to the number of sequences of length d out of a set of size n.

In more detail, the sequence of linear inequalities is defined as follows (see Figure 4 for an illustration): given the input point a, let x_0 be the bottom-most[12] vertex of $\mathcal{Q}$, and shoot a ray starting at x_0 which passes through a until it hits a facet $\mathcal{Q}_1$ of $\mathcal{Q}$ in a point $a_1 \in \mathcal{Q}_1$. Append to the constructed sequence the number of the linear inequality which became tight as a result of hitting $\mathcal{Q}_1$. Continue recursively, applying the same process to $\mathcal{Q}_1$ (*i.e.*, shoot a ray from $\mathcal{Q}_1$'s bottom vertex x_1 which passes through a_1 until it hits a facet $\mathcal{Q}_2$ at a_2, *etc.*). We refer the reader to Figure 6 for a formal description of this process. It can be shown that a is in the convex hull of the vertices $x_0, x_1, \ldots, x_d$.

We note that an alternative (but non-constructive) proof of the Dual Carathéodory theorem can be derived from the upper bound theorem for polytopes McMullen (1970). This proof, however, will not allow for an efficient construction of containers.

2.2. The Protocol for CSD (Theorem 2)

Equipped with ε-containers for halfspaces, we turn to design a protocol for CSD. We note that even in the simple (but non trivial) case $d = 2$,[13] it was shown by Kane et al. (2019) that any "low communication" protocol must have $\tilde{\Omega}(\log n)$ rounds of communication[14]. Thus, to reach our goal of constructing a protocol with only a logarithmic dependence on n, we need the communication in every round to be independent of n (in particular, one cannot even specify one point in U).

We construct a CSD protocol with $O(d^2 \log d \log n)$ communication in two steps:

Step (i): An $O(d \log d \log n)$ protocol for PromiseCSD (Theorem 21). Let PromiseCSD_U denote the variant of CSD in which it is *promised* that the inputs X, Y satisfy:

(i) $\text{conv}(X) \cap \text{conv}(Y) = \emptyset$, or

(ii) $X \cap Y \neq \emptyset$.

In particular, the output of the protocol is not restricted in the remaining case when $X \cap Y = \emptyset$ and $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset$.

We next explain how PromiseCSD_U can be solved with $O(d \log d \log n)$ bits of communication. Observe that if $\text{conv}(X) \cap \text{conv}(Y) = \emptyset$, then X and Y can be separated by a hyperplane and one of the two halfspaces it defines contains at most $n/2$ points from U. This suggests the following approach: Alice and Bob each privately checks if their input lies in a halfspace which contains at most $n/2$ points from U. If there is no such halfspace, then, by the above reasoning, it must

12. Or any other canonical vertex.
13. The case of $d = 1$ is easy, $d = 2$ is more sophisticated, and $d = 3$ seems to require a general approach.
14. We mention that Kane et al. (2019) also present a natural protocol based on boosting/multiplicative-weights update rule with $\Theta(\log^2 n)$ communication complexity. Such quadratic dependence is also exhibited by other approaches (*e.g.*, the protocol by Vempala et al. (2020) which is based on Clarkson's algorithm). These protocols consist of $\Theta(\log n)$ rounds and $\tilde{\Omega}(\log n)$ bits are communicated in every round.

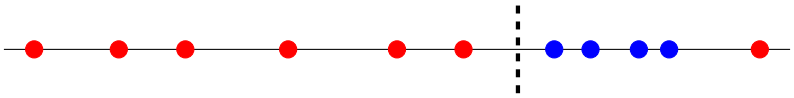

Figure 2: Example of a case where the algorithm for PromiseCSD does not extend to the general case of CSD: the convex hull of the red points intersects the convex hull of the blue points. Since the halfspace on the right of the dashed hyperplane contains all the blue points and less than half of the total, the parties may decide to remove all the points to the left of the hyperplane. However, once these points are removed, the convex hulls of the remaining red and blue points are disjoint.

be the case that $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$ and the protocol terminates. Else, since there are $n^{O(d)}$ halfspaces up to equivalences[15], they can agree on such a halfspace using $O(d \log n)$ bits and remove all domain points outside this halfspace.

Alice and Bob can iteratively proceed in this manner and in every step remove at least half of the (remaining) points *while maintaining that all points in $X \cap Y \subseteq U$ are never being removed*. The implied protocol consists of $O(\log n)$ rounds, each communicating $O(d \log n)$ bits, for a total of $O(d \log^2 n)$ communicated bits, which is $\log n$ factor away from the stated bound.

Our final protocol uses a similar recursive approach, but *transmits only $O(d \log d)$ bits in each round*. This is achieved by using halfspace containers (Theorem 3). Specifically, instead of finding a halfspace which contains the entire input of one of the players, they find an ε-container for this halfspace with $\varepsilon = 1/4$. This allows to reduce the domain size by a factor of $1/2 + 1/4 = 3/4$ in each round, and, by Theorem 3, requires only $O(d \log d)$ bits per round. Note that the use of containers, as opposed to an approximating[16] set from an ε-cover, is crucial here.

One may be tempted to try a similar approach for the non-promise variant. However, note that points in $\mathrm{conv}(X) \cap \mathrm{conv}(Y)$ that are not in $X \cap Y$ may be removed by the protocol. Indeed, Figure 2 depicts a situation where the protocol starts with sets X, Y with $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$ and removes some of the points in U to obtain a domain U' in which $\mathrm{conv}(X \cap U') \cap \mathrm{conv}(Y \cap U') = \emptyset$. This shows that without the promise, this approach fails.

Step (ii): Reducing CSD to PromiseCSD (Theorem 24). Clearly, $\mathsf{PromiseCSD}_U$ can only be easier to decide than CSD_U. In the opposite direction, it turns out that it is not much harder. Specifically, one can reduce CSD_U to $\mathsf{PromiseCSD}_V$, where V is a domain obtained from U by adding carefully chosen points to ensure that for every $X, Y \subseteq U$, if $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$ then $\mathrm{conv}(X) \cap \mathrm{conv}(Y)$ contains a point in V. Now, if $X, Y \subseteq U$ are the inputs for CSD_U, then $\mathrm{conv}(X) \cap V, \mathrm{conv}(Y) \cap V \subseteq V$ are valid inputs for $\mathsf{PromiseCSD}_V$.

The naive attempt of obtaining V from U by going over all pairs of sets $X, Y \subseteq U$ with $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$ and adding an auxiliary point from $\mathrm{conv}(X) \cap \mathrm{conv}(Y)$ to V results in the addition of too many points. Instead, we prove a "*symmetric variant of Carathéodory's Theorem*" (Theorem 23), which asserts that if $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$ then there are $X' \subseteq X$ and $Y' \subseteq Y$ such that $|X'| + |Y'| \leq d + 2$ and $\mathrm{conv}(X') \cap \mathrm{conv}(Y') \neq \emptyset$. It is therefore sufficient to only go over pairs of sets $X', Y' \subseteq U$ such that $|X'| + |Y'| \leq d + 2$ and add to V an auxiliary point in $\mathrm{conv}(X') \cap \mathrm{conv}(Y')$. Note that by doing so, we automatically cover *all possible inputs* X and

15. Recall that two halfspaces are equivalent if they have the same intersection with U.
16. I.e. a set with at most εn points from U in its symmetric difference with the halfspace.

Y to the parties *without* knowing what they happen to be. Moreover, this only requires the addition of $(2n)^{d+2}$ points and implies the stated upper bound of $O(d^2 \log d \log n)$ for CSD_U (as n is now $n + (2n)^{d+2}$ and the dimension d remains the same). We remark that by the classic Carethéodory Theorem, it suffices to go over all pairs of intersecting simplices. This yields a bound of $(2n)^{2d+2}$ on the number additional points which is off by a factor of 2. While the latter inferior bound is sufficient for the reduction, we include the tighter bound and the symmetric variant of Carathéodory as they may be of independent interest.

2.3. Lower Bound for CSD (Theorem 2)

We prove an $\Omega(d \log(n/d))$ lower bound for PromiseCSD, which clearly also holds for CSD.

Embedding DISJ in PromiseCSD with $d = 2$. The first part in the lower bound is a reduction from the Set Disjointness problem (denoted DISJ) on $\log m$ bits to planar PromiseCSD with m points. This is achieved by fixing m points in a convex position, say on the unit circle, and identifying each $\log m$ bit-string $\boldsymbol{z}$ with one of the m points, call it $v_{\boldsymbol{z}}$. Next, given inputs $\boldsymbol{x}, \boldsymbol{y} \in \{0,1\}^{\log m}$, Alice maps her input to the singleton set $\{v_{\boldsymbol{x}}\}$, whereas Bob maps his input to the set of all $v_{\boldsymbol{z}}$ such that $\boldsymbol{z} \cap \boldsymbol{y} \neq \emptyset$. Note that Alice's point is in Bob's set if and only if $\boldsymbol{x} \cap \boldsymbol{y} \neq \emptyset$. Moreover, since the m points are in convex position, Alice's point is in Bob's set if and only if the point cannot be separated from the set by a hyperplane; *i.e.*, if and only if their convex hulls intersect. See Figure 5 for an illustration of this construction.

Dimension lifting via direct sum. The second part of the lower bound is to lift the planar construction to higher dimensions using a "*direct sum argument*". Let $d \in \mathbb{N}$ and set $m = 3n/d$. Observe that solving $d/3$ copies of DISJ on $\log m$ coordinates (deciding if *all* pairs of inputs are disjoint or whether there *exists* an intersecting pair) is as hard as solving DISJ on $d \log m/3$ coordinates, which is known to admit a randomized communication complexity lower bound of $\Omega(d \log m) = \Omega(d \log(n/d))$. Thus, solving $d/3$ copies of our planer PromiseCSD with m points also requires at least $\Omega(d \log(n/d))$ communication bits.

We next claim that solving PromiseCSD with dimension d and $dm/3$ points is at least as hard as solving $d/3$ copies of the planar PromiseCSD with m points: consider the mapping $g_i : \mathbb{R}^2 \to \mathbb{R}^d$ that for every $i \in [d/3]$ takes a point $v = (v_1, v_2) \in \mathbb{R}^2$ and outputs the vector $(0, 0, \cdots, 0, v_1, v_2, 1, 0, 0, \cdots, 0)$ in $\mathbb{R}^d$, where v_1 is in position $3(i-1)+1$. Given $d/3$ input pairs $\{(X_i, Y_i)\}_{i \in [d]}$ to the planar PromiseCSD, we construct inputs X and Y for PromiseCSD in d dimensions: let $X = \{g_i(x_i)\}_{i \in [d/3], x_i \in X_i}$ and $Y = \{g_i(y_i)\}_{i \in [d/3], y_i \in Y_i}$. It is not hard to see that since we embed the inputs to different copies of the planar problem in different coordinates of the d-dimensional problem, it holds that

$$\Big(\forall i : \mathrm{conv}(X_i) \cap \mathrm{conv}(Y_i) = \emptyset\Big) \iff \mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset.$$

2.4. Learning Halfspaces (Theorem 1)

Protocol. Observe that our $O(d \log d \log n)$ protocol for PromiseCSD has the property that when the convex hulls of X, Y are disjoint, a function $f : U \to \{\pm 1\}$ certifying this fact can be derived from its execution (*i.e.*, $f(u) = +1$ for every $u \in X$ and $f(u) = -1$ for every $u \in Y$). Therefore, as explained in Section 1.2, this protocol immediately yields a learning protocol in the case when Alice only has negative examples and Bob only has positive examples. The case where both Alice

and Bob may have mixed examples is more subtle, but the protocol and analysis remain rather simple (see Appendix E.1).

Lower bound. We claim that the $\Omega(d \log(n/d))$ lower bound for PromiseCSD also applies to our learning problem. We will show that every protocol for the learning problem implies a protocol for PromiseCSD: let X, Y be the inputs for PromiseCSD. Run the learning protocol with inputs in the set X labeled $+1$ and inputs in the set Y labeled -1 to obtain a separator f. Have Alice check that f is positive on X, and have Bob check that f is negative on Y. If this is the case, f is a witness to the fact that $X \cap Y = \emptyset$. Otherwise, since we assume that the learning protocol is always correct, it must be the case that the convex hulls of X and Y cannot be separated. Observe that the argument crucially exploits the fact that X and Y are inputs to PromiseCSD (rather than to CSD), which enables an *improper* learner to be applied.

Acknowledgments

We thank Bernard Chazelle and an anonymous reviewer of a previous version of this manuscript for pointing out the connection between containers and hyperplane cuttings. We thank Noga Alon, Sepehr Assadi, Steve Hanneke, Shachar Lovett and Nikita Zhivotovskii for providing insightful comments.

Shay Moran is a Robert J. Shillman Fellow and is supported by the ISF, grant no. 1225/20, by an Azrieli Faculty Fellowship, and by BSF grant 2018385.

References

Terrence M. Adams and Andrew B. Nobel. Uniform convergence of vapnik–chervonenkis classes under ergodic sampling. *Ann. Probab.*, 38(4):1345–1367, 07 2010.

Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In *Conference on Learning Theory (COLT)*, pages 26.1–26.22, 2012.

Jozsef Balogh, Robert Morris, and Wojciech Samotij. The method of hypergraph containers, 2018.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *J. Assoc. Comput. Mach.*, 36(4):929–965, 1989.

C. Carathéodory. Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. *Math. Ann.*, 64(1):95–115, 1907. ISSN 0025-5831.

Edward Y. Chang, Kaihua Zhu, Hao Wang, Hongjie Bai, Jian Li, Zhihuan Qiu, and Hang Cui. Psvm: Parallelizing support vector machines on distributed computers. In *Neural Information Processing Systems (NIPS)*, pages 257–264. Curran Associates Inc., 2007.

Bernard Chazelle. Cutting hyperplanes for divide-and-conquer. *Discret. Comput. Geom.*, 9:145–158, 1993.

Shang-Tse Chen, Maria-Florina Balcan, and Duen Horng Chau. Communication efficient distributed agnostic boosting. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 1299–1307, 2016.

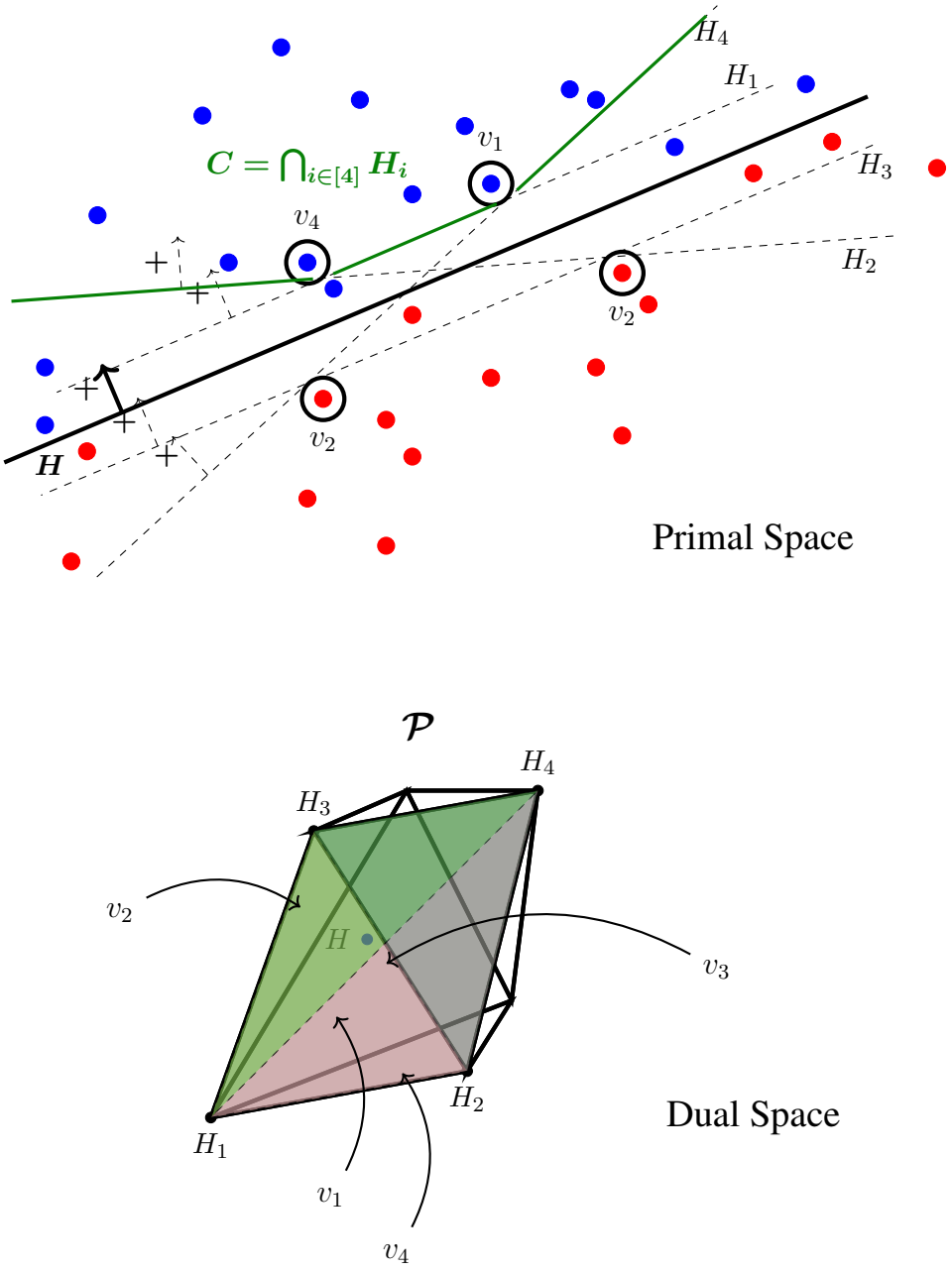

Figure 3: The auxiliary dual polytope: halfspaces in the primal space (top figure) are represented by points in the dual space (bottom figure), and points in the primal space correspond to halfspaces in the dual space. The circled points in the primal space figure denote the points in the ε-net V; these points define the facets of the auxiliary polytope $\mathcal{P}$, which is (a dual representation of) the set of halfspaces that induce the same partition of V as H.

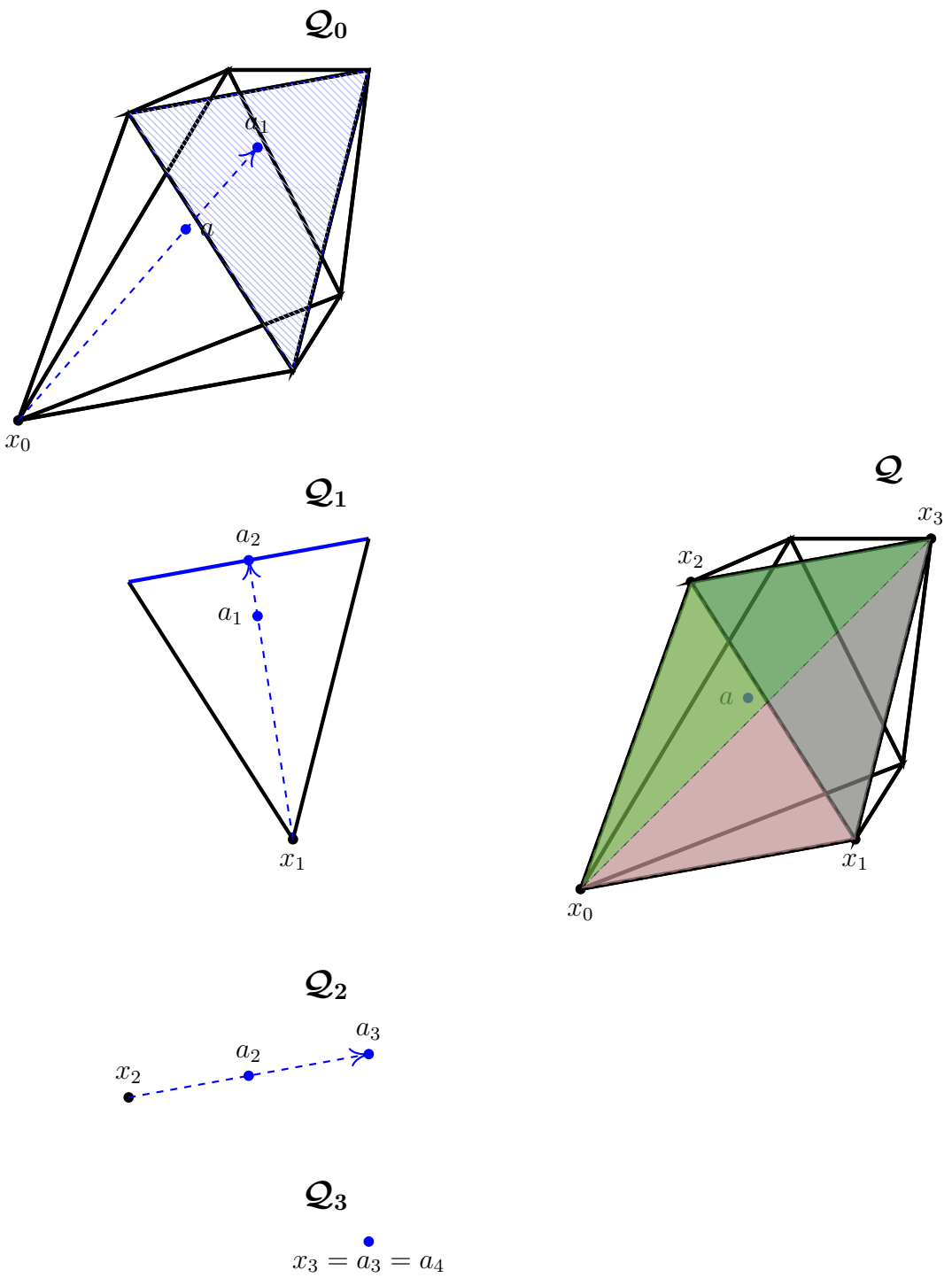

Figure 4: An illustration of Bottom Vertex Triangulation process applied to the polytope $\mathcal{Q}$ and the point $a \in \mathcal{Q}$. The process starts by shooting a ray from the bottom vertex, x_0, to a. The ray is extended until it hits the facets $\mathcal{Q}_1$ of $\mathcal{Q}$ in the point $a_1 \in \mathcal{Q}_1$ (see top picture). The process is then repeated with the facet $\mathcal{Q}_1$ as a polytope with (at least) one fewer dimension and the point a_1 (see the second from the top picture), *etc.*. The point a is in the convex hull of the vertices $x_0, x_1, \ldots, x_d$ (see picture on the right).

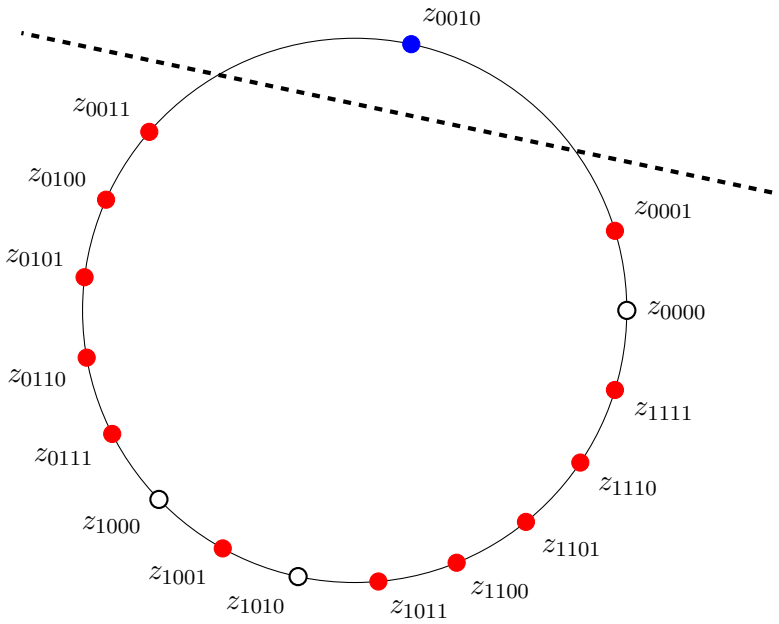

Figure 5: The reduction from DISJ on $\log m = 4$ bits to PromiseCSD on $m = 16$ points. Alice's input for DISJ is 0010 and Bob's input is 0101. The domain U of PromiseCSD has 16 equally spaced points on the unit circle (points are not evenly spaced in the above figure to emphasize the dashed separating hyperplane). Alice's input is mapped to the singelton containing the blue point. Bob's input is mapped to the set of red points.

Kenneth L. Clarkson. New applications of random sampling in computational geometry. *Discret. Comput. Geom.*, 2:195–222, 1987.

Kenneth L. Clarkson. A randomized algorithm for closest-point queries. *SIAM Journal on Computing*, 17(4):830–847, 1988.

Kenneth L. Clarkson. Las vegas algorithms for linear and integer programming when the dimension is small. *J. ACM*, 42(2):488–499, 1995.

Yuval Dagan, Gil Kur, and Ohad Shamir. Space lower bounds for linear prediction in the streaming model. In *Conference on Learning Theory (COLT)*, volume 99, pages 929–954. PMLR, 2019.

Hal Daumé III, Jeff M. Phillips, Avishek Saha, and Suresh Venkatasubramanian. Efficient protocols for distributed classification and optimization. In *Algorithmic Learning Theory (ALT)*, pages 154–168, 2012.

Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008.

R. M. Dudley. *Uniform Central Limit Theorems*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 1999.

Uriel Feige, David Peleg, Prabhakar Raghavan, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Pedro A. Forero, Alfonso Cano, and Georgios B. Giannakis. Consensus-based distributed support vector machines. *J. Mach. Learn. Res.*, 11:1663–1707, 2010.

Bernd Gärtner and Emo Welzl. Vapnik-chervonenkis dimension and (pseudo-)hyperplane arrangements. *Discrete & Computational Geometry*, 12:399–432, 1994.

Jacob E. Goodman and Joseph O'Rourke, editors. *Handbook of Discrete and Computational Geometry, Second Edition*. Chapman and Hall/CRC, 2004. ISBN 978-1-58488-301-2.

David Haussler. Sphere packing numbers for subsets of the boolean n-cube with bounded vapnik-chervonenkis dimension. *Journal of Combinatorial Theory, Series A*, 69(2):217–232, 1995.

David Haussler and Emo Welzl. Epsilon-nets and simplex range queries. In *CG*, pages 61–71, 1986.

Jian-Pei Zhang, Zhong-Wei Li, and Jing Yang. A parallel svm training algorithm on large-scale classification problems. In *International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 3, pages 1637–1641 Vol. 3, Aug 2005.

Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.

Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory (COLT)*, volume 99 of *Proceedings of Machine Learning Research*, pages 1903–1943. PMLR, 2019.

Jakub Konečný, H. Brendan McMahan, Felix X. Yu, Peter Richtarik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. In *NIPS Workshop on Private Multi-Party Machine Learning*, 2016.

Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

László Lovăsz and Michael Saks. Communication complexity and combinatorial lattice theory. *Journal of Computer and System Sciences*, 47(2):322 – 349, 1993.

Jivr'i Matouvsek. *Lectures on discrete geometry*, volume 212 of *Graduate texts in mathematics*. Springer, 2002.

Ryan McDonald, Keith Hall, and Gideon Mann. Distributed training strategies for the structured perceptron. In *Human Language Technologies (HLT)*, pages 456–464. Association for Computational Linguistics, 2010.

Peter McMullen. The maximum numbers of faces of a convex polytope. *Mathematika*, 17(2): 179–184, 1970.

A. Navia-Vazquez, D. Gutierrez-Gonzalez, E. Parrado-Hernandez, and J. J. Navarro-Abellan. Distributed support vector machines. *Trans. Neur. Netw.*, 17(4):1091–1097, 2006.

D. Pollard. *Convergence of Stochastic Processes*. Clinical Perspectives in Obstetrics and Gynecology. Springer New York, 1984.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.

A. van der Vaart, AW van der Vaart, A.W. van der Vaart, and J. Wellner. *Weak Convergence and Empirical Processes: With Applications to Statistics*. Springer Series in Statistics. Springer, 1996.

Ramon van Handel. The universal glivenko–cantelli property. *Probability Theory and Related Fields*, 155(3):911–934, 2013.

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

Santosh S. Vempala, Ruosong Wang, and David P. Woodruff. The communication complexity of optimization. In Shuchi Chawla, editor, *Symposium on Discrete Algorithms (SODA)*, pages 1733–1752, 2020.

Emanuele Viola. The communication complexity of addition. In *Symposium on Discrete Algorithms (SODA)*, pages 632–651, 2013.

A C Yao and F F Yao. A general approach to d-dimensional geometric queries. In *Symposium on Theory of Computing (STOC)*, pages 163–168. Association for Computing Machinery, 1985.

Andrew Chi-Chih Yao. Some complexity questions related to distributive computing (preliminary report). In *Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

Appendix A. Preliminaries

A.1. Notation

We use capital letters to denotes sets (*e.g.*, X, Y, U). We denote by calligraphic capital letters families of sets (*e.g.*, $\mathcal{C}, \mathcal{F}$). We use bold small letters to denote vectors (*e.g.*, $\boldsymbol{x}, \boldsymbol{y}$). We sometimes write $\boldsymbol{x}^{(k)}$ to stress that the vector $\boldsymbol{x}$ consists of k coordinates, numbered 1 to k. If $\boldsymbol{x}$ is a vector, we denote by x_i the i^{th} coordinate in $\boldsymbol{x}$.

For $d \in \mathbb{N}$, let HS_d denote the family of all halfspaces in $\mathbb{R}^d$. For $U \subseteq \mathbb{R}^d$ let $\mathsf{HS}(U) = \{H \cap U : H \in \mathsf{HS}_d\}$ denote the family of all halfspaces restricted to U.

A.2. Problems Definitions

A.2.1. CONVEX SET DISJOINTNESS

Definition 5 (CSD_U) *Let $U \subseteq \mathbb{R}^d$ be a finite set. The convex set disjointness function $\mathsf{CSD}_U(X, Y) : 2^U \times 2^U \to \{0, 1\}$ is defined as:*

$$\mathsf{CSD}_U(X, Y) = \begin{cases} 0 & , \text{conv}(X) \cap \text{conv}(Y) \neq \phi \\ 1 & , \textit{otherwise}. \end{cases}$$

Definition 6 ($\mathsf{PromiseCSD}_U$) *Let $U \subseteq \mathbb{R}^d$ be finite. The partial function $\mathsf{PromiseCSD}_U(X,Y) : 2^U \times 2^U \to \{0,1\}$ is defined as:*

$$\mathsf{PromiseCSD}_U(X,Y) = \begin{cases} 0 & , X \cap Y \neq \phi \\ 1 & , \mathrm{conv}(X) \cap \mathrm{conv}(Y) = \phi. \end{cases}$$

A.2.2. LEARNING HALFSPACES

Let $U \subseteq \mathbb{R}^d$ be a set. An *example* is a pair of the form $(u,b) \in U \times \{\pm 1\}$. An example (u,b) is called *positive* if $b = +1$ and *negative* if $b = -1$. A set of examples $S \subseteq U \times \{\pm 1\}$ is called a *sample*. The problem of *two-party distributed learning of halfspaces over U* refers to the following search problem: Alice's and Bob's inputs are samples $S_a, S_b \subseteq U \times \{\pm 1\}$ such that there exists a halfspace which contains all the positive examples in $S_a \cup S_b$ and does not contain any negative examples in $S_a \cup S_b$. The parties' goal is to output a function $f : U \to \{\pm 1\}$ such that $f(u) = b$ for every example $(u,b) \in S_a \cup S_b$. If the protocol always outputs f such that f is an indicator of a halfspace, then the protocol is called a *proper* learning protocol. Otherwise it is called an *improper* learning protocol.

A.3. VC Theory

We will use two basic results from VC theory. Recall that the *VC dimension* of a family $\mathcal{F} \subseteq 2^X$ is the size of the largest $Y \subseteq X$ such that $\{F \cap Y : F \in \mathcal{F}\} = 2^Y$. An *$\varepsilon$-net* for $\mathcal{F}$ is a set $N \subseteq X$ such that $N \cap F \neq \emptyset$ for all $F \in \mathcal{F}$ with $|F| \geq \epsilon |X|$. A useful property of families with small VC-dimension is that they have small ε-nets.

Theorem 7 (ε-net theorem) *(Haussler and Welzl, 1986; Vapnik and Chervonenkis, 2015) Let $\mathcal{F} \subseteq 2^X$ be a family with VC dimension d and let $\varepsilon > 0$. Then, there exists an ε-net for $\mathcal{F}$ of size $O\left(\frac{d \log(1/\varepsilon)}{\varepsilon}\right)$.*

We will also use the following lemma which bounds the growth in the VC dimension under set operations:

Lemma 8 (VC of k-fold compositions) *(Blumer et al., 1989) Let $\mathcal{F}_1 \ldots \mathcal{F}_k$ be a sequence of families with VC dimension at most d, and let $\star_1 \ldots \star_{k-1}$ be a sequence of binary operations on sets (e.g. $\star_1 = \cap, \star_2 = \cup, \star_3 = \Delta$, and so forth). Set*

$$\mathcal{F}^{\star k} = \left\{ F_1 \star_1 (F_2 \star_2 \ldots (F_{k-1} \star_{k-1} F_k)) : F_i \in \mathcal{F}_i \right\}.$$

Then, the VC dimension of $\mathcal{F}^{\star k}$ is at most $O(kd \log d)$.

This Lemma allows to use the VC dimension of $\mathcal{F}$ to bound the VC dimension of more complex families, *e.g.*,

$$\left\{ \left(F_1 \setminus (\cap_{i=2}^{100} F_i)\right) \cup F_{101} : F_i \in \mathcal{F} \right\}.$$

A.4. Communication Complexity

We use standard notation and terminology from Yao's communication complexity model (Yao, 1979), and refer the reader to (Kushilevitz and Nisan, 1997) for a textbook introduction. For a (possibly partial) function f, we denote by $D(f)$ the deterministic communication complexity of f, and by $R_\epsilon(f)$ the randomized communication complexity of f with error probability $\epsilon \geq 0$. We set $R(f) = R_{1/3}(f)$.

A.4.1. Communication Problems

Definition 9 (DISJ_n) *The disjointness function* $\mathsf{DISJ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *is defined as:*

$$\mathsf{DISJ}_n(\boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 0 & , \exists i \colon x_i = y_i = 1 \\ 1 & , \textit{otherwise}. \end{cases}$$

Definition 10 (AND_k) *For a function* $f : \mathcal{X} \times \mathcal{Y} \to \{0,1\}$*, the function* $\mathsf{AND}_k \circ f : \mathcal{X}^k \times \mathcal{Y}^k \to \{0,1\}$ *is defined as:*

$$\mathsf{AND}_k \circ f(\boldsymbol{x}^{(k)}, \boldsymbol{y}^{(k)}) = \bigwedge_{i=1}^{k} f(x_i, y_i).$$

A.4.2. Reductions

All functions in this section may be partial. We denote by $\mathrm{dom}(f)$ the domain of the (possibly partial) function f.

Definition 11 (Reduction) *We say a function* $f_1 : \mathcal{X}_1 \times \mathcal{Y}_1 \to \{0,1\}$ *reduces to a function* $f_2 : \mathcal{X}_2 \times \mathcal{Y}_2 \to \{0,1\}$ *(denoted* $f_1 \preceq f_2$*) if there exist functions* $\alpha : \mathcal{X}_1 \to \mathcal{X}_2$ *and* $\beta : \mathcal{Y}_1 \to \mathcal{Y}_2$ *such that for all* $(x, y) \in \mathrm{dom}(f_1)$*:*

$$f_1(x, y) = f_2(\alpha(x), \beta(y)).$$

We use the phrase "reduction functions" to refer to the functions α, β*. If* f_2 *is a partial function, we further require that* $(\alpha(x), \beta(y)) \in \mathrm{dom}(f_2)$*.*

The following results are straightforward:

Observation 12 *For functions* f_1*,* f_2*, and* f_3*, we have* $(f_1 \preceq f_2) \wedge (f_2 \preceq f_3) \implies f_1 \preceq f_3$*.*

Observation 13 *For functions* f_1*,* f_2*, we have* $f_1 \preceq f_2 \implies R_\epsilon(f_1) \leq R_\epsilon(f_2)$ *for all* $\epsilon \geq 0$*.*

Lemma 14 *For functions* f_1, f_2*, if* $f_1 \preceq f_2$*, then for any* $k > 0$*, we have* $\mathsf{AND}_k \circ f_1 \preceq \mathsf{AND}_k \circ f_2$*.*

Proof Since $f_1 \preceq f_2$, we know that there exists reduction functions α, β such that for all $(x, y) \in \mathrm{dom}(f_1)$:

$$f_1(x, y) = f_2(\alpha(x), \beta(y)).$$

Define:

$$\alpha^*(\boldsymbol{x}^{(k)}) = (\alpha(x_1), \alpha(x_2), \cdots, \alpha(x_k)),$$

$$\beta^*(\boldsymbol{y}^{(k)}) = (\beta(y_1), \beta(y_2), \cdots, \beta(y_k)).$$

Note that

$$\begin{aligned}\mathsf{AND}_k \circ f_1(\boldsymbol{x}, \boldsymbol{y}) &= \bigwedge_{i\in[k]} f_1(x_i, y_i) \\ &= \bigwedge_{i\in[k]} f_2(\alpha(x_i), \beta(y_i)) = \mathsf{AND}_k \circ f_2(\alpha^*(\boldsymbol{x}), \beta^*(\boldsymbol{y})).\end{aligned}$$

■

Appendix B. A Container Lemma for Halfspaces

The protocols in this paper hinge on *ε-containers*[17] (defined below). This is a variant of the notion of *ε-covers*, which we recall next: an *ε-cover* for a family $\mathcal{F} \subseteq 2^X$ is a family $\mathcal{C} \subseteq 2^X$ such that for every $F \in \mathcal{F}$ there is $C \in \mathcal{C}$ such that the symmetric difference[18] between C and F is of size at most $\varepsilon|X|$. In other words, the hamming balls of radius $\varepsilon|X|$ around $\mathcal{C}$ cover $\mathcal{F}$. Note that this is a special instance of the notion of ε-cover in metric spaces. In the case of containers, we also require that $F \subseteq C$:

Definition 15 (Containers) *Let X be a finite set and let $\mathcal{F} \subseteq 2^X$ be a family of subsets. A family $\mathcal{C} \subseteq 2^X$ is a family of ε-containers for $\mathcal{F}$ if*

$$(\forall F \in \mathcal{F})(\exists C \in \mathcal{C}) \ : \ F \subseteq C \ \text{ and } \ |C \setminus F| \leq \varepsilon|X|.$$

Note that every set of ε-containers is in particular an ε-cover (but not vice versa).

A Container Lemma for Halfspaces. A classical result by Haussler implies that $\mathsf{HS}(U)$, the family of all halfspaces restricted to U, has an ε-cover of size roughly $(1/\varepsilon)^d$ (Haussler, 1995). A remarkable property of this ε-cover is that its size depends only on ε and d; in particular, it does not depend on $|U|$.

Theorem 16 below (equivalent to Theorem 3), which one of our main technical contributions, establishes a similar statement for ε-containers.

Theorem 16 (halfspace containers) *Let $U \subseteq \mathbb{R}^d$ and $\varepsilon > 0$. Then, there exists a set of ε-containers for $\mathsf{HS}(U)$ of size $(d/\varepsilon)^{O(d)}$.*

Before proving Theorem 16, we first discuss how it relates to Haussler's result Haussler (1995).

B.1. Comparison with Haussler's Packing Lemma

As mentioned above, Theorem 16 is closely related to a result by Haussler (1995), which asserts that every family $\mathcal{F} \subseteq 2^X$ with VC dimension d (*e.g.*, $d-1$ dimensional halfspaces) has an ε-cover of size roughly $(1/\varepsilon)^d$. We note that unlike Haussler's result, Theorem 16 does not extend to arbitrary VC classes (below is a counterexample with VC dimension 2). This is also reflected in our proof of Theorem 16 which exploits the dual variant of Carathéodory's Theorem (Theorem 4), which does not extend to arbitrary VC classes.

17. This notation is inspired by a similar notion that arises in Graph Theory (see, *e.g.*, Balogh et al. (2018) and references within).
18. Equivalently, the hamming distance between the indicator vectors.

Example. Consider a projective plane P of order n with $N = n^2 + n + 1$ points and N lines. In particular the following holds: (i) for every pair of points there is a unique line containing them, (ii) every pair of lines intersects in one point, (iii) every line contains exactly n points, (iv) and every point is contained in exactly n lines.

Let $\mathcal{F}$ be the family

$$\{L : L \text{ is a line in } P\}.$$

One can verify that $\mathcal{F}$ has VC dimension 2. Set $\varepsilon = 1/4$. Since each line contains $n = O(\sqrt{N})$ points, then for a sufficiently large N, the existence of a set of ε-containers for $\mathcal{F}$ of size t amounts to the following statement:

There exist t sets of size at most $N/3$ each, such that every line in P is contained in at least one of them.

Therefore, by averaging, one of these t sets contains at least N/t lines $L_1, L_2, \ldots L_{N/t}$. Denote such a set by C. Assume towards contradiction that t depends only on $\varepsilon = 1/4$ and $d = 2$, and in particular that $t \leq N/n = \theta(n)$. Now, since every two lines intersect in one point it follows that

$$\begin{aligned} |\cup_{i=1}^{N/t} L_i| &\geq n + (n-1) + \ldots + 1 && \text{(because } |L_i \setminus \cup_{j<i} L_j| \geq n - (i-1)) \\ &\geq n^2/2, \end{aligned}$$

where in the first inequality we used that $N/t \geq n$. Thus, since C contains this union:

$$n^2/2 \leq |C| \leq N/3 = (n^2 + n + 1)/3,$$

which is a contradiction when n is sufficiently large. We note that this example was used in the probability literature in the context of uniform laws of large number (which is intimately related to brackets) (van Handel, 2013).

B.2. Proof of Halfspace Containers Theorem (Theorem 16)

The superset $\mathcal{C}'$. Let $\mathcal{C}' = \{U \setminus (\cap_{i=1}^{d} H_i) : H_i \in \mathsf{HS}_d\}$. It is easy to see that $\mathcal{C}' \supseteq \mathsf{HS}(U)$, and therefore it is an ε-cover for $\mathsf{HS}(U)$, for every ε. However $\mathcal{C}'$ is a much larger set than we can afford. The final cover $\mathcal{C}$ will be a carefully selected subfamily of $\mathcal{C}'$.

To select the subset $\mathcal{C} \subseteq \mathcal{C}'$, we use the following observation that provides a criteria to certify that $\mathcal{C}$ is a set of ε-containers for HS_d: it suffices to show that for every $H \in \mathsf{HS}_d$ there is $C \in \mathcal{C}$ such that C is an ε-container for F. Here, for any $C, F \subseteq 2^X$, we say that C is an ε-container for F if $F \subseteq C$, and $|C \setminus F| \leq \varepsilon |X|$.

Observation 17 *Let $\mathcal{F}, \mathcal{C} \subseteq 2^X$. Let V be an ε-net for $\{C' \setminus F' : C' \in \mathcal{C}, F' \in \mathcal{F}\}$. Let $C \in \mathcal{C}$ and $F \in \mathcal{F}$ such that*

1. *$F \subseteq C$ and*
2. *$C \cap V = F \cap V$.*

Then, C is an ε-container for F. (Namely, $F \subseteq C$, and $|C \setminus F| \leq \varepsilon |X|$).

Proof Given items 1 in the observation, it remains to show that $|C \setminus F| \leq \varepsilon |X|$. This follows by the second item, which implies that $\emptyset = (C \cap V) \setminus (F \cap V) = (C \setminus F) \cap V$, and since V is an ε-net for $\{C' \setminus F' : C' \in \mathcal{C}, F' \in \mathcal{F}\}$. We get that $|C \setminus F| \leq \varepsilon |X|$, as required. ■

The ε-net V. Our selection of $\mathcal{C} \subseteq \mathcal{C}'$ hinges on Observation 17, and therefore we use an ε-net V for the family $\mathcal{C}'' = \{C' \setminus H' : C' \in \mathcal{C}', H' \in \mathsf{HS}_d\}$ of size

$$|V| = O\left(\frac{d^2 \log d \log(1/\varepsilon)}{\varepsilon}\right).$$

(Note, in particular, that V is an ϵ-net for every subfamily of $\mathcal{C}''$). The bound on $|V|$ follows from Theorem 7 because the VC dimension of $\mathcal{C}''$ is $O(d^2 \log d)$. This bound on the VC dimension of $\mathcal{C}''$ follows because the VC dimension of HS_d is $d+1$, thus, due to Lemma 8, the VC dimension of $\mathcal{C}'$ and $\mathcal{C}''$ is $O(d^2 \log d)$.

The family of containers $\mathcal{C}$. Next we construct $\mathcal{C}$. The construction is based on an encoding-decoding scheme: given a halfspace $H \in \mathsf{HS}(U)$, the scheme encodes H into a bit-string $\boldsymbol{b} = \boldsymbol{b}(H)$ of length $t = O(d \log|V|)$. The bit-string $\boldsymbol{b}$ is then decoded to a set $C = C(\boldsymbol{b}) \in \mathcal{C}'$ satisfying the two items in Observation 17 with respect to V – and therefore C is an ε-container of H. The upper bound on the length t of $\boldsymbol{b}$ implies that the collection $\{C(\boldsymbol{b}) : \boldsymbol{b} \in \{0,1\}^t\} \subseteq \mathcal{C}'$ is a set of ε-containers for $\mathsf{HS}(U)$ of size $2^t = 2^{O(d \log|V|)} = |V|^{O(d)} = (d/\varepsilon)^{O(d)}$.

Let $H \in \mathsf{HS}(U)$. Let $a \in \mathbb{R}^d$, $\|a\|_\infty \leq 1$ and $b \in \mathbb{R}$, $|b| \leq 1$ be such that

$$H = \{u \in U : \langle a, u\rangle < b\}.$$

Since U is finite, we may assume without loss of generality that there exists a universal[19] small constant $\varepsilon > 0$ such that $\langle a, u\rangle < b - \varepsilon$ for every $u \in H$ and $\langle a, u\rangle > b + \varepsilon$ for every $u \in U \setminus H$.

The rest of the proof is devoted to constructing an ε-container C for H by first constructing $\boldsymbol{b} = \boldsymbol{b}(H)$ and then $C = C(\boldsymbol{b})$.

The auxiliary polytope $\mathcal{P}$. The definition of $\boldsymbol{b}(H)$ uses a polytope $\mathcal{P}$ that we define next. Recall that $V \subseteq U$ is an ε-net for $\mathcal{C}'' = \{C' \setminus H' : C' \in \mathcal{C}', H' \in \mathsf{HS}_d\}$. Let $V^- = V \cap H = \{v \in V : \langle a, v\rangle < b\}$, $V^+ = V \setminus H = \{v \in V : \langle a, v\rangle \geq b\}$. Define $\mathcal{P} \subseteq \mathbb{R}^{d+1}$:

$$\mathcal{P} = \left\{(\alpha, \beta) \in \mathbb{R}^d \times \mathbb{R} \;\middle|\; \big(\|(\alpha,\beta)\|_\infty \leq 1\big) \wedge \big(\forall v \in V^+ : \langle \alpha, v\rangle \geq \beta + \varepsilon\big) \wedge \big(\forall v \in V^- : \langle \alpha, v\rangle \leq \beta - \varepsilon\big)\right\}.$$

Observe that $\mathcal{P}$ contains a representation (α, β) for each halfspace $H' = \{u \in U : \langle \alpha, u\rangle < \beta\}$ such that $H' \cap V = H \cap V = V^-$, and only such representations. The constraint $\|(\alpha,\beta)\|_\infty \leq 1$ ensures that $\mathcal{P} \subseteq \mathbb{R}^{d+1}$ is bounded, a property which will enable us to apply Theorem 4 to $\mathcal{P}$. Note that $\mathcal{P}$ is a closed polytope which is defined by $|V| + 2(d+1)$ linear inequalities (the constraint $\|(\alpha,\beta)\|_\infty \leq 1$ amounts to $2(d+1)$ linear inequalities). Moreover, note that $\mathcal{P}$ is non-empty, since $(a, b) \in \mathcal{P}$ (see Figure 3).

The encoding $\boldsymbol{b}(H)$. The bit-string $\boldsymbol{b} = \boldsymbol{b}(H)$ encodes the polytope $\mathcal{P}$, as well as the names of $d+2$ vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{d+1}$ of $\mathcal{P}$ such that $(a, b) \in \mathrm{conv}(\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{d+1}\})$ (the existence of such vertices is promised by the Carathéodory's Theorem).

The polytope $\mathcal{P}$ can be encoded using $O(d \log(d/\varepsilon))$ bits, as $\mathcal{P}$ is determined by $V^- = H \cap V \in \mathsf{HS}(V)$, and V^- can be described using $\log|\mathsf{HS}(V)| \leq d \log|V| + 1 = O(d \log(d/\varepsilon))$ bits, where the first inequality is because $|\mathsf{HS}(V)| \leq 2|V|^d$ (see, *e.g.*, (Gärtner and Welzl, 1994)).

19. *I.e.*, depends only on U.

The points $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{d+1}$ can be naively conveyed using $O(d^2 \log(d/\varepsilon))$ bits[20]. To obtain a more compressed representation of these points, we use the dual version of Carathéodory Theorem (Theorem 4). Since $\mathcal{P} \subseteq \mathbb{R}^{d+1}$ is defined as the intersection of $|V|+2(d+1)$ halfspaces, Theorem 4 shows such vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{d+1}$ can be represented using $\log(|V|+2(d+1))^{d+1} = O(d \log(d/\varepsilon))$ bits. Thus, total length of $\boldsymbol{b}$ is some $O(d \log(d/\varepsilon))$, which implies an upper bound of $|C| \leq (d/\varepsilon)^{O(d)}$ on the number of containers.

The decoding $C(\boldsymbol{b})$. The next lemma shows how an ε-container $C = C(\boldsymbol{b})$ for H can be derived from $\boldsymbol{b}$, thus concluding the proof of Theorem 16.

Lemma 18 *Let $H = \{u \in U : \langle a, u\rangle < b\}$ as above. Let $(\alpha_0, \beta_0), \ldots, (\alpha_{d+1}, \beta_{d+1})$ be vertices of $\mathcal{P}$ such that $(a, b) \in \operatorname{conv}(\{(\alpha_i, \beta_i)\})$. Then, the set $C = U \setminus (\bigcap_{i=1}^{d+2} H_i)$, where $H_i = \{x : \langle \alpha_i, x\rangle \geq \beta_i\}$, satisfies the two items in Observation 17 with respect to H.*

Proof

(i) $H \subseteq C$: let $u \in H$. Therefore, $u \in U$ and $\langle a, u\rangle < b$. Now, since (a, b) is a convex combination of the (α_i, β_i)'s, it must be the case that $\langle \alpha_i, u\rangle < \beta_i$ for some $i \in \{0, \ldots, d+1\}$, *i.e.*, that $u \notin H_i$. The reason is that we can write $a = \sum_{i=0}^{d+1} \gamma_i \alpha_i$ and $b = \sum_{i=0}^{d+1} \gamma_i \beta_i$ where $\gamma_i \in [0, 1]$. Thus, if $\langle \alpha_i, u\rangle \geq \beta_i$ for all $i \in \{0, \ldots, d+1\}$, then $\langle a, u\rangle = \langle \sum_{i=0}^{d+1} \gamma_i \alpha_i, u\rangle = \sum_{i=0}^{d+1} \gamma_i \langle \alpha_i, u\rangle \geq \sum_{i=0}^{d+1} \gamma_i \beta_i = b$, contradicting the fact that $\langle a, u\rangle < b$. Since there exists $i \in \{0, \ldots, d+1\}$ such that $u \notin H_i$, we get $u \notin \bigcap_i H_i$. This implies $u \in C$, as required.

(ii) $C \cap V = H \cap V$: For every $i \in \{0, \ldots, d+1\}$, since $(\alpha_i, \beta_i) \in \mathcal{P}$, it follows that $V \setminus H_i = \bar{H}_i \cap V = H \cap V = V^-$. This implies $H \cap V = V^- = V \setminus (\bigcap_{i=1}^{m} H_i) = C \cap V$, as required. ■

B.3. Proof of Dual Carathéodory Theorem (Theorem 4)

The Encoding-Decoding Procedure. Let $\mathcal{Q} \subseteq \mathbb{R}^d$ be a polytope which is defined by n linear inequalities and let $\boldsymbol{a} \in \mathcal{Q}$. The proof boils down to an encoding and encoding procedures which are based on bottom vertex triangulation (Clarkson, 1988; Goodman and O'Rourke, 2004) and are described in Figure 6.

The encoding procedure receives $\mathcal{Q}$ and $a \in \mathcal{Q}$ as inputs and outputs a sequence $\boldsymbol{S}$ of d out of the n linear inequalities used to define $\mathcal{Q}$. The decoding procedure receives $\mathcal{Q}$ and $\boldsymbol{S}$ as inputs and output a sequence $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d$ of vertices of $\mathcal{Q}$ such that $a \in \operatorname{conv}(\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d\})$. That is, $\boldsymbol{S}$ encodes a subpolytope defined by $d+1$ vertices that contains a. Since there are at most n^d such sequences $\boldsymbol{S}$ and since every point $a \in \mathcal{Q}$ is contained in one of the encoded subpolytopes, this will imply that $\mathcal{Q}$ can be covered by n^d such subpolytopes as required.

We use the following convention: for every polytope $\mathcal{Q}'$, fix a pivot vertex $\boldsymbol{p}(\mathcal{Q}') \in \mathcal{Q}'$ (for example, $\boldsymbol{p}(\mathcal{Q}')$ can be the bottom vertex in $\mathcal{Q}$, or the smallest vertex with respect to the lexicographical order, etcetera). Also, let $\dim(\mathcal{Q}')$ denote the dimension of $\mathcal{Q}'$ (*i.e.*, the dimension of the affine span[21] of Q).

20. To see this, observe that the number of vertices in $\mathcal{P}$ is $O(\binom{|V|+2(d+1)}{d+1}) = \exp(d \log(d/\varepsilon))$, because $\mathcal{P}$ is defined by $|V| + 2(d+1)$ constraints, and each vertex is determined by $d+1$ constraints. Therefore, each vertex can be described using $O(d \log(d/\varepsilon))$ bits, and $d+2$ vertices can be represented by $O(d^2 \log(d/\varepsilon))$ bits.
21. Recall that the affine span of a set A is the minimal affine subspace that contains A.

A Dual Carathéodory's Theorem

Encoding:
Input: a polytope $\mathcal{Q} \in \mathbb{R}^d$ which is defined by n constraints (linear inequalities) and a point $\boldsymbol{a} \in \mathcal{Q}$.
Output: a sequence $\boldsymbol{S}$ of d constraints which encodes vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d \in \mathcal{Q}$ such that $\boldsymbol{a} \in \mathrm{conv}(\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d\})$.

(1) Initialize $\mathcal{Q}_0 = \mathcal{Q}$, $\boldsymbol{a}_0 = \boldsymbol{a}$, $\boldsymbol{x}_0 = \boldsymbol{p}(\mathcal{Q}_0)$, and $\boldsymbol{S} = \boldsymbol{\varepsilon}$ (the empty sequence).
($\boldsymbol{p}(\mathcal{Q}')$ denotes the bottom vertex of a polytope $\mathcal{Q}'$.)

(2) For $i = 1, \ldots, d$:

(2.1) Extend the ray that starts at $\boldsymbol{x}_{i-1}$ and passes through $\boldsymbol{a}_{i-1}$ until it hits the boundary of $\mathcal{Q}_{i-1}$.

(2.2) Set $\boldsymbol{a}_i$ to be the point on the boundary of $\mathcal{Q}_{i-1}$ that the ray hits. Set $\mathcal{Q}_i$ to be the[a] facet of $\mathcal{Q}_{i-1}$ that contains $\boldsymbol{a}_i$ and set $\boldsymbol{x}_i = \boldsymbol{p}(\mathcal{Q}_{i+1})$.

(2.3) Append to $\boldsymbol{S}$ the linear inequality which is tightened by $\mathcal{Q}_i$.

(3) Output $\boldsymbol{S}$.

Decoding:
Input: a polytope $\mathcal{Q} \in \mathbb{R}^d$ which is defined by n constraints (linear inequalities) and a sequence $\boldsymbol{S}$ of d constraints.
Output: a sequence of vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d \in \mathcal{Q}$.

(1) Initialize $\mathcal{Q}_0 = \mathcal{Q}$, $\boldsymbol{x}_0 = \boldsymbol{p}(\mathcal{Q}_0)$.

(2) For $i = 1, \ldots, d$:

(2.1) Set $\mathcal{Q}_i$ to be the facet of $\mathcal{Q}_{i-1}$ which is defined by tightening the i'th constraint in $\boldsymbol{S}$.

2.2 Set $\boldsymbol{x}_i = \boldsymbol{p}(\mathcal{Q}_i)$.

(3) Output $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d$.

a. If a_{i+1} belongs to several facets (*i.e.*, it sits on a face whose dimension is $< d-1$) then pick $\mathcal{Q}_{i+1}$ to be any facet that contains it.

Figure 6: The encoding and decoding procedures for the Dual Carathéodory's Theorem.

Analysis. The description of the encoding and decoding procedures appears in Figure 6. We finish the proof by showing that $\boldsymbol{a} \in \mathrm{conv}(\{\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d\})$. This follows by induction on $\dim(\mathcal{Q})$: the base case of $\dim(\mathcal{Q}) = 0$ is trivial. For the induction step, assume that the claim holds for every polytope of dimension strictly less than k, and prove the claim for $\dim(\mathcal{Q}) = k$: by construction, $\boldsymbol{a}$ is a convex combination of $\boldsymbol{x}_0$ and $\boldsymbol{a}_1$. Since $\dim(\mathcal{Q}_1) = k - 1$, by the induction hypothesis, $\boldsymbol{a}_1$ is in the convex hull of $\boldsymbol{x}_1 \ldots \boldsymbol{x}_d$. This implies that a is in the convex hull of $\boldsymbol{x}_0 \ldots \boldsymbol{x}_d$, as required.

B.4. The Computational Complexity of Finding Containers

The proofs of Theorem 16 and Theorem 4 imply a polynomial time algorithm for finding containers. In more detail, given a universe $U \subseteq \mathbb{R}^d$, during preprocessing we sample the ε-net V of size $\tilde{O}(d^2/\varepsilon)$ (this is the only step which requires randomness; thus, the whole algorithm can be derandomized if such an ε-net can be found deterministically). Assume now that we are given a halfspace H as an input and we wish to compute an ε-container of H. We proceed as in the above proof by determining the auxiliary polytope $\mathcal{P}$ (in $O(|V|)$ time), and detecting the vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d$ whose convex-hull contains H, as in the proof of Theorem 4, which also takes $\mathrm{poly}(|V|)$ time[22]. Once the vertices $\boldsymbol{x}_0, \ldots, \boldsymbol{x}_d$ are retrieved, a container C for H is obtained as in Theorem 18.

The computational complexity of our protocols. As a consequence it follows that also our protocols for learning halfspaces and for convex set disjointness can be implemented efficiently, at least if we assume that Alice and Bob have access to shared randomness. (The shared randomness is needed in order for Alice and Bob to agree on an ε-net V which is used to define the set of containers in each round.)

This gives an advantage over the construction of containers using cuttings, which requires preprocessing time which is exponential in d. The benefit of the cuttings-construction is that once preprocessing is done, finding the container for an input halfspace H is done very fast in $O(d \log n)$ time. Thus, if it is assumed that many such queries will be made, the cost of preprocessing will be amortized over the (many) queries. In the application considered in this paper only one query is made per a constructed family of containers. (In each round Alice and Bob construct a family of containers and then at most one of them computes a container, see Figure 7).

Appendix C. Protocols for CSD

In this section, we prove the following upper bound on the communication complexity of the CSD problem and its promise variant PromiseCSD:

Theorem 19 *Let $d, n \in \mathbb{N}$, and let $U \subseteq \mathbb{R}^d$ be a domain with n points. Then,*

$$\mathrm{D}(\mathsf{PromiseCSD}_U) = O(d \log d \log n).$$

Theorem 20 *Let $d, n \in \mathbb{N}$, and let $U \subseteq \mathbb{R}^d$ be a domain with n points. Then,*

$$\mathrm{D}(\mathsf{CSD}_U) = O(d^2 \log d \log n).$$

Observe that Theorem 20 implies the upper bound in Theorem 2.

22. Note that this requires solving $d + 1$ linear programs, as we need to compute the bottom vertex $\boldsymbol{p}(\mathcal{Q}')$ for the faces $\mathcal{Q}'$ which are encountered in the algorithm in Figure 6.

C.1. The Protocol for PromiseCSD (Theorem 19)

We next prove the following lemma, which implies Theorem 19.

Lemma 21 *Let $U \subseteq \mathbb{R}^d$ with $|U| = n$. Then, the protocol in Figure 7 witnesses that $\mathrm{D}(\mathsf{PromiseCSD}_U) = O(d \log d \log n)$. Furthermore, for inputs $X, Y \subseteq U$ such that $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$, the protocol outputs a function $h : U \to \{\pm 1\}$ such that $X \subseteq h^{-1}(-1)$ and $Y \subseteq h^{-1}(+1)$.*

The function h promised by the above lemma will later be used for learning halfspaces.

Proof A complete description of the protocol is presented in Figure 7. The correctness is based on the following simple observation:

Observation 22 *Consider the sets U_i, X_i, Y_i in the "While" loop in item (2) of the protocol in Figure 7.*

1. *If $\mathrm{conv}(X_i) \cap \mathrm{conv}(Y_i) = \emptyset$ then there is a halfspace $H \in \mathsf{HS}(U_i)$ such that $|H| \leq |U_i|/2$, and either $X_i \subseteq H$ or $Y_i \subseteq H$.*
2. *$X_i \cap Y_i = X_{i+1} \cap Y_{i+1}$.*

The first item follows since $\mathrm{conv}(X_i) \cap \mathrm{conv}(Y_i) = \emptyset$ implies that there is a hyperplane that separates X_i from Y_i, and therefore one of the two halfspaces defined by this hyperplane contains at most half of the points in U_i.

The second item follows since $C \in \mathcal{C}_i$ either contains X_i or Y_i. If $C \supseteq X_i$ then $X_{i+1} = X_i$ and $Y_{i+1} = Y_i \cap C \supseteq Y_i \cap X_i$. Otherwise, $C \supseteq Y_i$ and $X_{i+1} = X_i \cap C \supseteq X_i \cap Y_i$ and $Y_{i+1} = Y_i$. In both cases, $X_i \cap Y_i = X_{i+1} \cap Y_{i+1}$.

Correctness. We first assume that $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$. Consider iteration i of the "While" loop. Since $X_i \subseteq X$ and $Y_i \subseteq Y$, it holds that $\mathrm{conv}(X_i) \cap \mathrm{conv}(Y_i) \subseteq \mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$. By the first item of Observation 22, either Alice or Bob always find a container $C \in \mathcal{C}_i$ in item (2.2), and therefore the protocol will reach items (2.4) and (2.5). Since the protocol will never reach item (2.3), the "While" loop will eventually terminate with $|U_i| = 0$ and item (3) will be reached, outputting "1" as required. To see that the output function h satisfies $X \subseteq h^{-1}(-1), Y \subseteq h^{-1}(1)$, note that at the i'th step, h is defined over all points in $U \setminus U_i$ and satisfies $X \setminus X_i \in h^{-1}(-1), Y \setminus Y_i \in h^{-1}(1)$. Thus, the requirement is met since at the last iteration i^* we have $U_{i^*} = X_{i^*} = Y_{i^*} = \emptyset$.

Next, assume that $X \cap Y \neq \emptyset$. In this case, the protocol must terminate in item (2.3) within the "While" loop. This is because, by the second item of Observation 22, $|X_i \cap Y_i|$ is a positive constant for all i while $|U_i|$ decreases, thus eventually $X_i \cap Y_i$ becomes larger than $\frac{3}{4}|U_i|$. When this happens, no party can find a set C satisfying the requirements of (2.2) and the protocol outputs "0".

Communication Complexity. The "While" loop in item (2) proceeds for at most $O(\log n)$ iterations; this is because in each iteration U_i shrinks by a multiplicative factor of at most $3/4$. In each of the iterations the parties exchange $\log|\mathcal{C}_i| + O(1)$ bits, which is bounded by $O(d \log d)$ bits. Thus, the total number of bits communicated is $O(d \log d \log n)$.

■

The $O(d \log d \log n)$-bits Deterministic Protocol for $\mathsf{PromiseCSD}_U$

Let $U \subseteq \mathbb{R}^d$ and let $n = |U|$.
Alice's input: $X \subseteq U$,
Bob's input: $Y \subseteq U$.
Output: if $X \cap Y \neq \emptyset$ output "0",
if $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$ output "1" as well as a function $h : U \to \{\pm 1\}$ such that $X \subseteq h^{-1}(-1)$ and $Y \subseteq h^{-1}(+1)$ (h will be used in our learning protocol).

(1) Set $i = 1, U_1 = U, X_1 = X, Y_1 = Y, \varepsilon = 1/4$, and f as the empty function.

(2) While $|U_i| > 0$:

  (2.1) Without communication, the parties agree on a set $\mathcal{C}_i$ of ε-containers $\mathsf{HS}(U_i)$, such that $|\mathcal{C}_i| = (d/\varepsilon)^{O(d)}$ (as in Theorem 16).

  (2.2) Each of Alice and Bob checks whether there is $C \in \mathcal{C}_i$ such that $|C| \leq \frac{3}{4}|U_i|$ and C contains their current set; namely, Alice looks for such a $C \in \mathcal{C}_i$ that contains X_i and Bob looks for such a $C \in \mathcal{C}_i$ that contains Y_i.

  (2.3) If both Alice and Bob cannot find such a C then the protocol terminates with output "0".

  (2.4) Else, if Alice found C then she communicates it to Bob (using $O(d \log d)$ bits), and the parties do:

    (2.4.1) set $X_{i+1} = X_i \cap C,\ Y_{i+1} = Y_i \cap C,\ U_{i+1} = U_i \cap C$,

    (2.4.2) extend h to $U_i \setminus C$ by setting $h(u) = 0$ for all $u \in U_i \setminus C$,

    (2.4.3) increment $i \leftarrow i + 1$ and go to (2).

  (2.5) Similarly, if Bob found C then he communicates it to Alice (using $O(d \log d)$ bits), and the parties do:

    (2.4.1) set $X_{i+1} = X_i \cap C,\ Y_{i+1} = Y_i \cap C,\ U_{i+1} = U_i \cap C$,

    (2.4.2) extend h to $U_i \setminus C$ by setting $h(u) = 1$ for all $u \in U_i \setminus C$,

    (2.4.3) increment $i \leftarrow i + 1$ and go to (2).

(3) Output "1" and the function h.

Figure 7: The protocol for PromiseCSD.

C.2. The Protocol for CSD (Theorem 20)

In this section we show how to use a protocol for PromiseCSD to solve CSD with some loss in the parameters. To this end, we will prove a geometric theorem we call "a symmetric variant of Carathéodory's Theorem".

C.2.1. SYMMETRIC CARATHÉODORY

Carathéodory's Theorem concerns a relation between a point x and a set Y such that $x \in \text{conv}(Y)$. The following simple generalization provides a symmetric relation between two set X, Y such that $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset$.

Proposition 23 (A symmetric variant of Carathéodory's Theorem) *Let $X, Y \subseteq \mathbb{R}^d$ such that $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset$. Then $\text{conv}(S_1) \cap \text{conv}(S_2) \neq \emptyset$ for some $S_1 \subseteq X, S_2 \subseteq Y$ such that $|S_1| + |S_2| \leq d + 2$.*

Note that Carathéodory's Theorem boils down to the case where $X = \{x\}$ (and hence $\text{conv}(X) \cap \text{conv}(Y) \neq \emptyset \implies x \in \text{conv}(Y)$).

Proof [Proof of Theorem 23] The proof follows an argument similar to the linear algebraic proof of Carathéodory's Theorem. Assume $z \in \text{conv}(X) \cap \text{conv}(Y)$ can be represented as a convex combination of d_1 points $x_1 \dots x_{d_1} \in X$ and as a convex combination of d_2 points $y_1 \dots y_{d_2} \in Y$ such that $d_1 + d_2 > d + 2$. Consider the system of linear equalities in $d_1 + d_2$ variables $\alpha_1 \dots \alpha_{d_1}, \beta_1 \dots \beta_{d_2}$ defined by the constraints (i) $\sum \alpha_i x_i = \sum \beta_j y_j$, and (ii) $\sum \alpha_i = \sum \beta_j = 0$. This system has $d_1 + d_2 > d + 2$ variables and only $d + 2$ constraints (d constraints from (i) and 2 more constraints from (ii)). Thus, it has a solution such that not all α_i's and β_j's are 0. Consequently, one can shift z by a sufficiently small scaling of the vector $v = \sum \alpha_i x_i = \sum \beta_i y_i$, so that one of the coefficients of the x_i's or the y_j's vanishes. This process can be repeated until $d_1 + d_2 \leq d + 2$, which yields the desired sets $S_1 \subseteq X, S_2 \subseteq Y$.

■

Remark. Theorem 23 establishes a tight bound of $d + 2$ on the *coVC number* of halfspaces in $\mathbb{R}^d$. The coVC number is a combinatorial parameter which characterizes the concept classes that can be properly learned using polylogarithmic communication complexity (see Kane et al. (2019)). It is defined as follows: let $H \subseteq \{\pm 1\}^X$ be an hypothesis class over a domain X. Its coVC number is the smallest number k such that every sample $S \subseteq X \times \{\pm 1\}$ which is not realizable[23] by H has a subsample $S' \subseteq S$ of size $|S'| \leq k$ which is not realizable by H. A weaker upper bound of $2d + 2$ on the coVC number of halfspaces was given by Kane et al. (2019) (see Example 1 in their paper).

C.2.2. REDUCING CSD TO PromiseCSD

The next lemma implies that a bound of $C = C(n, d)$ on the communication complexity of the promise variant implies a bound of $C'(n, d) = C((2n)^{d+2}, d)$ on the communication complexity of the non-promise variant. The lemma implies Theorem 20 by plugging $(2n)^{d+2}$ instead of n in Lemma 21.

23. A sample S is realizable with respect to H if there is $h \in H$ such that $h(x) = y$ for every $(x, y) \in S$.

Lemma 24 *For any $U \subseteq \mathbb{R}^d$ of size n there is $V \subseteq \mathbb{R}^d$ of size at most $(2n)^{d+2}$ such that*

$$\mathsf{CSD}_U \preceq \mathsf{PromiseCSD}_V.$$

(Recall that "$\preceq$" denotes a reduction with zero communication, see Definition 11).

Proof

The set V is defined as follows: for any $S_1, S_2 \subseteq U$ such that $\mathrm{conv}(S_1) \cap \mathrm{conv}(S_2) \neq \emptyset$ and $|S_1| + |S_2| \leq d+2$ add to V (any) point $x = x(S_1, S_2) \in \mathrm{conv}(S_1) \cap \mathrm{conv}(S_2)$. Note that indeed $|V| \leq \sum_{d_1+d_2=d+2} \binom{|U|}{d_1}\binom{|U|}{d_2} \leq (2n)^{d+2}$. Next, given inputs $X, Y \subseteq U$ for CSD_U, Alice and Bob transform them to

$$\alpha(X) = \mathrm{conv}(X) \cap V \text{ and } \beta(Y) = \mathrm{conv}(Y) \cap V.$$

Validity. To establish the validity of this reduction we need to show that

$$\begin{aligned} \mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset &\implies \mathrm{conv}(\alpha(X)) \cap \mathrm{conv}(\beta(Y)) = \emptyset, \text{ and} \\ \mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset &\implies \alpha(X) \cap \beta(Y) \neq \emptyset. \end{aligned}$$

Indeed, if $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$ then also $\mathrm{conv}(\alpha(X)) \cap \mathrm{conv}(\alpha(Y)) = \emptyset$ (because $\alpha(X) \subseteq \mathrm{conv}(X)$ and $\beta(Y) \subseteq \mathrm{conv}(Y)$).

The second assertion follows from Theorem 23. To see how Theorem 23 implies the second assertion, assume that $\mathrm{conv}(X) \cap \mathrm{conv}(Y) \neq \emptyset$. By Theorem 23, there exists $S_1 \subseteq X, S_2 \subseteq Y$ with $|S_1| + |S_2| \leq d+2$ and $\mathrm{conv}(S_1) \cap \mathrm{conv}(S_2) \neq \emptyset$. By construction, V contains a point $x = x(S_1, S_2)$ in $\mathrm{conv}(S_1) \cap \mathrm{conv}(S_2)$. It holds that $x \in \mathrm{conv}(S_1) \cap V \subseteq \mathrm{conv}(X) \cap V = \alpha(X)$ and $x \in \mathrm{conv}(S_2) \cap V \subseteq \mathrm{conv}(Y) \cap V = \beta(Y)$. Hence, $\alpha(X) \cap \beta(Y) \neq \emptyset$, as claimed. ■

Appendix D. Lower Bound for Convex Set Disjointness

In this section we prove a lower bound on the randomized communication complexity of PromiseCSD, and therefore also of CSD.

Theorem 25 *Let $d, n \geq 2$. There is a set $U \subseteq \mathbb{R}^d$ with n points such that $R(\mathsf{PromiseCSD}_U) \geq \Omega(d \log(n/d))$.*

Observe that Theorem 25 implies the lower bound in Theorem 2.

The key ingredient in the proof of Theorem 25 is the following reduction:

Lemma 26 *For any integers $c, k > 0$, there is a set $U \subseteq \mathbb{R}^{3c}$ such that $|U| = 2^k c$ and*

$$\mathsf{DISJ}_{ck} \preceq \mathsf{PromiseCSD}_U.$$

We prove Theorem 26 below. Assuming Theorem 26, the following argument proves Theorem 25.

Proof [Proof of Theorem 25] Fix d and n. For $d = 2$, the required lower bound follows from the $\Omega(d + \log n)$ lower bound of Kane et al. (2019). We therefore assume $d \geq 3$. Set $c = d/3$ and

set k such that $n = 2^k c$. We assume without loss of generality that k, c are positive integers. By Theorem 26, there is set $U \subseteq \mathbb{R}^d$, $|U| = n$ such that

$$\mathsf{DISJ}_{ck} \preceq \mathsf{PromiseCSD}_U.$$

Using the the well known fact that $R(\mathsf{DISJ}_m) \geq \Omega(m)$ (see, *e.g.*, Kalyanasundaram and Schintger (1992)), and Observation 13, it follows that

$$R(\mathsf{PromiseCSD}_U) \geq \Omega(ck) = \Omega(d \log(n/d)).$$

■

D.1. Proof of Theorem 26

Let $c, k > 0$ be arbitrary. To prove Theorem 26, we show that there exist sets $U \subseteq \mathbb{R}^{3c}, V \subseteq \mathbb{R}^2$ such that $|U| = 2^k c$ and $|V| = 2^k$ such that the following sequence of reductions holds

$$\mathsf{DISJ}_{ck} \preceq \mathsf{AND}_c \circ \mathsf{DISJ}_k \preceq \mathsf{AND}_c \circ \mathsf{PromiseCSD}_V \preceq \mathsf{PromiseCSD}_U.$$

Each of these reductions is proved separately below. Theorem 26 then follows using Observation 12.

Proving $\mathsf{DISJ}_{ck} \preceq \mathsf{AND}_c \circ \mathsf{DISJ}_k$. The first reduction in our sequence is essentially using the fact that DISJ_m can be viewed as an AND of m simpler functions.

Lemma 27 $\mathsf{DISJ}_{ck} \preceq \mathsf{AND}_c \circ \mathsf{DISJ}_k$

Proof Let $\boldsymbol{x}^*, \boldsymbol{y}^* \in \{0,1\}^{ck}$ be an input for DISJ_{ck}. We can view $\boldsymbol{x}^*$ as a vector $\boldsymbol{x}^{(c)}$ with entries in $\mathbb{R}^k$. Precisely, $\boldsymbol{x_i}$ (respectively $\boldsymbol{y_i}$) is the $((i-1)k+1)^{\text{st}}$ to $(ik)^{\text{th}}$ coordinates of $\boldsymbol{x}^*$ (resp. $\boldsymbol{y}^*$). Let the reduction function α (resp. β) be the function that takes $\boldsymbol{x}^*$ to $\boldsymbol{x}$ (resp. $\boldsymbol{y}^*$ to $\boldsymbol{y}$). Note that:

$$\begin{aligned}
\mathsf{DISJ}_{ck}(\boldsymbol{x}^*, \boldsymbol{y}^*) = 0 &\iff \exists i \in [ck] \colon x^*_i = y^*_i = 1 \\
&\iff \exists i \in [c], j \in [k] \colon x_{ij} = y_{ij} = 1 \\
&\iff \exists i \in [c] \colon \mathsf{DISJ}_k(\boldsymbol{x_i}, \boldsymbol{y_i}) = 0 \\
&\iff \left(\bigwedge_{i=1}^{c} \mathsf{DISJ}_k(\boldsymbol{x_i}, \boldsymbol{y_i}) \right) = 0 \\
&\iff \mathsf{AND}_c \circ \mathsf{DISJ}_k(\boldsymbol{x}, \boldsymbol{y}) = 0 \\
&\iff \mathsf{AND}_c \circ \mathsf{DISJ}_k(\alpha(\boldsymbol{x}^*), \beta(\boldsymbol{y}^*)) = 0.
\end{aligned}$$

■

Proving $\mathsf{AND}_c \circ \mathsf{DISJ}_k \preceq \mathsf{AND}_c \circ \mathsf{PromiseCSD}_V$. By Theorem 14, the following result is sufficient:

Lemma 28 *For all $k > 0$, there exists $V \subseteq \mathbb{R}^2$, $|V| = 2^k$ such that $\mathsf{DISJ}_k \preceq \mathsf{PromiseCSD}_V$.*

Proof We define the set V to consist of 2^k points on the unit circle in $\mathbb{R}^2$. The crucial property satisfied by these set of points is that every $\boldsymbol{v} \in V$ can be separated by a line from $V \setminus \{\boldsymbol{v}\}$ (*i.e.*, these points are in *convex position*). Let us index the points in V by the vectors in $\{0,1\}^k$, *i.e.*, $V = \{\boldsymbol{v_x} \mid \boldsymbol{x} \in \{0,1\}^k\}$ (see Figure 5).

We next define the functions α, β which witness the desired reduction. Define $\alpha : \{0,1\}^k \to 2^V$ by

$$\alpha(\boldsymbol{x}) = \{\boldsymbol{v_x}\}.$$

Next, define $\beta : \{0,1\}^k \to 2^V$ as

$$\beta(\boldsymbol{y}) = \{\boldsymbol{v_z} \text{ for } \boldsymbol{z} \in \{0,1\}^k \text{ such that } \exists i \in [k] : z_i = y_i = 1\}.$$

Observe that for every input $\boldsymbol{x} \in \{0,1\}^k$, the set $\alpha(\boldsymbol{x}) = \{\boldsymbol{v_x}\}$ is a singleton. Thus, for every possible $\boldsymbol{y} \in \{0,1\}^k$, it is either the case that $\boldsymbol{v_x} \in \beta(\boldsymbol{y})$, or else, since $\boldsymbol{x} \in V$ and $\beta(\boldsymbol{y}) \subseteq V$, and due to the crucial property described above, it is the case that $\boldsymbol{v_x} \notin \mathrm{conv}(\beta(\boldsymbol{y}))$. Equivalently, it is either the case that $\alpha(\boldsymbol{x}) \cap \beta(\boldsymbol{y}) \neq \emptyset$ or that $\mathrm{conv}(\alpha(\boldsymbol{x})) \cap \mathrm{conv}(\beta(\boldsymbol{y})) = \emptyset$, thus the sets $\alpha(\boldsymbol{x})$ and $\beta(\boldsymbol{y})$ are in the domain of $\mathsf{PromiseCSD}_V$.

We have

$$\begin{aligned} \mathsf{DISJ}_k(\boldsymbol{x},\boldsymbol{y}) = 0 &\iff \exists i \in [k] \colon x_i = y_i = 1 \\ &\iff \alpha(\boldsymbol{x}) \cap \beta(\boldsymbol{y}) \neq \emptyset \\ &\iff \mathsf{PromiseCSD}_V(\alpha(\boldsymbol{x}), \beta(\boldsymbol{y})) = 0. \end{aligned}$$

■

Proving $\mathsf{AND}_c \circ \mathsf{PromiseCSD}_V \preceq \mathsf{PromiseCSD}_U$.

Lemma 29 *Let $V \subseteq \mathbb{R}^2$, $|V| = m$. For all integers $c > 0$, there is a set $U \subseteq \mathbb{R}^{3c}$ of size $c \cdot m$ such that*

$$\mathsf{AND}_c \circ \mathsf{PromiseCSD}_V \preceq \mathsf{PromiseCSD}_U.$$

Proof

We embed each of the c copies of $\mathsf{PromiseCSD}_V$ in a disjoint triplet of coordinates of $\mathbb{R}^{3c}$. Formally, for $j \in [c]$, define the j^{th} 'lift' function $g_j : \mathbb{R}^2 \to \mathbb{R}^{3c}$ as:

$$g_j((x_1,x_2)) = (\underbrace{0,0,\cdots,0}_{3(j-1) \text{ times}}, x_1, x_2, 1, \underbrace{0,0,\cdots,0}_{3(c-j) \text{ times}}).$$

Define the set $U = \{g_j(v) \mid j \in [c], v \in V\}$.

Let $\boldsymbol{X}^{(c)}, \boldsymbol{Y}^{(c)}$ be an input for $\mathsf{AND}_c \circ \mathsf{PromiseCSD}_V$. Define:

$$\alpha(\boldsymbol{X}) = \bigcup_{j=1}^{c} g_j(X_j) \qquad \beta(\boldsymbol{Y}) = \bigcup_{j=1}^{c} g_j(Y_j).$$

(Recall that X_j, Y_j denote the j'th copies of $\boldsymbol{X}^{(c)}, \boldsymbol{Y}^{(c)}$ respectively.) We prove that α, β define the desired reduction. First, assume that $\mathsf{AND}_c \circ \mathsf{PromiseCSD}_V(\boldsymbol{X}, \boldsymbol{Y}) = 1$, that is, $\forall j \in [c], \mathrm{conv}(X_j) \cap \mathrm{conv}(Y_j) = \emptyset$. By the hyperplane separation theorem, for every $j \in [c]$ there exists an affine function $l_j : \mathbb{R}^2 \to \mathbb{R}$ of the form $l_j((x_1, x_2)) = l_j + l'_j x_1 + l''_j x_2$ such that $l_j(x) > 0$ for all $x \in X_j$, while $l_j(y) < 0$ for all $y \in Y_j$.

Define the affine function $l : \mathbb{R}^{3c} \to \mathbb{R}$ by $l((x_1, x_2, \cdots, x_{3c})) = \sum_{i \in [c]} l_j x_{3j} + l'_j x_{3j-2} + l''_j x_{3j-1}$. Observe that for all $j \in [c]$, we have $\forall (x_1, x_2) \in \mathbb{R}^2 \colon l(g_j((x_1, x_2))) = l_j((x_1, x_2))$. This implies that $l(x) > 0$ for all $x \in \alpha(\boldsymbol{X})$, while $l(y) < 0$ for all $y \in \beta(\boldsymbol{Y})$. Thus, $\alpha(\boldsymbol{X}) \cap \beta(\boldsymbol{Y}) = \emptyset$, implying $\mathsf{PromiseCSD}_U(\alpha(\boldsymbol{X}), \beta(\boldsymbol{Y})) = 1$.

For the other direction, assume that $\mathsf{AND}_c \circ \mathsf{PromiseCSD}_V(\boldsymbol{X}, \boldsymbol{Y}) = 0$, that is, $\exists j \in [c], z \in V \colon \ z \in X_j \cap Y_j$. Then, $g_j(z) \in \alpha(\boldsymbol{X}) \cap \beta(\boldsymbol{Y})$, implying $\alpha(\boldsymbol{X}) \cap \beta(\boldsymbol{Y}) \neq \emptyset$ and therefore also $\mathsf{PromiseCSD}_U(\alpha(\boldsymbol{X}), \beta(\boldsymbol{Y})) = 0$.

■

Appendix E. Bound for Distributed Learning of Halfspaces

E.1. The Learning Protocol

We next prove the following upper bound for learning halfspaces, which implies the upper bound in Theorem 1.

Theorem 30 *Let $d, n \in \mathbb{N}$, and let $U \subseteq \mathbb{R}^d$ be a domain with n points. Then, there exists a deterministic protocol for the problem of two-party distributed learning of halfspaces over U with communication complexity $O(d \log d \log n)$.*

Proof We present a learning protocol which relies on Theorem 19 and uses the protocol in Figure 7 as a black-box. The learning protocol is presented in Figure 8.

Analysis. First, note that the communication complexity is at most $O(d \log d \log n)$ bits: indeed, there is no communication in steps (3) and (6), each of steps (1) and (2) involves an application of the protocol from Figure 7 which costs $O(d \log d \log n)$ bits, and each of steps (4) and (5) involves transmitting a separator from $\mathsf{HS}(U)$ which costs $O(d \log n)$ bits (since $|\mathsf{HS}(U)| \leq O(n^d)$, see *e.g.* (Gärtner and Welzl, 1994)).

As for correctness, note that since it is assumed that the negative and positive examples in $S_a \cup S_b$ are separated by a hyperplane, Theorem 19 implies that the functions f, g which are outputted in steps (1) and (2) satisfy:

- $f(\boldsymbol{u}) = +1$ for every $(\boldsymbol{u}, +1) \in S_a$ and $f(\boldsymbol{u}) = -1$ for every $(\boldsymbol{u}, -1) \in S_b$, and similarly
- $g(\boldsymbol{u}) = -1$ for every $(\boldsymbol{u}, -1) \in S_a$ and $g(\boldsymbol{u}) = +1$ for every $(\boldsymbol{u}, +1) \in S_b$.

We will show that the h (the function outputted by the protocol) classifies correctly each of the regions $F^+ \cap G^+, F^- \cap G^-, F^+ \cap G^-$, and $F^- \cap G^+$ (the definition of these regions appears in the protocol). Since these 4 regions cover U, it will follow that h classifies correctly all examples. Indeed $F^+ \cap G^+$ contains only positive examples and $F^- \cap G^-$ contains only negative examples, therefore h classifies correctly these regions. As for $F^+ \cap G^-$ and $F^- \cap G^+$, note that $F^+ \cap G^-$

The $O(d \log d \log n)$-bits Deterministic Learning Protocol for Halfspaces

Let $U \subseteq \mathbb{R}^d$ and let $n = |U|$.
Alice's input: a sample $S_a \subseteq U \times \{\pm 1\}$.
Bob's input: a sample $S_b \subseteq U \times \{\pm 1\}$.
Assumption: $\exists$ hyperplane separating the positive and negative examples in $S_a \cup S_b$.
Output: a function $h : U \to \{\pm 1\}$ such that $h(\boldsymbol{x}) = y$ for every $(\boldsymbol{x}, y) \in S_a \cup S_b$.

(1) Apply the protocol from Figure 7 on inputs X^-, Y^+, where $X^- = \{\boldsymbol{u} : (\boldsymbol{u}, -1) \in S_a\}$ and $Y^+ = \{\boldsymbol{u} : (\boldsymbol{u}, +1) \in S_b\}$.

  (1.1) If the protocol outputted "0" then output "Error".

  (1.2) Else, let $g : U \to \{\pm 1\}$ denote the function outputted by the protocol, such that $g(\boldsymbol{u}) = +1$ for every $\boldsymbol{u} \in Y^+$ and $g(\boldsymbol{u}) = -1$ for every $\boldsymbol{u} \in X^-$.

(2) Apply the protocol from Figure 7 on inputs X^+, Y^-, where $X^+ = \{\boldsymbol{u} : (\boldsymbol{u}, +1) \in S_a\}$ and $Y^- = \{\boldsymbol{u} : (\boldsymbol{u}, -1) \in S_b\}$.

  (2.1) If the protocol outputted "0" then output "Error".

  (2.2) Else, let $f : U \to \{\pm 1\}$ denote the negation of the function outputted by the protocol, such that $f(\boldsymbol{u}) = +1$ for every $\boldsymbol{u} \in X^+$ and $f(\boldsymbol{u}) = -1$ for every $\boldsymbol{u} \in Y^-$.

(3) Let $F^+ = f^{-1}(+1), F^- = f^{-1}(-1)$ and $G^+ = g^{-1}(+1), G^- = g^{-1}(-1)$. (Note that these 4 sets are known to both Alice and Bob.)

(4) Alice transmits to Bob, using $O(d \log n)$ bits, an indicator $I_{+-} : U \to \{\pm 1\}$ of a halfspace in $\mathsf{HS}(U)$ which separates her positive and negative examples in $F^+ \cap G^-$; namely, $I_{+-}(\boldsymbol{u}) = b$ for every $\boldsymbol{u} \in F^+ \cap G^-$ such that $(\boldsymbol{u}, b) \in S_a$.

(5) Bob transmits to Alice, using $O(d \log n)$ bits, an indicator $I_{-+} : U \to \{\pm 1\}$ of a halfspace in $\mathsf{HS}(U)$ which separates his positive and negative examples in $F^- \cap G^+$; namely, $I_{-+}(\boldsymbol{u}) = b$ for every $\boldsymbol{u} \in F^- \cap G^+$ such that $(\boldsymbol{u}, b) \in S_b$.

(6) Alice and Bob output the function h defined by

$$h(u) = \begin{cases} +1 & \boldsymbol{u} \in F^+ \cap G^+, \\ -1 & \boldsymbol{u} \in F^- \cap G^-, \\ I_{+-}(u) & \boldsymbol{u} \in F^+ \cap G^-, \\ I_{-+}(u) & \boldsymbol{u} \in F^- \cap G^+. \end{cases}$$

Figure 8: The protocol for two-party distributed learning of halfspaces over U.

contains only examples in S_a and $F^- \cap G^+$ contains only examples in S_b. Thus, I_{+-} classifies correctly every example in $F^+ \cap G^-$ and I_{-+} classifies correctly every example in $F^- \cap G^+$. It therefore follows that h classifies correctly also these regions. ■

Remark. The above protocol actually learns a more general problem than halfspaces: let S_a^+, S_a^- denote Alice's positive and negative examples respectively, and let S_b^+, S_b^- denote Bob's positive and negative examples respectively. The protocol will output a consistent function h for as long as each of the pairs S_a^+ and S_a^-, S_b^+ and S_b^-, S_a^+ and S_b^-, and S_b^+ and S_a^- can be separated by a hyperplane (possibly a different hyperplane for every pair). However it is not necessary that there will be a single hyperplane separating all positive examples from all negative examples.

E.2. Learning Lower Bound

We next prove the following lower bound for learning halfspaces, which implies the lower bound in Theorem 1.

Theorem 31 *Let $d, n \geq 2$. Then, there exists a domain $U \subseteq \mathbb{R}^d$ with n points such that every (possibly improper and randomized) protocol for the problem of two-party distributed learning of halfspaces over U must communicate at least $\Omega(d \log(n/d))$ bits.*

Proof This is a corollary of Theorem 25: let $U \subseteq \mathbb{R}^d$ be as in the conclusion of Theorem 25. We claim that every protocol that learns $\mathsf{HS}(U)$ can be used to decide $\mathsf{PromiseCSD}_U$. Indeed, let X, Y be inputs to $\mathsf{PromiseCSD}_U$. Alice and Bob apply the learning protocol on the samples $X \times \{+1\}$ and $Y \times \{-1\}$. (i) If $\mathrm{conv}(X) \cap \mathrm{conv}(Y) = \emptyset$ then X, Y can be separated by a hyperplane and the protocol will output a function $h : U \to \{\pm 1\}$ such that $h(\boldsymbol{u}) = +1$ for every $\boldsymbol{u} \in X$ and $h(\boldsymbol{u}) = -1$ for every $\boldsymbol{u} \in Y$. (ii) In the other case, if $X \cap Y = \emptyset$ then there exists no such function and therefore the learning protocol must output "Error". Therefore, by Theorem 25, every such learning protocol must transmit at least $\Omega(d \log(n/d))$ bits. ■