Near-optimal linear decision trees for k-SUM and related problems

Kane-Lovett-Moran 2017

Micha Sharir

The k-Sum problem

$A = \{x_1, \ldots, x_n\}$: Set of n real numbers; $k \geq 1$: integer

Do there exist k elements of A that sum up to 0?

$$\exists i_1 < i_2 < \cdots < i_k \quad | \quad x_{i_1} + \cdots + x_{i_k} = 0?$$

Special instance of:

- k-**LDT**: k-Linear Degeneracy Testing:

$A = \{x_1, \ldots, x_n\}$: Set of n real numbers; $k \geq 1$: integer

$f(y_1, \ldots, y_k) = a_0 + a_1 y_1 + \cdots + a_k y_k$: Real linear function

Do there exist k elements of A that satisfy

$$f(x_{i_1}, x_{i_2}, \ldots, x_{i_k}) = 0?$$

Special instance of:

SubsetSum and **Knapsack**:

$A = \{x_1, \ldots, x_n\}$: Set of n real numbers

Does there exist a subset of A whose elements sum up to 0 (**SubsetSum**) or to 1 (**Knapsack**)?

NP-complete problems

Other problems considered by [KLM'17]

Sorting $A + B$ ([Fredman 76]):
A and B: Two sets of n real numbers
$A + B = \{a + b \mid a \in A,\ b \in B\}$
Sort $A + B$

Zero triangles ([Grønlund, Pettie 14]):
$G = (V, E)$ a graph on n vertices and m edges
Given real edge weights, does G contain a triangle whose edge weights sum to 0?

s-Sparse Linear Decision Tree Model

Count only (linear) comparisons, each of the form

$$\sum_{i=1}^{s} \alpha_i x_{m_i} = 0, \text{ or } < 0, \text{ or } > 0$$

for $x_i \in A$, α_i reals (integers)

Other operations are
(a) Free; and
(b) Not allowed to touch $x_1, \ldots, x_n$
(Can use symbolic info on input, inferred from earlier tests)

Linear Decision Tree Model: $s = n$

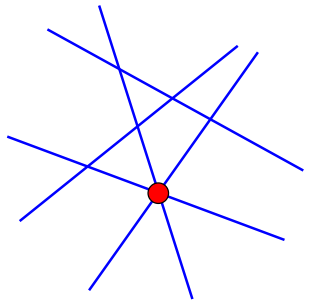
k-Sum: Background

Simplest instance: 3-**Sum**:
Do there exist $a, b, c \in A$ such that $a + b + c = 0$?

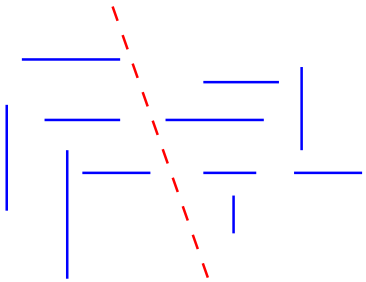
Many geometric problems are 3-**Sum**-hard:
(Subquadratic) reductions from 3-**Sum**
[Gajentaan, Overmars 95], [Barequet, Har-Peled 01]

Some 3-Sum hard problems

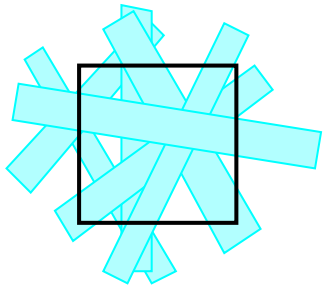
Does a set of n lines in the plane have three concurrent lines?



Given a set of n axis-parallel pairwise disjoint segments in the plane, is there a line that separates them?



Does a set of n strips in the plane cover the unit square?



Computing 3-**Sum**

In the real RAM model:

Can be solved in $O(n^2)$ time (easy)

In $O(n^2/\log^\beta n)$ time
Groundbreaking [Grønlund, Pettie 14]
Slight subsequent improvements [Freund 17], [Gold, Sharir 17]

No subquadratic solutions (yet) for any of the
Geometric 3-**Sum**-hard problems

Computing k-Sum

In the real RAM model:

For k odd, $O\left(n^{\lceil k/2 \rceil}\right)$ time

For k even, $O\left(n^{k/2} \log n\right)$ time

[Erickson 99]

Number of comparisons is always $O\left(n^{\lceil k/2 \rceil}\right)$

[Lambert 92], [Erickson 99]

Bound is **optimal** in the k-sparse linear decision tree model

[Erickson 99], [Ailon, Chazelle 05]

Comparisons for k-Sum: Depend on sparsity

Number of comparisons is $\Theta\left(n^{\lceil k/2 \rceil}\right)$
For k-sparse trees

Improves (for k odd) to $\approx O\left(n^{k/2}\right)$
for $r = 2k - 2$ ($r = 4$ for 3-**Sum**) [Grønlund, Pettie 14]

Improves much more drastically when $s = n$
(arbitrary linear queries):

$O(n^4 \log n)$ [Meyer auf der Heide 84], [Meiser 93]
$O(n^3 \text{polylog}(n))$ [Cardinal, Iacono, Ooms 16]
$O(n^2 \log^2 n)$ [Ezra, Sharir, 17]
Constants depend on k

Best lower bound: $\Omega(n \log n)$ [Dobkin, Lipton 79]

The KLM Breakthrough

All problems mentioned above go down in one fell swoop
Using $O(n \log^2 n)$ linear tests (of only two special kinds)

Recall: s-sparse linear test:

$$\sum_{i=1}^{s} \alpha_i x_{m_i} = 0, \text{ or } < 0, \text{ or } > 0$$

Assume the α_i's are integers
Weight of a test: its L_1-norm $\sum_{i=1}^{s} |\alpha_i|$

Additive test: All nonzero coefficients are ± 1
Additive s-sparse test: Weight $\leq s$ (only s nonzero terms)

The KLM Breakthrough

- k-**Sum**:
$O(kn \log^2 n)$ additive $2k$-sparse tests (down from $O(kn^2 \log^2 n)$)

- k-LDT:
$O(kn \log^2 n)$ additive $(2k+2)$-sparse tests ($O(kn^2 \log^2 n)$)

- Sorting $A + B$:
$O(n \log^2 n)$ additive 8-sparse tests ($O(n^2)$)

- **SubsetSum**:
$O(n^2 \log n)$ additive tests ($O(n^3 \log n)$)

- Zero triangles:
$O(m \log^2 m)$ additive 6-sparse tests ($O(m^{5/4})$)

Technique: Point location in arrangements of hyperplanes

k-Sum and point location

Input set $A = (x_1, x_2, \ldots, x_n)$: Point $\mathbf{x}$ in $\mathbb{R}^n$

H: Set of the $\binom{n}{k}$ hyperplanes

$$x_{i_1} + x_{i_2} + \cdots + x_{i_k} = 0$$

for $1 \leq i_1 < i_2 < \cdots < i_k \leq n$

The problem: Does $\mathbf{x}$ lie on any hyperplane of H?
A simple variant of point location

For **SubsetSum**: Take $H =$ all $2^n - 1$ such hyperplanes (all k)

Point location in arrangements of $N = \binom{n}{k}$ hyperplanes
In (high) dimension n

- Well studied problem for nearly 30 years

Mainly for small / constant dimensions

Ignoring dependence of constants of proportionality on dimension

- Essentially all previous techniques use

Recursive space decomposition based on random sampling

Point location in arrangements of hyperplanes: Preprocessing

- Take a random sample R of r hyperplanes of H

r a sufficiently large "constant" (depending on dimension)

- Construct a space decomposition $\mathcal{D}(R)$

Of the cells of the arrangement $\mathcal{A}(R)$ into simple cells
(Bottom-vertex triangulation / vertical decomposition)

- Random sampling theory: Each cell of $\mathcal{D}(R)$ is crossed by only a few hyperplanes of H

(Because it is not crossed by any hyperplane of R)

- Recurse within each cell τ with its conflict list:

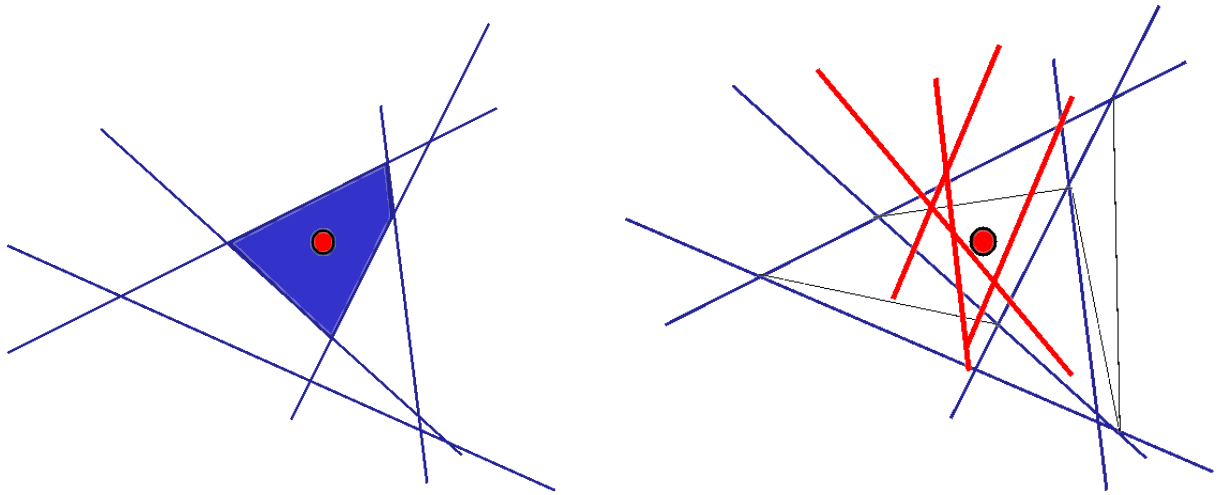
Set of hyperplanes crossing τ

Point location in arrangements of hyperplanes: Query

- Given a query $\mathbf{x} \in \mathbb{R}^n$,

Find (in brute force) the cell τ of $\mathcal{D}(R)$ that contains $\mathbf{x}$

- Recurse on the (random sample from the) conflict list of τ



Main issue (for large dimension d)

Need to choose r large (in terms of d)
To ensure that each conflict list is small (of size $\leq |H|/2$)
To imply: recursion depth $= O(\log |H|)$

Affects query cost:
Expensive to locate the (sub)cell containing the query

All previous works face this issue
Latest and greatest (in terms of number of pages):
[Ezra, Har-Peled, Kaplan, Sharir 17]

Also: Need to decompose cells into simple subcells
To make the random sampling theory work
Complicates the algorithm and the analysis

Point location in arrangements of hyperplanes: The [KLM'17] approach

- Completely different (not really; see / hear later)

Based on inferences and inference dimension

Concepts from active learning

- Only two types of linear tests:

(Assume: all hyperplanes go through the origin, as in k-**Sum**)

Label tests: sign($\langle h, x \rangle$)

Comparison tests: sign($\langle h - h', x \rangle$)

For original hyperplanes h only

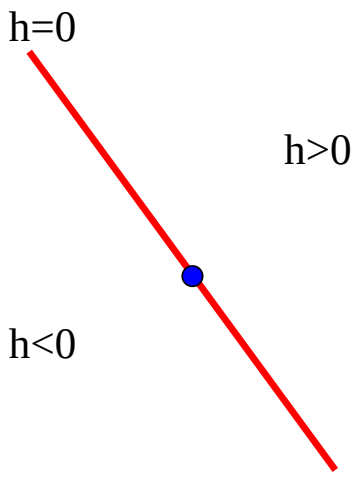
- Works only for very special hyperplanes

Inference

The right tool at the right place at the right time

Consider only hyperplanes h that pass through the origin
Regard h as a linear form (dual to a point in $\mathbb{R}^n$)
Partitioning $\mathbb{R}^d$ into the regions

$$h^+ = \langle h, x \rangle < 0, \quad h^0 = \langle h, x \rangle = 0, \quad h^- = \langle h, x \rangle > 0$$



Inference

Label tests: sign($\langle h, x \rangle$)
Identify $H^0(x) =$ those $h \in H$ with $\langle h, x \rangle = 0$

Comparison tests: sign($\langle h - h', x \rangle$)
Sort the two subsets
$H^+(x) =$ those $h \in H$ with $\langle h, x \rangle > 0$
$H^-(x) =$ those $h \in H$ with $\langle h, x \rangle < 0$

This is the data we play with

(Note: $H^+(x)$, $H^-(x)$ are somewhat arbitrary:
h and λh are the same hyperplane
With different values of $\langle h, x \rangle$
Affine vs. projective)

Inference

S: Set of hyperplanes
h: Hyperplane (not in S)
x: point

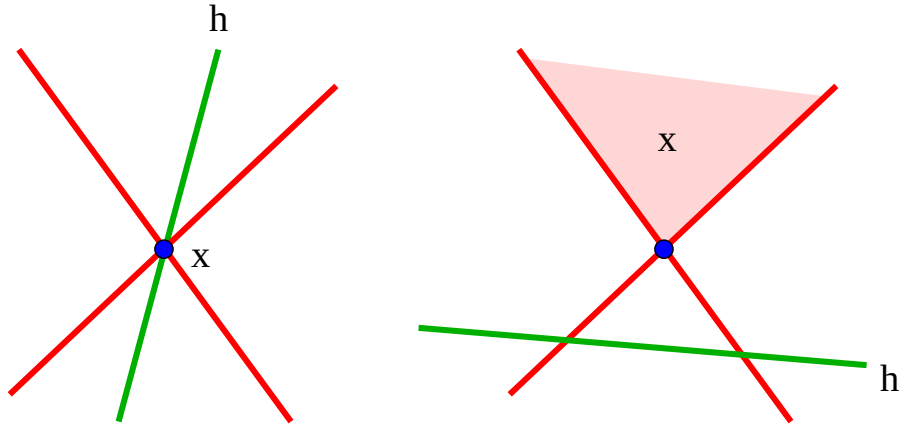
S infers h at x:
Knowing all the signs $\langle h', x\rangle$, for $h' \in S$
And the signs of all comparisons $\langle h' - h'', x\rangle$, for $h', h'' \in S$
(**Equivalently**, knowing $S^0(x)$ and the sorted $S^+(x)$, $S^-(x)$)
Determines $\langle h, x\rangle$

Inference

Equivalently,

$C(S;x) := \{y \in \mathbb{R}^d \mid S^0(y) = S^0(x),$
$S^+(y) = S^+(x)$ (sorted), and $S^-(y) = S^-(x)$ (sorted)$\}$

Then S infers h at $x \equiv$
$h = 0$ either contains or is disjoint from $C(S;x)$



$C(S;x)$ is a cell (cone) in $\mathcal{A}(S \cup (S - S))$

Inference

Testing whether S infers h at x is easy
(without computing $\langle h, x \rangle$):

- Obtain all the signs of $\langle h', x \rangle$, for $h' \in S$

And of $\langle h' - h'', x \rangle$, for $h', h'' \in S$

- Compute $C(S; x)$

(As intersection of $|S|$ halfspaces and/or hyperplanes)

- Check whether h contains or is disjoint from $C(S; x)$

Last two steps are for free

Main issue: Guarantee that S has many inferrable hyperplanes

Inference dimension of a set H of hyperplanes:

Smallest d such that
For any subset $S \subseteq H$ of size d, and
For any $x \in \mathbb{R}^d$

There exists $h \in S$ such that $S \setminus \{h\}$ infers h at x

Step I: k-Sum and friends have low inference dimension

Theorem 1:
The inference dimension of $H = \{h \in \mathbb{Z}^n \mid \|h\|_1 \leq w\}$
is $d = O(n \log w)$

For k-**Sum**: $d = O(n \log k)$
For **SubsetSum**: $d = O(n \log n)$

Step II: Low inference dimension $\Longrightarrow$ Fast randomized decision tree

Theorem 2: $H \subset \mathbb{R}^n$ finite, with inference dimension d
Can compute $\{\text{sign}(\langle h, x\rangle) \mid h \in H\}$
For any query $x \in \mathbb{R}^n$
By a zero-error randomized comparison decision tree
With expected query complexity $O\left((d + n \log d) \log |H|\right)$

For k-**Sum**: $d = O(n \log k)$, $|H| = O(n^k)$:
Expected query complexity $O\left(kn \log^2 n\right)$

For **SubsetSum**: $d = O(n \log n)$, $|H| = 2^n - 1$:
Expected query complexity $O\left(n^2 \log n\right)$

Step III: Fast randomized decision tree $\Longrightarrow$ Fast deterministic tree

Derandomizing Theorem 2:

Theorem 3: $H \subset \mathbb{R}^n$ finite, with inference dimension d
Can compute $\{\text{sign}(\langle h, x\rangle) \mid h \in H\}$
For any query $x \in \mathbb{R}^n$
By a deterministic comparison decision tree
With query complexity $O\left((d + n\log(nd))\log|H|\right)$

$\approx$ Same (now deterministic) bounds for k-**Sum** and **SubsetSum**

For many applications, get more than we need:

- Typically, we only want to know whether some hyperplane $h \in H$ contains x (whether $\langle h, x \rangle = 0$)

- For sorting $A + B$, all signs are needed:

Once we have the signs, we can sort $A + B$ "for free"
Without touching the input any more

Fast randomized trees for low inference dimension

Claim: $S \subset \mathbb{R}^n$ with inference dimension d, and $|S| = d + m$
Then for each $x \in \mathbb{R}^n$ there exist $h_1, \ldots, h_m \in S$
Such that $h_i \in \text{infer}\,(S \setminus \{h_i\})$ for each i

Proof: Find one such h_1, by definition, remove, and repeat

Fast randomized trees for low inference dimension

Main technical Claim: For any $x \in \mathbb{R}^n$, a random sample $S \subseteq H$ of size $2d$ infers in expectation $\geq |H|/2$ elements of H at x

In arrangements lingo: For any $x \in \mathbb{R}^n$, a random sample $S \subseteq H$ of size $2d$ has the property that at most $|H|/2$ hyperplanes of H cross $C(S; x)$ (The cell containing x in $\mathcal{A}(S \cup (S - S))$)
(Cross: intersect but not contain)

Fast randomized trees for low inference dimension

Main technical Claim: For any $x \in \mathbb{R}^n$, a random sample $S \subseteq H$ of size $2d$ infers in expectation $\geq |H|/2$ elements of H at x

Proof: Estimate $\mathbf{Pr}\,[S \text{ infers } h \text{ at } x]$
Over all random choices of $|S| = 2d$ and $h \notin S$

Same as estimating $\mathbf{Pr}\left[\{h_1, \ldots, h_{2d}\} \text{ infers } h_{2d+1} \text{ at } x\right]$
Over all random choices of $2d+1$ distinct elements $h_1, \ldots, h_{2d+1} \in H$

Fast randomized trees for low inference dimension

Estimating $\mathbf{Pr}\left[\{h_1, \ldots, h_{2d}\} \text{ infers } h_{2d+1} \text{ at } x\right]$
Over all random choices of $2d+1$ distinct elements
$h_1, \ldots, h_{2d+1} \in H$:

By symmetry, same as $\sum_{h \in S'} \mathbf{Pr}\left[S' \setminus \{h\} \text{ infers } h \text{ at } x\right]$

Conditioned on the choice of a set $S' \subseteq H$ of size $2d+1$

By first claim, this is $\geq \frac{|S'| - d}{2d+1} = \frac{d+1}{2d+1} > \frac{1}{2}$

And the claim follows

The (simple) algorithm

Input: H set of hyperplanes with inference dimension d; $x \in \mathbb{R}^n$
Output: $\{\text{sign}(\langle h, x\rangle) \mid h \in H\}$ (or just find $h \in H$ with $\langle h, x\rangle = 0$)

- Take a random sample $S \subseteq H$ of $2d$ hyperplanes
- Compute $S^0(x)$, $S^+(x)$ (sorted) and $S^-(x)$ (sorted)
- Compute infer(S, x) (hyperplanes not crossing $C(S; x)$)

For each $h \in \text{infer}(S, x)$, find sign$(\langle h, x\rangle)$ (Both substeps free!)

- Remove all inferred hyperplanes from H

(At least half in expectation!)

- Recurse on remaining hyperplanes (Brute force when $|H| \leq 2d$)

The (simple) analysis

- $\log |H|$ stages in expectation

- In each stage:

$2d$ label queries yield $S^0(x)$, and the unsorted $S^+(x)$, $S^-(x)$

Then $O(d \log d)$ comparison queries sort $S^+(x)$, $S^-(x)$

Overall: $O(d \log d \log |H|)$ linear queries

The (simple) analysis

- Can be improved to $O((d + n \log d) \log |H|)$ queries

Using [Fredman 76]:

Sorting m elements, when the output order belongs to a set Π of permutations, can be done by a comparison tree of depth at most $2m + \log |\Pi|$

Here $\Pi =$ all possible orders of $2d$ hyperplanes along a vertical line in $\mathbb{R}^{n+1}$

$\equiv$ Complexity of an arrangement of $O(d^2)$ hyperplanes in $\mathbb{R}^n$

$|\Pi| = d^{O(n)}$, so sorting takes $4d + \log |\Pi| = O(d + n \log d)$ steps

Still missing: Low inference dimension

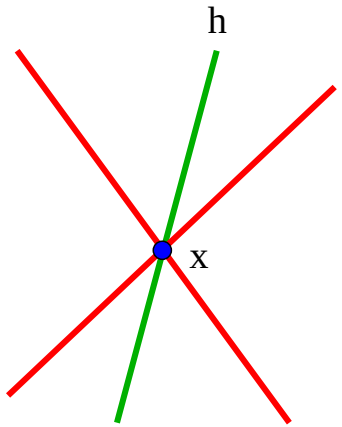
Recall: Estimate smallest d such that
For any set S of d hyperplanes from H and any $x \in \mathbb{R}^n$
S has one hyperplane inferrable from the others at x

Case I: $|S^0(x)| \geq n+1$:
At least $n+1$ hyperplanes $h \in S$ satisfy $\langle h, x \rangle = 0$
Easy: Find $h_1, \ldots, h_n, h_{n+1}$ in $S^0(x)$ such that

$$h_{n+1} = \sum_{i=1}^{n} \alpha_i h_i$$

Then $S \setminus \{h_{n+1}\}$ infers h_{n+1} at x: $\langle h, x \rangle = 0$

Low inference dimension, Case I



Low inference dimension, Case II

Case II: For most $h \in S$, $\langle h, x \rangle > 0$ or $\langle h, x \rangle < 0$
Not so easy: Take the sorted $S^+(x)$ (say)
Enumerate $S^+(x)$ as $(h_1, \ldots, h_m)$, so that

$$0 < \langle h_1, x \rangle \leq \langle h_2, x \rangle \leq \cdots \leq \langle h_m, x \rangle$$

Main technical Lemma: If $m = \Theta(n \log w)$ then
There exists $j \leq m$ and $\alpha_1, \ldots, \alpha_{j-1} \in \{-1, 0, 1\}$ such that

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

Proof: Later

Low inference dimension, Case II

Assume $\alpha_{j-1} = -1$. Add

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

to

$$h_j = h_1 + \sum_{i=1}^{j-1} (h_{i+1} - h_i)$$

and get

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

All coefficients are non-negative, and the last one is 0

So h_j inferrable from $h_1, \ldots, h_{j-1}$ at x: $\langle h_j, x \rangle > 0$

Last missing ingredient

Proof of main technical Lemma: If $m = \Theta(n \log w)$ then There exists $j \leq m$ and $\alpha_1, \ldots, \alpha_{j-1} \in \{-1, 0, 1\}$ such that

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

Proof of main technical Lemma

Pigeons to the rescue: For each $\beta \in \{0,1\}^{m-1}$ define $f(\beta) = \sum_{i=1}^{m-1} \beta_i (h_{i+1} - h_i)$

Get 2^{m-1} vectors in $\mathbb{Z}^n$, each of L_1-norm at most $2(m-1)w$

How many such (distinct) vectors?

(a) Place $2(m-1)w$ balls into $n+1$ bins: $\binom{2(m-1)w+n}{n}$

(b) Choose the sign of each (nonzero) coordinate: 2^n

Proof of main technical Lemma

Choose m large so that

$$2^{m-1} > 2^n \binom{2(m-1)w+n}{n} \approx \left(\frac{4emw}{n}\right)^n$$

Holds if $m = \Theta(n \log w)$

Pigeonhole: There are two distinct vectors $\beta, \beta' \in \{0,1\}^{m-1}$ Such that

$$0 = f(\beta) - f(\beta') = \sum_{i=1}^{m-1} (\beta_i - \beta'_i)(h_{i+1} - h_i)$$

Take $\alpha = \beta - \beta'$

Fast deterministic trees

If we sample $S \subseteq H$ of size $\Theta(d + n \log d)$
(instead of $2d$)
Then, with constant probability, at each $x \in \mathbb{R}^n$,
S infers at least $|H|/8$ hyperplanes of H
(Every cell of $\mathcal{A}(S \cup (S - S))$ crossed by $\leq 7|H|/8$ hyperplanes)

Proof via "double sampling"
(Similar to [Vapnik, Chervonenkis 71], [Haussler, Welzl 87])

Use the same algorithm otherwise

Query time barely changes

Summary, reflections

- Overall approach similar to cutting-based solutions:

(1) Take a (small) random sample S

(2) Argue that each "cell" (or only cell containing the query) in a suitable decomposition of $\mathcal{A}(S)$, has small conflict list

(3) Find the cell containing the query (linear queries only here!)

(4) Recurse

- A simple alternative to cuttings:

(1) Almost no decomposition: Work with cells of $\mathcal{A}(S \cup (S - S))$

(2) No recursion on dimension (in constructing the decomposition)

(3) Use only original hyperplanes (and their pairwise differences)

Summary, reflections

- But, cooling down,

Only works for "nice" hyperplanes:

Integer coefficients with bounded L_1-norm

- Actually, everything works for general hyperplanes too

Except for the pigeonhole lemma

Can it be strengthened / generalized ?

- Can we extend this to the RAM model (for nice hyperplanes)

Counting everything, including preprocessing and storage?

Can we improve current best techniques?

Yes! (work in progress [Kaplan, Sharir])

Thank You