

# Near-Optimal Linear Decision Trees for k-SUM and Related Problems

Daniel M. Kane<sup>\*</sup>  
University of California, San Diego  
La Jolla, CA, USA  
dakane@ucsd.edu

Shachar Lovett<sup>†</sup>  
University of California, San Diego  
La Jolla, CA, USA  
slovett@cs.ucsd.edu

Shay Moran<sup>‡</sup>  
Institute for Advanced Study  
Princeton, NJ, USA  
shaymoran1@gmail.com.

## ABSTRACT

We construct near optimal linear decision trees for a variety of decision problems in combinatorics and discrete geometry. For example, for any constant  $k$ , we construct linear decision trees that solve the  $k$ -SUM problem on  $n$  elements using  $O(n \log^2 n)$  linear queries. Moreover, the queries we use are comparison queries, which compare the sums of two  $k$ -subsets; when viewed as linear queries, comparison queries are  $2k$ -sparse and have only  $\{-1, 0, 1\}$  coefficients. We give similar constructions for sorting sumsets  $A + B$  and for solving the SUBSET-SUM problem, both with optimal number of queries, up to poly-logarithmic terms.

Our constructions are based on the notion of “inference dimension”, recently introduced by the authors in the context of active classification with comparison queries. This can be viewed as another contribution to the fruitful link between machine learning and discrete geometry, which goes back to the discovery of the VC dimension.

## CCS CONCEPTS

• Theory of computation → Computational complexity and cryptography;

## KEYWORDS

Linear decision tree, hyperplane arrangement, inference dimension

## ACM Reference Format:

Daniel M. Kane, Shachar Lovett, and Shay Moran. 2018. Near-Optimal Linear Decision Trees for k-SUM and Related Problems. In *Proceedings of 50th Annual ACM SIGACT Symposium on the Theory of Computing (STOC’18)*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3188745.3188770>

<sup>\*</sup>Supported by NSF CAREER Award ID 1553288 and a Sloan fellowship.

<sup>†</sup>Research supported by NSF CAREER award 1350481, CCF award 1614023 and a Sloan fellowship.

<sup>‡</sup>Research supported by the National Science Foundations and Simons Foundations.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

STOC’18, June 25–29, 2018, Los Angeles, CA, USA

© 2018 Copyright held by the owner/author(s). Publication rights licensed to the Association for Computing Machinery.

ACM ISBN 978-1-4503-5559-9/18/06...\$15.00

<https://doi.org/10.1145/3188745.3188770>

## 1 INTRODUCTION

This paper studies the linear decision tree complexity of several combinatorial problems, such as  $k$ -SUM, SUBSET-SUM, KNAPSACK, sorting sumsets, and more. A common feature these problems share is that they are all instances of the following fundamental problem in computational geometry.

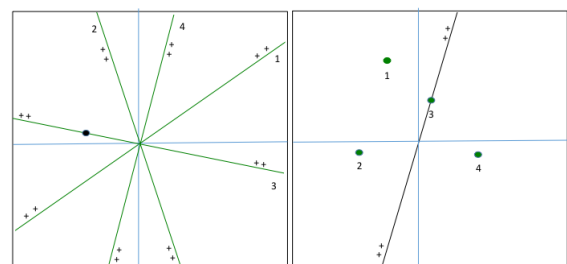
*The point-location problem.* Let  $H \subset \mathbb{R}^n$  be a finite set. Consider the problem in which given  $x \in \mathbb{R}^n$  as an input, the goal is to compute the function

$$\mathcal{A}_H(x) := (\text{sign}(\langle x, h \rangle) : h \in H) \in \{-, 0, +\}^H,$$

where  $\text{sign} : \mathbb{R} \rightarrow \{-, 0, +\}$  is the sign function and  $\langle \cdot, \cdot \rangle$  is the standard inner product in  $\mathbb{R}^n$ .

In discrete geometry this is known as the *point-location in an hyperplane-arrangement problem*, in which each  $h \in H$  is identified with the hyperplane orthogonal to  $h$ , and  $\mathcal{A}_H(x)$  corresponds to the cell in the partition induced by the hyperplanes in  $H$  to which the input point  $x$  belongs.

A dual formulation of this problem has been considered in learning theory, specifically within the context of active learning: here, each  $h \in H$  is thought of as a point,  $x$  is thought of as the learned half-space, and computing  $\mathcal{A}_H(x)$  corresponds to learning how each point  $h \in H$  is classified by  $x$ . In this work it will often be more intuitive to consider this dual formulation. See Figure 1 for a planar illustration of both interpretations.



**Figure 1: Primal and dual forms of the point-location problem.**  $H$  is the green lines/points, and  $x$  is the black point/line, and  $\mathcal{A}_H(x) = (+, +, 0, -)$

*Linear decision tree.* A linear decision tree for the point-location problem  $\mathcal{A}_H$  is an adaptive deterministic algorithm  $T$ . The set

$H \subset \mathbb{R}^n$  is known in advance, and the input is  $x \in \mathbb{R}^n$ . The algorithm does not have direct access to  $x$ . Instead, at each iteration the algorithm chooses some  $h \in \mathbb{R}^n$  and queries “ $\text{sign}(\langle h, x \rangle) = ?$ ” (note that  $h$  is not necessarily in  $H$ ). At the end, the algorithm should be able to compute  $\mathcal{A}_H(x)$  correctly. The *query complexity* is the maximum over  $x$  of the number of queries performed. Equivalently, such an algorithm can be described by a ternary decision tree which computes the sign of a linear query at each inner node. A query is  $s$ -sparse if it involves at most  $s$  nonzero coefficients. A linear decision tree is  $s$ -sparse if all its queries are  $s$ -sparse.

**Comparison decision tree.** A comparison decision tree for the point-location problem  $\mathcal{A}_H$  is a special type of a linear decision tree, where the only queries used are either of the form  $\text{sign}(\langle h, x \rangle)$  for  $h \in H$  (*label queries*), or  $\text{sign}(\langle h' - h'', x \rangle)$  for  $h', h'' \in H$  (*comparison queries*). Note that  $\langle h' - h'', x \rangle \geq 0$  if and only if  $\langle h', x \rangle \geq \langle h'', x \rangle$ , which is why we call these comparison queries. In the dual version (in which we view  $H$  as a set of points), comparison queries have a natural geometric interpretation: assuming that  $\text{sign}(\langle h', x \rangle) = \text{sign}(\langle h'', x \rangle)$ , a comparison query  $\langle h' - h'', x \rangle$ , corresponds to querying which one of  $h', h'' \in H$  is further from the hyperplane defined by  $x$ . Observe that if all elements  $h \in H$  are  $s$ -sparse then a comparison decision tree is  $2s$ -sparse.

## 1.1 Results

Our main result is a method that produces near optimal decision trees for many natural and well studied combinatorial instances for the point-location problems by using comparison decision trees. We first describe a few concrete instances, and then the general framework.

**1.1.1  $k$ -SUM.** In the  $k$ -SUM problem an input array  $x \in \mathbb{R}^n$  of  $n$  numbers is given, and the goal is to decide whether the sum of  $k$  distinct numbers is 0. This problem (in particular 3-SUM) has been extensively studied since the 1990s, as it embeds into many problems in computational geometry, see for example [G095]. More recently, it has also been studied in the context of fine-grained complexity, see for example the survey [VW15].

The  $k$ -SUM problem corresponds to the following point-location problem. Let  $H \subseteq \{0, 1\}^n$  denote all vectors of hamming weight  $k$ . Thus,  $x \in \mathbb{R}^n$  contains  $k$  numbers whose sum is 0 if and only if  $\mathcal{A}_H(x)$  contains at least one 0 entry.

In this context, comparison decision trees allow for two types of linear queries: label queries of the form “ $\sum_{i \in I} x_i \geq 0$ ?” where  $I \subset [n]$  has size  $|I| = k$ , and comparison queries of the form “ $\sum_{i \in I} x_i \geq \sum_{j \in J} x_j$ ?” where  $I, J \subset [n]$  have size  $|I| = |J| = k$ .

**THEOREM 1.1.** *The  $k$ -SUM problem on  $n$  elements can be computed by a comparison decision tree of depth  $O(kn \log^2 n)$ . In particular, all the queries are  $2k$ -sparse and have only  $\{-1, 0, 1\}$  coefficients.*

This improves a series of works. There is a basic deterministic algorithm in time  $\tilde{O}(n^{\lceil k/2 \rceil})$  that first sorts all sums of  $\lfloor k/2 \rfloor$  input elements, and then traverses the sorted list in a way that enables to decide the  $k$ -SUM problem. It can be transformed to a linear decision tree with the same number of queries, which in our language are label and comparison queries with sparsity  $k$ . Erickson [Eri99] showed that  $\Omega(n^{\lceil k/2 \rceil})$  queries are indeed necessary to solve  $k$ -SUM if only such queries are allowed (or more generally, if only

$k$ -sparse linear queries are allowed). Ailon and Chazelle [AC05] extended this work by giving lower bounds of the form  $n^{1+\Omega(1)}$  for sparsity between  $k$  and  $k + k^{o(1)}$ . Our construction has sparsity  $2k$ . It remains open to understand the linear decision depth needed for sparsity up to  $2k$ .

In a breakthrough work, Grønlund and Pettie [GP14] were the first to break the  $n^{\lceil k/2 \rceil}$  bound. They constructed a randomized  $(2k-2)$ -linear decision tree for  $k$ -SUM which makes  $O(n^{k/2} \sqrt{\log n})$  queries. This was improved to  $O(n^{k/2})$  by Gold and Sharir [GS17].

In the general linear decision tree model, without any sparsity assumptions, a series of works in discrete geometry have designed linear decision trees for the general point-location problem. In the context of  $k$ -SUM, the best result is of Ezra and Sharir [ES17], who constructed a linear decision tree of depth  $O(n^2 \log^2 n)$  for any constant  $k$ . This improves on previous results of Meyer auf der Heide [MadH84], Meiser [Mei93] and Cardinal et al. [CIO16].

**1.1.2 Sorting  $A + B$ .** Let  $A, B \subset \mathbb{R}$  be sets of size  $|A| = |B| = n$ . Their sumset, denoted by  $A + B$  is the set  $\{a + b : a \in A, b \in B\}$ . Consider the goal of sorting  $A + B$  while minimizing the number of comparisons (here, by comparisons we mean the usual notion in sorting, that is comparing two elements of  $A + B$ ). While it is possible that  $|A + B| = n^2$ , it is well known that the number of possible orderings of  $A + B$  is only  $n^{O(n)}$  [Fre76a]. Thus, from an information theoretic perspective it is conceivable that  $A + B$  can be sorted using only  $O(n \log n)$  comparisons. However, Fredman [Fre76a] gave a tight bound of  $\Theta(n^2)$  on the number of comparisons needed to sort  $A + B$ .

It is natural to ask whether enabling the algorithm more access to the data in the form of simple local queries can achieve  $o(n^2)$  query-complexity. We show that if the algorithm can use *differences-comparisons* then an almost optimal query-complexity of  $O(n \log^2 n)$  suffices to sort  $A + B$ . A differences-comparison on an array  $[x_1, \dots, x_n]$  is a query of the form

$$“x_i - x_j \geq x_k - x_l ?”;$$

in words: “is  $x_i$  greater than  $x_j$  more than  $x_k$  is greater than  $x_l$ ?”.

The problem of sorting  $A + B$  corresponds to the following point-location problem. Let  $A = \{a_1, \dots, a_n\}, B = \{b_1, \dots, b_n\}$  and identify  $x \in \mathbb{R}^{2n}$  with  $x = (a_1, \dots, a_n, b_1, \dots, b_n)$ . Let  $H \subset \{-1, 0, 1\}^{2n}$  consist of vectors with exactly one 1 and one  $-1$  in the first  $n$  elements, and exactly one 1 and one  $-1$  in the last  $n$  elements. Then computing  $\mathcal{A}_H(x)$  corresponds to answering all queries of the form “ $a_i + b_j \geq a_k + b_l$ ?” for all  $i, j, k, l \in [n]$ , which amounts to sorting  $A + B$ . In this context, the two types of queries used by comparison decision trees are comparison queries in  $A + B$ , namely “ $a_i + b_j \geq a_k + b_l$ ?” where  $i, j, k, l \in [n]$  (which correspond to the label queries in the point location problem), and differences-comparison queries in  $A + B$ , namely “ $a_i + b_j - a_{i'} - b_{j'} \geq a_k + b_l - a_{k'} - b_{l'}$ ?” where  $i, j, k, l, i', j', k', l' \in [n]$  (which correspond to comparison queries in the point location problem).

**THEOREM 1.2.** *Given  $A, B \subset \mathbb{R}$  of size  $|A| = |B| = n$ , their sumset  $A + B$  can be sorted by a comparison decision tree of depth  $O(n \log^2 n)$ . In particular, all queries are 8-sparse with  $\{-1, 0, 1\}$  coefficients.*

The problem of sorting sumsets has been considered by Fredman [Fre76a], who showed that if only comparison queries are

allowed, then  $\Theta(n^2)$  queries are sufficient and necessary to sort  $A + B$ . Grønlund and Pettie [GP14] use it in their work, and specifically ask for a better linear decision tree for sorting sumsets.

**1.1.3 NP-hard Problems.** Several NP-hard problems can be phrased as point-location problems. For example, the *SUBSET-SUM* problem is to decide, given a set  $A$  of  $n$  real numbers, whether there exists a subset of  $A$  whose sum is 0. The *KNAPSACK* problem is to decide whether there exists a subset of  $A$  whose sum is 1. We focus here on *SUBSET-SUM* for concreteness.

The *SUBSET-SUM* problem corresponds to the following point-location problem. Let  $A = \{a_1, \dots, a_n\}$  and take  $x = (a_1, \dots, a_n) \in \mathbb{R}^n$ . Let  $H = \{0, 1\}^n \setminus \{0^n\}$ . Then  $A$  has a subset whose sum is 0 if and only if  $\mathcal{A}_H(x)$  contains at least one 0.

In this context, comparison decision trees have two types of queries: label queries of the form “ $\sum_{i \in A'} a_i \geq 0$ ?” for some  $A' \subseteq A$ , and comparison queries of the form “ $\sum_{i \in A'} a_i \geq \sum_{i \in A''} a_i$ ?” for some  $A', A'' \subseteq A$ .

**THEOREM 1.3.** *The SUBSET-SUM problem can be solved using a comparison decision tree of depth  $O(n^2 \log n)$ , where  $n$  is the size of the input-set. In particular, all the queries are linear with  $\{-1, 0, 1\}$  coefficients.*

Note that the bound is tight up to the log factor: indeed, in the corresponding point-location problem,  $H = \{0, 1\}^n$ , and thus  $\{\mathcal{A}_H(x) : x \in \mathbb{R}^n\}$  corresponds to the family of thresholds function on the boolean cube. It is well known that the number of such functions is  $2^{\Theta(n^2)}$  [GT62], and thus any decision tree (even one that uses arbitrary queries, each with a constant number of possible answers) that computes  $\mathcal{A}_H(x)$  must use at least  $\Omega(n^2)$  queries.

The surprising fact that *SUBSET-SUM*, an NP-hard problem, has a polynomial time algorithm in a nonuniform model (namely, linear decision trees) was first discovered by Meyer auf der Heide [MadH84], answering an open problem posed by Dobkin and Lipton [DL74] and Yao [Yao81]. It originally required  $O(n^4 \log n)$  linear queries. It was generalized by Meiser [Mei93] to the general point-location problem, and later improved by Cardinal [CIO16] and Ezra and Sharir [ES17]. This last work, although it does not address the *SUBSET-SUM* directly, seems to improve the number of queries to  $O(n^3 \log^2 n)$ . Observe that our construction gives a near-optimal number of linear queries, namely  $O(n^2 \log n)$ . Moreover, the queries are simple, in the sense that they involve only  $\{-1, 0, 1\}$  coefficients, and natural from a computational perspective as they only compare the sums of subsets. This is unlike the previous works mentioned, which requires arbitrary coefficients due to the geometric nature of their techniques.

**1.1.4 Other Applications.** Our framework (see Corollary 1.9) is pretty generic, and as such gives near optimal linear decision trees for a host of problems considered in the literature. For example, the following problems were considered in [GP14]. We discuss each one briefly, and refer the interested reader to [GP14] for a deeper discussion.

**$k$ -LDT.** Given a fixed linear equation  $\phi(x_1, \dots, x_k) = \alpha_0 + \sum_{i=1}^k \alpha_i x_i$  and a set  $A \subset \mathbb{R}$  of size  $|A| = n$ , the goal is to decide if there exist distinct  $a_1, \dots, a_k \in A$  such that  $\phi(a_1, \dots, a_k) = 0$ .

This problem is a variant of the  $k$ -SUM problem, and can be embedded as a point-location problem in  $\mathbb{R}^{nk+1}$  as follows. Let  $x = (1, \alpha_1 a_1, \dots, \alpha_1 a_n, \dots, \alpha_k a_1, \dots, \alpha_k a_n)$  and  $H \subset \{-1, 0, 1\}^{nk+1}$  consists of  $h$  which have a “ $-1$ ” in their first coordinate, a single “ $+1$ ” in each of the  $k$  blocks of size  $n$ , and 0 elsewhere. Corollary 1.9 implies a comparison decision tree with  $O(k^2 n \log^2 n)$  queries which are  $(2k+2)$ -sparse and with  $\{-1, 0, 1\}$  coefficients. For constant  $k$  this gives  $O(n \log^2 n)$ , which improves upon the previous best bound of  $O(n^2 \log^2 n)$  of [ES17].

**Zero triangles.** Let  $G = (V, E)$  be a graph on  $|V| = n$  vertices and  $|E| = m$  edges, which is known in advance (it is not part of the input). The inputs are edge weights  $x : E \rightarrow \mathbb{R}$ . The goal is to decide if there is a triangle in  $G$  whose sum is zero. This problem clearly embeds as a point-location problem in  $\mathbb{R}^m$ . Corollary 1.9 gives a comparison decision tree which solves this problem with  $O(m \log^2 m)$  queries. All the queries are 6-sparse and have  $\{-1, 0, 1\}$  coefficients. This improves upon the previous bound of  $O(m^{5/4})$  of [GP14].

**All pairs shortest path.** The All-Pairs-Shortest-Path (APSP) problem is to compute, given a complete  $n$ -vertex graph with non-negative weights on the edges, what is the shortest path between all pairs of nodes. It is known that the complexity of this problem is asymptotically equivalent to that of computing a  $(\min, +)$  matrix product. Here,  $A, B$  are  $n \times n$  matrices with non-negative entries, and the goal is to compute their  $(\min, +)$  product  $C$ , which is the  $n \times n$  matrix given by  $C_{i,j} = \min_k A_{i,k} + B_{k,j}$ . All known algorithms that compute  $C$  run in worst-case time  $n^{3-o(1)}$ , where the best bounds are by Williams [Wil14]. In an influential paper in 1976, Fredman [Fre76b] constructed a linear decision tree of depth  $O(n^{2.5})$  that solves the problem. Pettie [Pet02] improved this bound to  $O(mn \log \alpha)$  ( $m$  is the number of edges), where  $\alpha(m, n)$  is the inverse-Ackermann function. Corollary 1.9 gives a comparison decision tree which solves this problem with  $O(n^2 \log^2 n)$  queries, which is near-optimal as the input size is  $2n^2$ . Moreover, all the queries are either label queries of the form  $A_{i,k} + B_{k,j} - A_{i,k'} - B_{k',j}$  or comparison queries of the form  $(A_{i,k} + B_{k,j} - A_{i,k'} - B_{k',j}) - (A_{i',k''} + B_{k'',j'} - A_{i'',k'''} - B_{k''',j'})$ .

## 1.2 General Framework

Our results are based on the notion of “inference dimension”, which was recently introduced by the authors [KLMZ17] in the context of active learning.

**DEFINITION 1.4 (INFERENCE).** *Let  $S \subset \mathbb{R}^n$  and  $h, x \in \mathbb{R}^n$ . We say that  $S$  infers  $h$  at  $x$  if “ $\text{sign}(\langle h, x \rangle)$ ” is determined by the answers to the label and comparison queries on  $S$ . That is, if we set*

$$P_S(x) := \{x' \in \mathbb{R}^n : \mathcal{A}_{S \cup \{x\}}(x') = \mathcal{A}_{S \cup \{x\}}(x)\}$$

*then  $\text{sign}(\langle x', h \rangle) = \text{sign}(\langle x, h \rangle)$  for all  $x' \in P_S(x)$ . We further define the inference set of  $S$  at  $x$  to be*

$$\text{infer}(S, x) := \{h \in \mathbb{R}^n : S \text{ infers } h \text{ at } x\}.$$

*For each  $h \in \text{infer}(S, x)$ , we refer to  $\text{sign}(\langle h, x \rangle)$  as the inferred value of  $h$  at  $x$ .*

An equivalent geometric condition to “ $S$  infers  $h$  at  $x$ ” is that the hyperplane defined by  $h$  is either disjoint from  $P_S(x)$  or contains  $P_S(x)$ .

For example, if  $h_1, h_2$  are such that  $\text{sign}(\langle h_1, x \rangle) = \text{sign}(\langle h_2, x \rangle) = 0$ , and  $h$  is in the linear space spanned by  $h_1, h_2$  then  $\text{sign}(\langle h, x \rangle) = 0$  and so  $\{h_1, h_2\}$  infer  $h$  at  $x$ . Similarly, if  $\text{sign}(\langle h_1, x \rangle) = \text{sign}(\langle h_2 - h_1, x \rangle) = +1$ , and  $h$  is in the cone spanned by  $h_1, h_2 - h_1$  (i.e.  $h = \alpha h_1 + \beta(h_2 - h_1)$  for  $\alpha, \beta > 0$ ) then  $\text{sign}(\langle h, x \rangle) = +1$  and so  $\{h_1, h_2\}$  infer  $h$  at  $x$ .

**DEFINITION 1.5 (INFERENCE DIMENSION).** Let  $H \subset \mathbb{R}^n$ . The inference dimension of  $H$  is the minimal  $d \geq 1$  for which the following holds. For any subset  $S \subset H$  of size  $|S| \geq d$ , and for any  $x \in \mathbb{R}^n$ , there exists  $h \in S$  such that  $S \setminus \{h\}$  infers  $h$  at  $x$ .

We refer the reader to [KLMZ17] for some simple examples and further discussion regarding the inference dimension.

The first step in the proof of Theorem 1.1, Theorem 1.2 and Theorem 1.3, is to show that the sets  $H$  in the corresponding point location problems are of low inference dimension. The following general theorem provides a uniform treatment for this.

For  $h \in \mathbb{Z}^n$  defines its  $\ell_1$  norm as  $\|h\|_1 = \sum_{i=1}^n |h_i|$ .

**THEOREM 1.6.** The inference dimension of  $H = \{h \in \mathbb{Z}^n : \|h\|_1 \leq w\}$  is  $d = O(n \log w)$ .

The second step is to show that sets of low inference dimension have efficient comparison decision trees. First, we show this for zero-error randomized comparison decision trees. A zero-error randomized comparison decision tree is a distribution over (deterministic) comparison decision trees  $T$ , each solves  $\mathcal{A}_H(x)$  correctly for all inputs. The expected query complexity is the maximum over  $x$ , of the expected number of queries performed by  $T(x)$  to compute  $\mathcal{A}_H(x)$ .

**THEOREM 1.7.** Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Then there exists a zero-error randomized comparison decision tree which computes  $\mathcal{A}_H$ , whose expected query complexity is  $O((d + n \log d) \log |H|)$ .

A slightly weaker version of Theorem 1.7 appears in [KLMZ17] (see Theorem 4.1 there). The third and last step is to de-randomize Theorem 1.7 and obtain a deterministic comparison decision tree.

**THEOREM 1.8.** Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Then there exists a comparison decision tree which computes  $\mathcal{A}_H$ , whose query complexity is  $O((d + n \log(nd)) \log |H|)$ .

The proof of Theorem 1.8 uses a double-sampling argument, a technique originated in the study of uniform convergence bounds in statistical learning theory [VC71]. The following corollary summarizes the above theorems concisely. For  $h \in \mathbb{Z}^n$  define  $\|h\|_\infty = \max |h_i|$ .

**COROLLARY 1.9.** Let  $H \subset \mathbb{Z}^n$  be such that  $\|h\|_\infty \leq w$  for all  $h \in H$ . Then there exists a comparison decision tree computing  $\mathcal{A}_H$  whose query complexity is  $O(n \log(nw) \log |H|)$ .

**PROOF.** Observe that  $\|h\|_1 \leq n\|h\|_\infty \leq nw$ . By Theorem 1.6, the inference dimension of  $H$  is  $d = O(n \log(nw))$ . The corollary now follows from Theorem 1.8.  $\square$

One can now verify that Theorem 1.1, Theorem 1.2, and Theorem 1.3 follow from Corollary 1.9 by setting  $w = 1$ .

### 1.3 Proof Overview

**Theorem 1.6 proof overview.** We need to show that if  $S \subseteq H$  is sufficiently large (at least some  $O(n \log w)$ ), then there is some  $h \in S$  whose label  $\text{sign}(\langle h, x \rangle)$  can be inferred from the comparison and label queries on  $S \setminus \{h\}$ . We derive this using two simple kinds of inference rules:

- (i) Say we already inferred for  $h_1, h_2$  that  $\langle h_1, x \rangle = \langle h_2, x \rangle = 0$ , and  $h$  is a linear combination of  $h_1, h_2$ . Then infer that  $\langle h, x \rangle = 0$ .
- (ii) Say we already inferred that  $\langle h_1, x \rangle > 0$  and that  $\langle h_3, x \rangle \geq \langle h_2, x \rangle$  (using a comparison query), and  $h = h_1 + (h_3 - h_2)$ . Then infer that  $\langle h, x \rangle > 0$  (similarly when  $\langle h_1, x \rangle < 0$ ).

The challenging part is to show that one of these rules must apply whenever  $S \subseteq H$  is sufficiently large (at least some  $O(n \log w)$ ); the easier case is when there are more than  $n$  different  $h_i \in S$  with  $\text{sign}(\langle h_i, x \rangle) = 0$ : here we can apply rule (i), since one of the  $h_i$ 's must be in the linear span of the others. The other case, is when there are at least some  $O(n \log w)$  different  $h_i \in S$  having the same  $\text{sign}(\langle h_i, x \rangle)$  that is not 0. Here we crucially rely on comparison queries in the following way: assume that the  $h_i$ 's in  $S$  are ordered such that  $0 < \langle h_1, x \rangle \leq \langle h_2, x \rangle \leq \dots$ . We show that for some  $i^*$ ,  $h_{i^*}$  can be expressed as  $h_1 + \sum_{i < j} \alpha_{i,j} (h_j - h_i)$ , where  $\alpha_{i,j} \geq 0$ , and  $h_{i^*}$  does not participate in the summation. Note that from this one can infer  $\text{sign}(\langle h_{i^*}, x \rangle)$  using inference rule (ii). To show that such an  $i^*$  exists, it turns out (by a simple calculation) that it suffices to show that there are two distinct boolean combinations of the  $(h_{i+1} - h_i)$ 's that yield the same vector. The existence of such boolean combinations is derived via a pigeon-hole argument, using the assumption that the  $h_i$ 's are short integral vectors.

**Theorem 1.7 proof overview.** Theorem 1.7 is derived by a fairly simple randomized algorithm that repeats the following procedure until all labels in  $H$  are inferred: sample  $2d$  points from the set of points whose labels are still not known. Then, query their labels and sort them (using comparison queries) according to the value of their inner product with the target point  $x$ . Lastly, using the two inferences rules stated above, infer all possible labels of the other points in  $H$ , and remove all labeled points.

The correctness of this algorithm follows by showing that at each application of the procedure, half of the remaining labels are inferred, in expectation (Lemma 4.2). This is derived by a short symmetrization argument, and basic properties of the Inference dimension (Claim 4.1).

**Theorem 1.8 proof overview.** In the last step, we show that the above randomized algorithm can be derandomized; we show that instead of picking  $2d$  points at random in each application of the procedure, we can deterministically pick a slightly larger subset (of size  $O(d + n \log d)$ ) with the same inference guarantee. The existence of such a subset is derived via a probabilistic argument that utilizes the double sampling argument from [VC71].



## 1.4 Further Research

We prove that many combinatorial point-location problems have near optimal linear decision trees. Moreover, these are comparison decision trees, in which the linear queries are particularly simple: both sparse (in many cases) and have only  $\{-1, 0, 1\}$  coefficients. This raises the possibility of having improved algorithms for these problems in other models of computations. To be concrete, we focus on 3-SUM below, but the same questions can be asked for any other problem of a similar flavor.

*Uniform computation.* The most obvious question is whether the existence of a near optimal linear decision tree implies anything about uniform computation. As showed in [GP14], this can lead to log-factor savings. It is very interesting whether greater savings can be achieved. We do not discuss this further here, as this question has been extensively discussed in the literature (see e.g. [VW15]).

*Nondeterministic computation.* Let  $A \subset \mathbb{R}$  be a set of size  $|A| = n$ . It is very easy to “prove” that  $A$  is a positive instance of 3-SUM, by demonstrating three elements whose sum is zero. However, it is much less obvious how to prove that  $A$  is a negative instance of 3-SUM. This problem was explicitly studied in [CGI<sup>+</sup>16] in the context of nondeterministic ETH. They constructed such a proof which can be verified in time  $O(n^{3/2})$ . It seems plausible that our current approach may lead to improved bounds. Thus, we propose the following problem.

**OPEN PROBLEM 1.10.** *Given a set of  $n$  real numbers no three of which sums to 0. Is there a proof of that fact which can be verified in near-linear time?*

*3-SUM with preprocessing.* Let  $A \subset \mathbb{R}$  of size  $|A| = n$ . The 3-SUM with preprocessing problem allows one to preprocess the set  $A$  in quadratic time. Then, given any subset  $A' \subset A$ , the goal is to solve that 3-SUM problem on  $A'$  in time significantly faster than  $n^2$ . Chan and Lewenstein [CL15] designed such an algorithm, which solves that 3-SUM problem on any subset in time  $O(n^{2-\varepsilon})$  for some small constant  $\varepsilon > 0$ . It is interesting whether our techniques can help improve this to near-linear time.

**OPEN PROBLEM 1.11.** *Given a set of  $n$  real numbers, can they be preprocessed in  $O(n^2)$  time, such that later on, for every subset of the numbers the 3-SUM problem can be solved in time near-linear in  $n$ ?*

*General point-location problem.* It is natural to ask whether the techniques used in this paper, and in particular, the inference-dimension, can be used to improve the state-of-the-art upper bounds for general point location problems. Unfortunately, unless the set of hyperplanes  $H$  has some combinatorial structure, its inference dimension may be unbounded: in [KLMZ17] we construct examples of  $H \subset \mathbb{R}^3$  whose inference dimension is unbounded. Nevertheless, we conjecture that by generalizing comparison queries (which are  $\pm 1$  linear combinations of two elements in  $H$ ) to arbitrary linear combinations of two elements from  $H$  might solve the problem.

**CONJECTURE 1.12.** *Let  $H \subset \mathbb{R}^n$ . There exists a linear decision tree which computes  $\mathcal{A}_H$  of depth  $O(n \log |H|)$ . Moreover, all the linear queries are in  $\{\alpha h' + \beta h'' : \alpha, \beta \in \mathbb{R}, h', h'' \in H\}$ .*

*High dimensional gadgets.* Another possibility of circumventing the examples of  $H \subseteq \mathbb{R}^3$  with unbounded inference dimension is

to find stronger *gadgets* than comparisons. Define a gadget with arity  $t$  on  $\mathbb{R}^3$  as a function  $g : (\mathbb{R}^3)^t \rightarrow \mathbb{R}^3$ . Given  $H \subseteq \mathbb{R}^3$  and such a gadget  $g$ , a gadget query is  $\text{sign}(g(h_1, \dots, h_t), x) = ?$  for  $h_i \in H$ . For example a comparison query is a gadget with arity 2 defined by  $g(h_1, h_2) = h_1 - h_2$ .

Is there a fixed gadget such that for every  $H \subseteq \mathbb{R}^3$  there exists an LDT of depth  $O(\log |H|)$  that only uses label queries and gadget queries? (in [KLMZ17] it is shown that for  $d = 2$  the comparison gadget satisfies it.) Clearly, if the answer is yes for  $d = 3$ , it is interesting to generalize the construction for general dimension  $d$ .

*Optimal bounds.* We suspect that our analysis can be sharpened to improve the log-factors that separate it from the information theoretical lower bounds. For concreteness, we pose the following conjecture.

**CONJECTURE 1.13.** *For any  $H \subset \{-1, 0, 1\}^n$  there exists a comparison decision tree which computes  $\mathcal{A}_H$  with  $O(n \log |H|)$  many queries. In particular,*

- 3-SUM on  $n$  real numbers can be solved by a 6-sparse linear decision tree which makes  $O(n \log n)$  queries.
- Sorting  $A + B$ , where  $A, B$  are sets of  $n$  real numbers, can be solved by a 8-sparse linear decision tree which makes  $O(n \log n)$  queries.
- SUBSET-SUM on  $n$  real numbers can be solved by a linear decision tree which makes  $O(n^2)$  queries.

Note that Corollary 1.9 gives a bound, which is off by a log  $n$  factor for these problems.

*Acknowledgements.* We thank the Simons institute at Berkeley, where this work was performed, for their hospitality.

*Organization.* We begin with some preliminaries in Section 2. Theorem 1.6 is proved in Section 3. Theorem 1.7 is proved in Section 4. Theorem 1.8 is proved in Section 5.

## 2 PRELIMINARIES

Let  $H \subseteq \mathbb{R}^n$  be a finite set. For every  $x \in \mathbb{R}^n$ ,  $\mathcal{A}_H(x)$  denotes the function

$$\mathcal{A}_H(x) := (\text{sign}(\langle x, h \rangle) : h \in H) \in \{-1, 0, +1\}^H,$$

where  $\text{sign} : \mathbb{R} \rightarrow \{-1, 0, +1\}$  is the sign function and  $\langle \cdot, \cdot \rangle$  is the standard inner product in  $\mathbb{R}^n$ . The following lemma is a variant of standard bounds on the number of cells in a hyperplane arrangement.

**LEMMA 2.1.** *Let  $H \subset \mathbb{R}^n$  be a set of size  $|H| = m$ . Then  $|\{\mathcal{A}_H(x) : x \in \mathbb{R}^n\}| \leq (2em)^n$ .*

**PROOF.** It is well known that a set of  $m$  hyperplanes partitions  $\mathbb{R}^n$  to at most  $\binom{m}{\leq n}$  open cells. The lemma follows by first choosing  $i \leq n$  linearly independent hyperplanes to which  $x$  belongs, and then applying the above bound to the remaining ones (restricted to

a subspace of dimension  $n - i$ ). Thus

$$\begin{aligned} |\{\mathcal{A}_H(x) : x \in \mathbb{R}^n\}| &\leq \sum_{i=0}^n \binom{m}{i} \binom{m-i}{\leq n-i} = \sum_{i=0}^n \sum_{j=0}^{n-i} \binom{m}{i} \binom{m-i}{j} \\ &= \sum_{s=0}^n \sum_{i=0}^s \binom{m}{s} \binom{s}{i} = \sum_{s=0}^n \binom{m}{s} 2^s \\ &\leq \binom{m}{\leq n} 2^n \leq (em/n)^n \leq (em)^n, \end{aligned}$$

where the second equality follows from the identity  $\binom{m}{i} \binom{m-i}{j} = \binom{m}{s} \binom{s}{i}$ , where  $s = i + j$ , and the last inequality follows from the well known upper bound  $\binom{m}{\leq n} \leq (em/n)^n \leq (em)^n$ .  $\square$

### 3 BOUNDING THE INFERENCE DIMENSION

We prove Theorem 1.6 in this section.

**Theorem 1.6 (restated).** *The inference dimension of  $H = \{h \in \mathbb{Z}^n : \|h\|_1 \leq w\}$  is  $d = O(n \log w)$ .*

Let  $S \subset \mathbb{Z}^n$  be such that  $\|h\|_1 \leq w$  for all  $h \in S$ . We assume  $|S| = d$  where  $d$  is large enough to be determined later. Fix  $x \in \mathbb{R}^n$ . We will show that there exists  $h \in S$  such that  $S \setminus \{h\}$  infers  $h$  at  $x$ .

Partition  $S$  into  $\{S_b : b \in \{-, 0, +\}\}$ , where

$$S_b := \{h \in S : \text{sign}(\langle h, x \rangle) = b\}.$$

We will show that if  $S$  is sufficiently large then  $S_b \setminus \{h\}$  infers  $h$  at  $x$  for some  $s \in S_b$  and  $b \in \{-, 0, +\}$ . The simplest case is when  $S_0$  is large:

**CLAIM 3.1.** *If  $|S_0| > n$  then there exists  $h \in S_0$  such that  $S_0 \setminus \{h\}$  infers  $h$  at  $x$ . In particular,  $S \setminus \{h\}$  infers  $h$  at  $x$ .*

**PROOF.** Let  $h_1, \dots, h_{n+1} \in S_0$  be distinct elements such that  $h_{n+1}$  belongs to the linear span of  $h_1, \dots, h_n$ . We claim that  $\{h_1, \dots, h_n\}$  infer  $h_{n+1}$  at  $x$ . More specifically, we claim that having

- (i)  $\text{sign}(\langle h_i, x \rangle) = 0$  for  $i \leq n$ , and
- (ii)  $h_{n+1} \in \text{span}\{h_i : i \leq n\}$

imply that  $\text{sign}(\langle h_{n+1}, x \rangle) = 0$ . Indeed, by (ii) there exist coefficients  $\alpha_i$ 's such that  $h_{n+1} = \sum_{i=1}^n \alpha_i h_i$ , and therefore, using (i), it follows that  $\langle h_{n+1}, x \rangle = \langle \sum_{i=1}^n \alpha_i h_i, x \rangle = \sum_{i=1}^n \alpha_i \langle h_i, x \rangle = 0$ .  $\square$

Thus, we assume from now on that  $|S_0| \leq n$ . We assume without loss of generality that  $|S_+| \geq |S_-|$ , and show that there is some  $h \in S_+$  such that  $S_+ \setminus \{h\}$  infers  $h$  at  $x$ . The other case is analogous. Set  $m = \lfloor (d - n)/2 \rfloor$  and let  $h_1, \dots, h_m \in S_+$  sorted by

$$0 < \langle h_1, x \rangle \leq \dots \leq \langle h_m, x \rangle.$$

The idea is to show that some  $h_i$  satisfies that  $h_i - h_1$  is in the cone spanned by the  $h_k - h_l$  where  $1 \leq l \leq k < i$ . Then, a simple argument shows that  $S_+ \setminus \{h_i\}$  infers  $h_i$  at  $x$ . The existence of such an  $h_i$  is derived by a counting argument that boils down to the following lemma.

**CLAIM 3.2.** *Assume that  $2^{m-1} > \left(\frac{2e(2w+1)m}{n}\right)^n$ . Then there exist  $\alpha_1, \dots, \alpha_{m-1} \in \{-1, 0, 1\}$ , not all zero, such that*

$$\sum_{i=1}^{m-1} \alpha_i (h_{i+1} - h_i) = 0.$$

*In particular, this holds for  $m = O(n \log w)$  with a large enough constant.*

**PROOF.** For any  $\beta \in \{0, 1\}^{m-1}$  define  $f(\beta) := \sum \beta_i (h_{i+1} - h_i)$ . Note that  $f(\beta) \in \mathbb{Z}^n$ , and as since  $\|h_i\|_1 \leq w$  for all  $i$ , it follows that  $\|f(\beta)\|_1 \leq 2w(m-1)$  by the triangle inequality. Let  $F := \{f(\beta) : \beta \in \{0, 1\}^{m-1}\}$ . Next, we bound  $|F|$ . We claim that

$$|F| \leq 2^n \binom{2w(m-1) + n}{n}.$$

To see that, note that there are  $2^n$  possible signs for each  $f \in F$ . The number of patterns for the absolute values is at most the number of ways to express  $2w(m-1)$  as the sum of  $n+1$  nonnegative integers. Equivalently, it is the number of ways of placing  $2w(m-1)$  balls in  $n+1$  bins, which is  $\binom{2w(m-1) + n}{n}$ . We further simplify

$$|F| \leq 2^n \binom{2w(m-1) + n}{n} \leq 2^n \binom{2w+1}{n} m^n \leq \left(\frac{2e(2w+1)m}{n}\right)^n.$$

By our assumptions  $2^{m-1} > |F|$ . Thus by the pigeonhole principle there exist distinct  $\beta', \beta''$  for which  $f(\beta') = f(\beta'')$ . The claim follows for  $\alpha = \beta' - \beta''$ .  $\square$

We assume that  $d = O(n \log w)$  with a large enough constant, so that the conditions of Claim 3.2 hold. Let  $\alpha_1, \dots, \alpha_{m-1} \in \{-1, 0, 1\}$ , not all zero, be such that  $\sum \alpha_i (h_{i+1} - h_i) = 0$ . Let  $1 \leq p \leq m-1$  be maximal such that  $\alpha_p \neq 0$ . We may assume that  $\alpha_p = -1$ , as otherwise we can negate all of  $\alpha_1, \dots, \alpha_{m-1}$ .

Adding  $h_{p+1} - h_1 = \sum_{i=1}^p (h_{i+1} - h_i)$  to 0 =  $\sum \alpha_i (h_{i+1} - h_i)$ , we obtain that

$$h_{p+1} - h_1 = \sum_{i=1}^p (\alpha_i + 1)(h_{i+1} - h_i) = \sum_{i=1}^{p-1} (\alpha_i + 1)(h_{i+1} - h_i),$$

where the first equality holds as  $\alpha_i = 0$  if  $i > p$ , and the second equality holds as  $\alpha_p = -1$ .

We claim that  $R = \{h_1, \dots, h_p\}$  infers  $h_{p+1}$  at  $x$ , which completes the proof. More specifically, we claim that having

- (i)  $0 < \langle h_1, x \rangle \leq \dots \leq \langle h_p, x \rangle$ ,
- (ii)  $h_{p+1} - h_1 = \sum_{i=1}^{p-1} (\alpha_i + 1)(h_{i+1} - h_i)$ , where the coefficients  $\alpha_i + 1 \geq 0$  for all  $i$ ,

imply that  $\text{sign}(\langle h_{p+1}, x \rangle) \geq 0$ . Indeed, item (i) implies that  $\langle x, h_i - h_j \rangle \geq 0$ , for every  $1 \leq j < i \leq p$ , and item (ii) implies that  $h_{p+1} - h_1$  is in the cone spanned by  $h_i - h_j$  for  $1 \leq j < i \leq p$ . Thus, also  $\langle x, h_{p+1} - h_1 \rangle \geq 0$ , which implies, by the left-most inequality of item (ii), that  $\langle x, h_{p+1} \rangle \geq \langle x, h_1 \rangle > 0$ , as required.

### 4 ZERO-ERROR RANDOMIZED COMPARISON DECISION TREE

We prove Theorem 1.7 in this section.

**Theorem 1.7 (restated).** *Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Then there exists a zero-error randomized comparison decision tree which computes  $\mathcal{A}_H$ , whose expected query complexity is  $O((d + n \log d) \log |H|)$ .*

We begin with the following claim. Recall that  $\text{infer}(S, x)$  is the set of  $h \in \mathbb{R}^n$  which can be inferred from  $S$  at  $x$ .

CLAIM 4.1. Let  $S \subset \mathbb{R}^n$  with inference dimension  $d$  and  $|S| = d + m$ . Then for every  $x \in \mathbb{R}^n$ , there exist  $h_1, \dots, h_m \in S$  such that

$$h_i \in \text{infer}(S \setminus \{h_i\}, x).$$

PROOF. We apply the definition of inference dimension iteratively. Fix  $x \in \mathbb{R}^n$ . Assume that we constructed  $h_1, \dots, h_{i-1}$  so far for  $i \leq m$ . Let  $S_i = S \setminus \{h_1, \dots, h_{i-1}\}$ . As  $|S_i| \geq d$  there exist  $h_i \in S_i$  such that  $S_i \setminus \{h_i\}$  infers  $h_i$  at  $x$ . That is,  $h_i \in \text{infer}(S_i \setminus \{h_i\}, x)$ . But as  $S_i \subset S$  then also  $h_i \in \text{infer}(S \setminus \{h_i\}, x)$ .  $\square$

LEMMA 4.2. Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Let  $S \subset H$  be a uniformly chosen subset of size  $|S| = 2d$ . Then for every  $x \in \mathbb{R}^n$ ,

$$\mathbb{E}_S \left[ |\text{infer}(S, x) \cap H| \right] \geq \frac{|H|}{2}.$$

PROOF. Fix  $x \in \mathbb{R}^n$ . We have

$$\begin{aligned} \mathbb{E}_S \left[ \frac{|\text{infer}(S, x) \cap H|}{|H|} \right] &= \frac{\Pr_{S \subset H, h \in H} [h \in \text{infer}(S, x)]}{|H|} \\ &\geq \frac{\Pr_{S \subset H, h \in H \setminus S} [h \in \text{infer}(S, x)]}{|H|} \\ &= \Pr[h_{2d+1} \in \text{infer}(\{h_1, \dots, h_{2d}\}, x)], \end{aligned}$$

where  $h_1, \dots, h_{2d+1} \in H$  are uniformly chosen distinct elements. The inequality " $\Pr_{S \subset H, h \in H} [h \in \text{infer}(S, x)] \geq \Pr_{S \subset H, h \in H \setminus S} [h \in \text{infer}(S, x)]$ " follows as  $h \in \text{infer}(S, x)$  for any  $h \in S$ .

Let  $R := \{h_1, \dots, h_{2d+1}\}$ . By symmetry it holds that

$$\begin{aligned} &\Pr[h_{2d+1} \in \text{infer}(\{h_1, \dots, h_{2d}\}, x)] \\ &= \frac{1}{2d+1} \sum_{i=1}^{2d+1} \Pr[h_i \in \text{infer}(R \setminus \{h_i\}, x)] \\ &= \mathbb{E}_R \left[ \frac{|\{h_i \in R : h_i \in \text{infer}(R \setminus \{h_i\}, x)\}|}{2d+1} \right]. \end{aligned}$$

By Claim 4.1, for any  $R \subset H$  it holds that  $|\{h_i \in R : h_i \in \text{infer}(R \setminus \{h_i\}, x)\}| \geq |R| - d$ . Thus,

$$\mathbb{E}_S \left[ \frac{|\text{infer}(S, x) \cap H|}{|H|} \right] \geq \frac{d+1}{2d+1} \geq \frac{1}{2}.$$

$\square$

We are now in position to describe the algorithm which establishes Theorem 1.7.

### Zero-error randomized comparison decision tree for $\mathcal{A}_H$

Input:  $x \in \mathbb{R}^n$

Output:  $\mathcal{A}_H(x)$

- (1) Initialize:  $H_0 = H$ ,  $i = 0$ ,  $v(h) = ?$  for all  $h \in H$ .
- (2) Repeat while  $|H_i| \geq 2d$ :
  - (2.1) Sample uniformly  $S_i \subset H_i$  of size  $|S_i| = 2d$ .
  - (2.2) Query  $\text{sign}(\langle h, x \rangle)$  for  $h \in S_i$  and sort the  $\langle h, x \rangle$  using comparison queries.
  - (2.3) Compute  $\text{infer}(S_i, x) \cap H_i$ .
  - (2.4) For all  $h \in \text{infer}(S_i, x) \cap H_i$ , set  $v(h) \in \{-, 0, +\}$  to be the inferred value of  $h$  at  $x$ .
  - (2.5) Set  $H_{i+1} := H_i \setminus (\text{infer}(S_i, x) \cap H_i)$ .
  - (2.6) Set  $i := i + 1$ .
- (3) Query  $\text{sign}(\langle h, x \rangle)$  for all  $h \in H_i$ , and set  $v(h)$  accordingly.
- (4) Return  $v$  as the value of  $\mathcal{A}_H(x)$ .

*Analysis.* In order to establish Theorem 1.7, we first show that for every  $x \in \mathbb{R}^n$ , the algorithm terminates after  $O(\log |H|)$  iterations in expectation. This follows as  $\mathbb{E}[|H_i|] \leq 2^{-i}|H|$ , which we show by induction on  $i$ . It clearly holds for  $i = 0$ . For  $i > 0$  by Lemma 4.2, if we condition on  $H_{i-1}$  then

$$\mathbb{E}_{S_i} [|H_i| \mid H_{i-1}] \leq \frac{|H_{i-1}|}{2}.$$

and hence

$$\mathbb{E}[|H_i|] = \mathbb{E}_{H_{i-1}} [\mathbb{E}_{S_i} [|H_i| \mid H_{i-1}]] \leq \mathbb{E} \left[ \frac{|H_{i-1}|}{2} \right] \leq 2^{-i}|H|.$$

Thus, it remains to bound the number of queries in every round. Observe that the only queries to  $x$  are in steps (2.2) and (3). In step (3) the algorithm makes at most  $2d$  label queries. In step (2.2), we need to compute  $\text{sign}(\langle x, h \rangle)$  for all  $h \in S_i$ , which requires  $|S_i| = 2d$  label queries; and to compute  $\text{sign}(\langle x, h' - h'' \rangle)$  for all  $h', h'' \in S_i$ . This can be done in  $O(d \log d)$  comparison queries by sorting the elements  $\{\langle x, h \rangle : h \in S_i\}$  giving some  $O(d \log d \log |H|)$  bound on the expected total number of queries.

This bound can be improved using Fredman's sorting algorithm [Fre76a].

THEOREM 4.3 ([Fre76a]). Let  $\Pi$  be a family of orderings over a set of  $m$  elements. Then there exists a comparison decision tree that sorts every  $\pi \in \Pi$  using at most

$$2m + \log |\Pi|$$

comparisons.

To use Fredman's algorithm, observe that the ordering, " $<$ ", on  $S_i$  that is being sorted in the  $i$ 'th round is defined by the inner product with  $x$ ,

$$h' < h'' \iff \langle h', x \rangle \leq \langle h'', x \rangle.$$

The following claim bounds the number of such orderings.

CLAIM 4.4. Let  $S \subset \mathbb{R}^n$ . Let  $\Pi_{S,x}$  be the ordering on  $S$  define by inner product with  $x \in \mathbb{R}^n$ . Then

$$|\{\Pi_{S,x} : x \in \mathbb{R}^n\}| \leq (2e|S|^2)^n.$$

PROOF. Observe that  $\Pi_{S,x'} \neq \Pi_{S,x''}$  if and only if there are  $h', h'' \in S$  such that  $\text{sign}(\langle h' - h'', x' \rangle) \neq \text{sign}(\langle h' - h'', x'' \rangle)$ . Thus, the number of different orderings is at most the size of  $\{\mathcal{A}_{S-S}(x) : x \in \mathbb{R}^n\}$ , where  $S - S = \{h' - h'' : h', h'' \in S\}$ . Since  $|S - S| \leq |S|^2$ , Lemma 2.1 implies an upper bound of  $(2e|S|^2)^n$  as claimed.  $\square$

Thus, by using Fredman's algorithm we can sort  $S_i$  with just  $O(|S_i| + n \log |S_i|) = O(d + n \log d)$  comparisons in each round, which gives a total number of

$$O((d + n \log d) \log |H|)$$

queries in total.

## 5 DETERMINISTIC COMPARISON DECISION TREE

We prove Theorem 1.8 in this section, which is a de-randomization of Theorem 1.7.

**Theorem 1.8 (restated).** *Let  $H \subset \mathbb{R}^n$  with inference dimension  $d$ . Then there exists a deterministic comparison decision tree which computes  $\mathcal{A}_H$ , whose query complexity is  $O((d + n \log(nd)) \log |H|)$ .*

First, note the following straightforward Corollary of Lemma 4.2.

**COROLLARY 5.1.** *Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Let  $S \subset H$  be uniformly chosen of size  $|S| = 2d$ . Then*

$$\left( \forall x \in \mathbb{R}^n \right) : \Pr_S \left[ |\text{infer}(S, x) \cap H| \geq \frac{|H|}{4} \right] \geq \frac{1}{4}.$$

Theorem 1.8 follows by establishing a universal set  $S$  which is good for all  $x \in \mathbb{R}^n$ .

**LEMMA 5.2.** *Let  $H \subset \mathbb{R}^n$  be a finite set with inference dimension  $d$ . Then there exists  $S \subseteq H$  of size  $|S| = O(d + n \log d)$  such that:*

$$\left( \forall x \in \mathbb{R}^n \right) : |\text{infer}(S, x) \cap H| \geq \frac{|H|}{8}.$$

We first argue that Theorem 1.8 follows directly from the existence of such an  $S$ . The algorithm is a straightforward adaptation of the zero-error randomized comparison algorithm, except that now we use this set  $S$  which works for all  $x \in \mathbb{R}^n$  in parallel.

### Deterministic Comparison Decision Tree for $\mathcal{A}_H$

Input:  $x \in \mathbb{R}^n$

Output:  $\mathcal{A}_H(x)$

- (1) Initialize:  $H_0 = H$ ,  $i = 0$ ,  $v(h) = ?$  for all  $h \in H$ . Let  $s = O(d + n \log d)$  as in Lemma 5.2.
- (2) Repeat while  $|H_i| \geq s$ :
  - (2.1) Pick  $S_i \subset H_i$  of size  $|S_i| = s$  such that
 
$$\forall x \in \mathbb{R}^n, |\text{infer}(S_i, x) \cap H| \geq \frac{|H|}{8}.$$
  - (2.2) Query  $\text{sign}(\langle h, x \rangle)$  for  $h \in S_i$  and sort the  $\langle h, x \rangle$  using comparison queries.
  - (2.3) Compute  $\text{infer}(S_i, x) \cap H_i$ .
  - (2.4) For all  $h \in \text{infer}(S_i, x) \cap H_i$ , set  $v(h) \in \{-, 0, +\}$  to be the inferred value of  $h$  at  $x$ .
  - (2.5) Set  $H_{i+1} := H_i \setminus (\text{infer}(S_i, x) \cap H_i)$ .
  - (2.6) Set  $i := i + 1$ .
- (3) Query  $\text{sign}(\langle h, x \rangle)$  for all  $h \in H_i$ , and set  $v(h)$  accordingly.
- (4) Return  $v$  as the value of  $\mathcal{A}_H(x)$ .

*Analysis.* Lemma 5.2 ensures that a set  $S_i$  always exist. Thus, for any  $x$ , the algorithm terminates after  $O(\log |H|)$  rounds. Observe that the only queries to  $x$  are in steps (2.2) and (3). In step (3) the algorithm makes at most  $s = O(d + n \log d)$  label queries. In step (2.2), we need to compute  $\text{sign}(\langle x, h \rangle)$  for all  $h \in S_i$ , and to compute  $\text{sign}(\langle x, h' - h'' \rangle)$  for all  $h', h'' \in S_i$ , which can be done sorting the elements  $\{\langle x, h \rangle : h \in S_i\}$ . Using Fredman's algorithm, this requires  $O(|S_i| + n \log |S_i|) = O(d + n \log(dn))$  many comparisons in each round, which gives a total number of

$$O((d + n \log(dn)) \log |H|)$$

queries.

### 5.1 Proof of Lemma 5.2

Let  $S \subset H$  be a uniform subset of size  $|S| = s$  where  $s = O(d + n \log d)$ . Define the event

$$E(S) := \left[ \exists x \in \mathbb{R}^n, |\text{infer}(S, x) \cap H| < \frac{|H|}{8} \right].$$

It suffices to prove that  $\Pr[E(S)] < 1$  to prove the existence of  $S$ . In fact, as we will see, by choosing sufficiently large constants in the choice of  $s = O(d + n \log d)$ , the probability  $\Pr[E(S)]$  can be made  $\leq 1/2$  (say), so a random set would also work.

In order to establish that  $\Pr[E(S)] < 1$  we use a variant of the *double sampling method* [VC71] (see also [VC15]). Let  $T \subset S$  be a uniformly chosen subset of size  $|T| = 2d$ . Define the event

$$E(S, T) := \left[ \exists x \in \mathbb{R}^n, |\text{infer}(T, x) \cap H| < \frac{|H|}{8} \wedge |\text{infer}(T, x) \cap S| \geq \frac{|S|}{4} \right].$$

We bound  $\Pr[E(S)]$  in two steps. We first show that (i)  $\Pr[E(S)] \leq 4 \Pr[E(S, T)]$ , and then that (ii)  $\Pr[E(S, T)] \leq \frac{1}{8}$ .

**CLAIM 5.3.**  $\Pr[E(S)] \leq 4 \Pr[E(S, T)]$ .



PROOF. For each  $S$  for which  $E(S)$  holds fix  $x_S \in \mathbb{R}^n$  such that  $|\text{infer}(S, x_S) \cap H| < \frac{|H|}{8}$ . Then

$$\Pr[E(S, T) \mid S] \geq \Pr \left[ |\text{infer}(T, x_S) \cap H| < \frac{|H|}{8} \bigwedge |\text{infer}(T, x_S) \cap S| \geq \frac{|S|}{4} \right].$$

The first condition holds with probability one, since  $T \subset S$  and hence  $\text{infer}(T, x_S) \subset \text{infer}(S, x_S)$ . For the second condition, as  $T \subset S$  is a uniformly chosen subset of size  $|T| = 2d$ , Corollary 5.1 gives

$$\Pr_T \left[ |\text{infer}(T, x_S) \cap S| \geq \frac{|S|}{4} \mid S \right] \geq \frac{1}{4}.$$

Thus

$$\Pr[E(S, T) \mid S] \geq \frac{1}{4}.$$

As this holds for every  $S$  for which  $E(S)$  holds, we have  $\Pr[E(S, T) \mid E(S)] \geq 1/4$ , which implies the claim.  $\square$

We next bound the probability of  $E(S, T)$ . We will prove that for every fixed  $T$ ,

$$\Pr[E(S, T) \mid T] \leq \frac{1}{8},$$

which will conclude the proof. So, fix  $T \subset H$  of size  $|T| = 2d$ . Let  $T - T$  denote the set  $\{h' - h'' : h', h'' \in T\}$ , and let  $T^* = T \cup (T - T)$ . Recall that  $\mathcal{A}_{T^*}(x)$  is defined by

$$\mathcal{A}_{T^*}(x) = (\text{sign}(\langle h, x \rangle) : h \in T^*) \in \{-, 0, +\}^{T^*}.$$

Observe that the set  $\text{infer}(T, x)$  depends only on  $\mathcal{A}_{T^*}(x)$ ; that is, if  $\mathcal{A}_{T^*}(x') = \mathcal{A}_{T^*}(x'')$  then  $\text{infer}(T, x') = \text{infer}(T, x'')$ . Let  $X_T \subset \mathbb{R}^n$  be a set that contains one representative from each equivalence class of the relation  $x' \sim x'' \iff \mathcal{A}_{T^*}(x') = \mathcal{A}_{T^*}(x'')$ . Thus we can rephrase the event  $E(S, T)$  as

$$E(S, T) = \left[ \exists x \in X_T, |\text{infer}(T, x) \cap H| < \frac{|H|}{8} \bigwedge |\text{infer}(T, x) \cap S| \geq \frac{|S|}{4} \right].$$

The advantage of considering  $X_T$  is that now we can bound the probability of  $E(S, T)$  using a union bound that depends on the (finite) set  $X_T$ . More specifically, let

$$X'_T := \left\{ x \in X_T : |\text{infer}(T, x) \cap H| < \frac{|H|}{8} \right\}.$$

We thus established the following claim.

CLAIM 5.4. *For every  $T \subset H$ ,*

$$\Pr[E(S, T) \mid T] \leq \sum_{x \in X'_T} \Pr_S \left[ |\text{infer}(T, x) \cap S| \geq \frac{|S|}{4} \mid T \right].$$

To conclude, it suffices to upper bound  $|X'_T|$  and the probability that  $|\text{infer}(T, x) \cap S| \geq \frac{|S|}{4}$  for  $x \in X'_T$ . Lemma 2.1 gives an upper bound on  $|X_T|$  which also bounds  $|X'_T|$ ,

$$|X'_T| \leq |X_T| = |\mathcal{A}_{T^*}| \leq (2e|T^*|)^n = 2^{O(n \log d)}.$$

We next bound the probability (over  $S \supset T$ ) that  $|\text{infer}(T, x) \cap S| \geq \frac{|S|}{4}$  for  $x \in X'_T$ .

CLAIM 5.5. *Fix  $T \subset H$  of size  $|T| = 2d$  and fix  $x \in X'_T$ . Assume that  $s \geq 10|T|$ , and let  $S$  be a uniformly sampled set of size  $|S| = s$  such that  $T \subset S \subset H$ . Then*

$$\Pr_T \left[ |\text{infer}(T, x) \cap S| \geq \frac{|S|}{4} \mid T \right] \leq 2^{-\Omega(s)}.$$

PROOF. Let  $R = S \setminus T$ . It suffices to bound the probability of the event that  $|\text{infer}(T, x) \cap R| \geq \frac{|R|}{6}$ . Indeed, if  $|\text{infer}(T, x) \cap S| \geq \frac{|S|}{4}$  then

$$|\text{infer}(T, x) \cap R| \geq \frac{|S|}{4} - |T| = \frac{|R| + |T|}{4} - |T| \geq \frac{|R|}{6},$$

where in the last inequality we used the assumption that  $|R| \geq 9|T|$ .

The set  $R$  is a uniform subset of  $H \setminus T$  of size  $|R| = |S| - |T|$ . By assumption, at most  $\frac{|H \setminus T|}{8}$  of the elements in  $H \setminus T$  are in  $\text{infer}(T, x)$ . By the Chernoff bound, the probability that at least  $|R|/6$  of the sampled elements belong to  $\text{infer}(T, x)$  is thus exponentially small in  $|R|$ . This finishes the proof as  $|R| \geq (9/10)s$ .  $\square$

We now conclude the proof.

$$\Pr[E(S, T) \mid T] \leq |X'_T| 2^{-\Omega(s)} \leq 2^{O(n \log d) - \Omega(s)} \leq 1/8,$$

as we choose  $s = O(d + n \log d)$  with a large enough hidden constant. Then we also have  $\Pr[E(S, T)] \leq 1/8$  and

$$\Pr[E(S)] \leq 4 \Pr[E(S, T)] \leq 1/2.$$

## REFERENCES

- [AC05] Nir Ailon and Bernard Chazelle. Lower bounds for linear degeneracy testing. *Journal of the ACM (JACM)*, 52(2):157–171, 2005.
- [CGI<sup>+</sup>16] Marco L Carmosino, Jiawei Gao, Russell Impagliazzo, Ivan Mihajlin, Ramamohan Paturi, and Stefan Schneider. Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 261–270. ACM, 2016.
- [CIO16] Jean Cardinal, John Iacono, and Aurélien Ooms. Solving k-SUM using few linear queries. In *24th Annual European Symposium on Algorithms, ESA 2016*, pages 25:1–25:17, 2016.
- [CL15] Timothy M Chan and Moshe Lewenstein. Clustered integer 3SUM via additive combinatorics. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 31–40. ACM, 2015.
- [DL74] David Dobkin and Richard J Lipton. On some generalizations of binary search. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 310–316. ACM, 1974.
- [Eri99] Jeff Erickson. Bounds for linear satisfiability problems. *Chicago Journal of Theoretical Computer Science*, page 8, 1999.
- [ES17] Esther Ezra and Micha Sharir. A nearly quadratic bound for the decision tree complexity of k-SUM. In *33rd International Symposium on Computational Geometry, SoCG 2017*, pages 41:1–41:15, 2017.
- [Fre76a] Michael L Fredman. How good is the information theory bound in sorting? *Theoretical Computer Science*, 1(4):355–361, 1976.
- [Fre76b] Michael L Fredman. New bounds on the complexity of the shortest path problem. *SIAM Journal on Computing*, 5(1):83–89, 1976.
- [GO95] Anka Gajentaan and Mark H Overmars. On a class of  $o(n^2)$  problems in computational geometry. *Computational geometry*, 5(3):165–185, 1995.
- [GP14] Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 621–630. IEEE, 2014.
- [GS17] Omer Gold and Micha Sharir. Improved bounds for 3sum, k-sum, and linear degeneracy. In *25th Annual European Symposium on Algorithms, ESA 2017*, pages 42:1–42:13, 2017.
- [GT62] Eiichi Goto and Hidetosi Takahasi. Some theorems useful in threshold logic for enumerating boolean functions. In *IFIP Congress*, pages 747–752, 1962.
- [KLMZ17] Daniel M Kane, Shachar Lovett, Shay Moran, and Jiapeng Zhang. Active classification with comparison queries. *arXiv preprint arXiv:1704.03564*, 2017.
- [MadH84] Friedhelm Meyer auf der Heide. A polynomial linear search algorithm for the n-dimensional knapsack problem. *Journal of the ACM (JACM)*, 31(3):668–676, 1984.

- [Mei93] Stefan Meiser. Point location in arrangements of hyperplanes. *Information and Computation*, 106(2):286–303, 1993.
- [Pet02] Seth Pettie. On the comparison-addition complexity of all-pairs shortest paths. In *Algorithms and Computation, 13th International Symposium, ISAAC 2002 Vancouver, BC, Canada, November 21-23, 2002, Proceedings*, pages 32–43, 2002.
- [VC71] VN Vapnik and A Ya Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.
- [VC15] 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.
- [VW15] Virginia Vassilevska Williams. Hardness of easy problems: Basing hardness on popular conjectures such as the strong exponential time hypothesis (invited talk). In *LIPICs-Leibniz International Proceedings in Informatics*, volume 43. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.
- [Wil14] Ryan Williams. Faster all-pairs shortest paths via circuit complexity. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 664–673. ACM, 2014.
- [Yao81] Andrew Chi-Chih Yao. On the parallel computation for the knapsack problem. In *Proceedings of the thirteenth annual ACM symposium on Theory of computing*, pages 123–127. ACM, 1981.