Learning from Nisan’s natural proofs

Ari Karchmer
Boston University

MIT CIS seminar
03/22/24

Zero-Knowledge proofs...
Natural Proofs:
where we gain more than “just” a theorem!

*example of cryptographer*
A **style** or **type** of circuit lower bound

All known circuit lower bounds at the time were natural proofs, or could be made so
We introduce the notion of natural proof. We argue that the known proofs of lower bounds on the complexity of explicit Boolean functions in non-monotone models fall within our definition of natural. We show based on a hardness assumption that natural proofs can’t prove superpolynomial lower bounds for general circuits. We show that the weaker class of $AC^0$-natural proofs which is sufficient to prove the parity lower bounds of Fortnow, Saxe, and Sipser; Yao; and Hastad is inherently incapable of proving the bounds of Razborov and Smolensky. We give some formal evidence that natural proofs are indeed natural by showing that every formal complexity measure which can prove super-polynomial lower bounds for a single function, can do so for almost all functions, which is one of the key requirements to a natural proof in our sense.

A style or type of circuit lower bound

All known circuit lower bounds at the time were natural proofs, or could be made so

Why care? We should understand whether this technique could be used to separate $P$ and $NP$, or whether other techniques are needed.

Precedent for concern: Baker, Gill and Solovay’s prior work on relativizing proofs
Natural Proofs

Alexander A. Razborov*
School of Mathematics
Institute for Advanced Study
Princeton, NJ 08540
and
Steklov Mathematical Institute
Vavilova 42, 117966, GSP-1
Moscow, RUSSIA

Steven Rudich†
Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

We introduce the notion of natural proof. We argue that the known proofs of lower bounds on the complexity of explicit Boolean functions in non-monotone models fall within our definition of natural. We show based on a hardness assumption that natural proofs can't prove super-polynomial lower bounds for general circuits. We show that the weaker class of $AC^0$-natural proofs which is sufficient to prove the parity lower bounds of Fürst, Saxe, and Sipser; Yao; and Hastad is inherently incapable of proving the bounds of Razborov and Smolensky. We give some formal evidence that natural proofs are indeed natural by showing that every formal complexity measure which can prove super-polynomial lower bounds for a single function, can do so for almost all functions, which is one of the key requirements to a natural proof in our sense.

In this paper we introduce the notion of a natural proof. We argue that all lower bound proofs for non-monotone models known to us in non-uniform Boolean complexity either are natural or can be represented as natural. We show that if a cryptographic hardness assumption is true, then no natural proof can prove super-polynomial lower bounds for general circuits.
Natural Proofs and Properties
(Razborov-Rudich, 1997)
Lower bounds that encode algorithms

Natural Proofs of lower bounds against circuit class $\Lambda$ identify Natural Properties

**Large:** $Q_n$ is at least $1/4$ the size of $F_n$

**Useful:** If $f \in \Lambda_n$, then $f \notin Q_n$

**Constructive:** The predicate “is $f \in Q_n$” can be computed in polynomial time (in the size of the truth table)
Natural Proofs and Properties
(Razborov-Rudich, 1997)
Lower bounds that encode algorithms

Natural Proofs of lower bounds against circuit class $\Lambda$ identify Natural Properties

**Large:** $Q_n$ is at least 1/4 the size of $F_n$

**Useful:** If $f \in \Lambda_n$, then $f \notin Q_n$

**Constructive:** The predicate “is $f \in Q_n$” can be computed in polynomial time (in the size of the truth table)

This is the algorithm

Hard functions live here
Power of Natural Properties

A distinguisher?

Natural Proofs of lower bounds against circuit class $\Lambda$ identify Natural Properties

**Large:** $Q_n$ is at least $1/4$ the size of $F_n$

**Useful:** If $f \in \Lambda_n$, then $f \notin Q_n$

**Constructive:** The predicate “is $f \in Q_n$” can be computed in polynomial time (in the size of the truth table)

$$\Pr_{f \sim F_n} \left[ A(t_t f) = 1 \right] - \Pr_{f \sim \Lambda_n} \left[ A(t_t f) = 1 \right] \geq \frac{1}{4}$$

This is the algorithm
Natural Proofs
(Razborov-Rudich, 1997)
Lower bounds that are self-defeating

Natural Proofs of lower bounds against circuit class $P/poly$ break strong one-way functions (And basically all of cryptography)

**Large**: $Q_n$ is at least $1/4$ the size of $F_n$

**Useful**: If $f \in P/poly$, then $f \not\in Q_n$

**Constructive**: The predicate “is $f \in Q_n$” can be computed in polynomial time (in the size of the truth table)

This is the algorithm

**Sketch:**

**OWF $\rightarrow$ PRF** (HILL, ‘99 / GGM, ‘86)

**Large** implies that a random truth table is accepted by the property with probability $> 1/4$

**Useful** implies that PRFs are never accepted

**Constructive** implies that given query access to the PRF, we can actually run the algorithm efficiently
Natural Proofs
(Razborov-Rudich, 1997)
Lower bounds that are self-defeating

Natural Proofs of lower bounds against circuit class $P/poly$ break strong one-way functions (And basically all of cryptography)

Sketch:

$\text{OWF} \rightarrow \text{PRF}$ (HILL, `99 / GGM, `86)

Large implies that a random truth table is accepted by the property with probability $> 1/4$

Useful implies that PRFs are never accepted

Constructive implies that given query access to the PRF, we can actually run the algorithm efficiently
Learning from Natural Proofs
(Carmosino-Impagliazzo-Kabanets-Kolokolova, 2016)

Natural Proofs of lower bounds against circuit class \( P/poly \) imply that \( P/poly \) is learnable. This is stronger than just breaking PRF.

Sketch:

Use queries to map the unknown function to a truth table TT. Queries are derived from NW-generator (Nisan-Wigderson, 1994) —very intricate.

Large, Useful and Constructive implies that given query access to the the table, we can run the natural proof to obtain a distinguisher for TT, which then becomes A learning algorithm by unwinding NW.
Learning from Natural Proofs
(Carmosino-Impagliazzo-Kabanets-Kolokolova, 2016)

Natural Proofs of lower bounds against circuit class $P/poly$ imply that $P/poly$ is learnable. This is stronger than just breaking PRF.
Learning from Natural Proofs
(Carmosino-Impagliazzo-Kabanets-Kolokolova, 2016)

Open Question:
Can we use “random examples”—not queries—and can we get \( \Lambda \) that doesn’t contain \( \text{AC}^0[2] \)?

- Uses very intricate queries stemming from hardness amplification procedures and Nisan-Wigderson generator
- Hypothesis circuit only approximates over the uniform distribution (from hardness amplification procedure)
- Only applies to \( \Lambda \) that contains \( \text{AC}^0[2] \) (constant depth, unbounded fan-in, And/Or/Not circuit)
  - An artifact of the proof of CIKK — Nisan-Wigderson generator is \( \text{AC}^0[2] \)-local but not \( \text{AC}^0 \)-local s with MOD2 gates)
PAC-learning (original model)

Unlabelled examples $x \in \{0,1\}^n$ are sampled according to any unknown distribution $x \sim \rho$.
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for poly(n) size $\Lambda$-circuits, in the original PAC-learning model?
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$ -circuits of size $\exp(n)$ imply polynomial time learning algorithms for $\text{poly}(n)$ size $\Lambda$-circuits, in the original PAC-learning model?

Probably not!
And this follows from a simple but under-the-radar observation.
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for $\text{poly}(n)$ size $\Lambda$-circuits, in the original PAC-learning model?

So why probably not?

Nisan (1993) proved lower bounds against exponential size depth-2 majority circuits

Nisan’s proof is *Natural*. (If you look hard, you can find references to this as early as (Raz, 2000), but we are the first to explicitly formalize)
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for $\poly(n)$ size $\Lambda$-circuits, in the original PAC-learning model?

So why probably not?

Nisan (1993) proved lower bounds against exponential size depth-2 majority circuits

Nisan’s proof is *Natural*. (If you look hard, you can find references to this as early as (Raz, 2000), but we are the first to explicitly formalize)

**But:** Klivans-Sherstov (2009) show that depth-2 majority circuits are not PAC-learnable, under Lattice-based cryptographic assumptions.

“*Yes, for every $\Lambda$*” breaks crypto!
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\text{exp}(n)$ imply polynomial time learning algorithms for $\text{poly}(n)$ size $\Lambda$-circuits, in the original PAC-learning model?

So why probably not?

Nisan (1993) proved lower bounds against exponential size circuits

Nisan’s proof is Natural. (If you look hard, you can find references to this as early as (Raz, 2000), but we are the first to explicitly formalize)

But: Klivans-Sherstov (2009) show that depth-2 majority circuits are not PAC-learnable, under Lattice-based cryptographic assumptions.

“Yes, for every $\Lambda$” breaks crypto!
Nisan’s natural proofs

So what is Nisan’s natural proof method?
Nisan’s natural proofs

So what is Nisan’s natural proof method?

Isolate a function $F$ with very high communication complexity like $\Omega(n)$ in 2-party randomized model. There are many such functions (e.g. Inner product mod 2).

Consider the candidate circuit class $\Lambda$ you would like to prove the lower bound for.

1. Show that every Ckt in $\Lambda$ (size $s(n)$) has a CC protocol of complexity $k(s(n))$
2. Conclude that $F$ does not have $\Lambda$-circuits of size $g(n)$, where $g(k(s(n))) = \Omega(n)$! ...... (By contradiction)
So what is Nisan’s natural proof method?

Isolate a function $F$ with very high communication complexity like $\Omega(n)$ in 2-party randomized model. There are many such functions (e.g. Inner product mod 2)

Consider the candidate circuit class $\Lambda$ you would like to prove the lower bound for.

1. Show that every Ckt in $\Lambda$ (size $s(n)$) has a CC protocol of complexity $k(s(n))$
2. Conclude that $F$ does not have $\Lambda$-circuits of size $g(n)$, where $g(k(s(n))) = \Omega(n)!$ ….. (By contradiction)

E.g.: (Nisan, 1993). Depth-2 Maj-circuits of size $s(n)$ have a randomized CC protocol of complexity $k(s(n))$ for $k = O(\log(.))$

Thus, IPmod2 requires Depth-2 Maj-circuits of size $\exp(\Omega(n))$!
Informal main theorem of this work (K., 2024)

Any circuit class $\Lambda$ (size $s(n)$), which has a $g(n)$ lower bound via Nisan’s method, has a “Distributional PAC-learning” algorithm that runs in time $\exp(g^{-1}(s(n)))$. Consider what happens when $s(n) := \text{poly}(n)$, and $g(n) := \exp(n)$. 
Informal main theorem of this work (K., 2024)

Any circuit class $\Lambda$ (size $s(n)$), which has a $g(n)$ lower bound via Nisan’s method, has a “Distributional PAC-learning” algorithm that runs in time $\exp(g^{-1}(s(n)))$.

Consider what happens when $s(n) := \text{poly}(n)$, and $g(n) := \exp(n)$

This gets around the impossibility by considering:

a) Any $p$-samp distribution over concepts
b) Complexity of evaluation of concepts, not concepts themselves
c) Non-black box usage of lower bounds (Nisan’s specific techniques!)

Turns out this is essentially best possible, if you further inspect the hardness of learning result of (Klivans-Sherstov, 2009)
Informal main theorem of this work (K., 2024)

Any circuit class $\Lambda$ (size $s(n)$), which has a $g(n)$ lower bound via Nisan’s method, has a “Distributional PAC-learning” algorithm that runs in time $\exp(g^{-1}(s(n)))$.

Consider what happens when $s(n) := \text{poly}(n)$, and $g(n) := \exp(n)$

This gets around the impossibility by considering:

a) Any p-samp **distribution** over concepts
b) Complexity of **evaluation of concepts**, not concepts themselves
c) **Non-black box usage of lower bounds** (Nisan’s specific techniques!)

Turns out this is essentially best possible, if you further inspect the hardness of learning result of (Klivans-Sherstov, 2009)

Recall concrete example: depth-2 majority-circuits of size $\text{poly}(n)$ have and $\exp(n)$ lower bound.
Distributional PAC-learning (K., 2024)

Just like PAC-learning, but “Bayesian”
Distributional PAC-learning (K., 2024)

Just like PAC-learning, but “Bayesian”

There are lots of independent benefits of distributional PAC-learning!

- it allows black-box boosting (Schapire 1990)
- other avg-case learning models do not
- still related well to theory of cryptography
  - (Kearns-Valiant, 1994) hardness still goes through
    - we can consider interesting $f \sim \mu$ anyways
  - Hardness with fixed $x \sim \rho$ (p-samp) implies OWFs
Distributional PAC-learning (K., 2024)

Just like PAC-learning, but “Bayesian”

There are lots of independent benefits of distributional PAC-learning:

- it allows black-box boosting (Schapire 1990)
- other avg-case learning models do not
- still related well to theory of cryptography
- (Kearns-Valiant, 1994) hardness still goes through
- we can consider interesting $f \sim \mu$ anyways
- Hardness with fixed $x \sim \rho$ (p-samp) implies OWFs

Change the rules so cryptographers and learners can both win!!
Open Question
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for $\poly(n)$ size $\Lambda$ -circuits, in the original PAC-learning model?

Left out so far is that CIKK16 actually invokes their implication from natural proofs to query learning using the existing natural proofs against $\AC^0[p]$ by (Razborov-Smolensky, 1987)
Ruling out weak PRFs with distPAC-learning (K., 2024)

Open Question
Let \( \Lambda \) be any circuit class. Do Natural Proofs useful against \( \Lambda \)-circuits of size \( \exp(n) \) imply polynomial time learning algorithms for \( \text{poly}(n) \) size \( \Lambda \)-circuits, in the original PAC-learning model?

Left out so far is that CIKK16 actually invokes their implication from natural proofs to query learning using the existing natural proofs against \( \text{AC}^0[p] \) by (Razborov-Smolensky, 1987)

One of the motivations of Open Question is to perhaps rule out conjecture weak PRFs in \( \text{AC}^0[2] \) (Boyle et al., 2021)

DistPAC-learning is enough to rule out weak PRFs. Thus we invoke our theorem with Nisan’s natural proofs to rule out weak PRFs evaluable by depth-2 majority circuits, in a very strong way.
Ruling out weak PRFs with distPAC-learning (K., 2024)

Open Question
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for poly(n) size $\Lambda$-circuits, in the original PAC-learning model?

Left out so far is that CIKK16 actually invokes their implication from natural proofs to query learning using the existing natural proofs against $\text{AC}^0[p]$ by (Razborov-Smolensky, 1987)

One of the motivations of Open Question is to rule out **weak** PRFs in $\text{AC}^0[2]$ (Boyle et al., 2021)

DistPAC-learning is enough to rule out weak PRFs. Thus we invoke our theorem with Nisan’s natural proofs to rule out weak PRFs evaluable by depth-2 majority circuits, in a very strong way.

Weak PRFs often suffice for crypto applications (whereas strong PRFs are overkill). There’s a well motivated research direction to find the absolute minimum hardware (e.g. size of low-depth circuits) that compute wPRFs.
DistPAC-learning rules out weak PRFs. Thus we rule out encoded-input weak PRFs by depth-2 majority circuits, in a very strong way.
Ruling out weak PRFs with distPAC-learning (K., 2024)

Even with encoded inputs. Analogous to BIP+18

DistPAC-learning rules out weak PRFs. Thus we rule out encoded-input weak PRFs by depth-2 majority circuits, in a very strong way.
Ruling out weak PRFs with distPAC-learning (K., 2024)

Even with encoded inputs

Open Question
Let $\Lambda$ be any circuit class. Do Natural Proofs useful against $\Lambda$-circuits of size $\exp(n)$ imply polynomial time learning algorithms for $\poly(n)$ size $\Lambda$-circuits, in the original PAC-learning model?

One of the motivations of Open Question is to perhaps rule out conjecture weak PRFs in $\AC^0[2]$ (Boyle et al., 2021)

DistPAC-learning is enough to rule out weak PRFs. Thus we invoke our theorem with Nisan's natural proofs to rule out weak PRFs evaluable by depth-2 majority circuits, in a very strong way.

DistPAC-learning for $\AC^0[2]$ remains open!

CIKK16 actually invokes their implication from natural proofs using the existing natural proofs against $\AC^0[p]$ by (Razborov-Smolensky, 1987)
Core technique (K., 2024)

Exploit HOW the natural proofs works.

Correlation bounds for randomized communication protocols (we provide a new application of this)

Definition 1.2 (2-party norm). For $f : ([0,1]^n)^2 \rightarrow \{-1,1\}$, the 2-party norm of $f$ is defined as

$$R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \left[ \prod_{x_1, x_2 \in \{0,1\}} f(x_1^i, x_2^j) \right]$$

Recall informal theorem:

Any circuit class $\Lambda$ (size $s(n)$), which has a $g(n)$ lower bound via Nisan’s method, has a “Distributional PAC-learning” algorithm that runs in time $\exp(g^{-1}(s(n)))$. 
Core technique (K., 2024)
Exploit HOW the natural proofs works.

Correlation bounds for randomized communication protocols (we provide a new application of this)

Definition 1.2 (2-party norm). For $f : \{0, 1\}^n \rightarrow \{-1, 1\}$, the 2-party norm of $f$ is defined as

$$R_2(f) := \mathbb{E}_{x_1, x_2, y_1, y_2 \sim U_n} \left[ \prod_{x_1, x_2 \in \{0, 1\}} f(x_1^i, x_2^j) \right]$$

Evaluation Functions:
Eval($\pi_f, x$) $\rightarrow f(x)$
Induces a concept class:
$$C_{\text{Eval}} = \{\text{Eval}(\pi_f, \cdot) : \pi_f \in \{0, 1\}^{s(n)}\}$$

Concept distribution $\mu$ is thus thought of as over $\{0, 1\}^{s(n)}$
Correlation bounds for randomized communication protocols (we provide a new application of this)

**Definition 1.2** (2-party norm). For $f : \{0, 1\}^n \rightarrow \{-1, 1\}$, the 2-party norm of $f$ is defined as

$$
R_2(f) := \mathbb{E}_{x_1^1, x_1^2, x_2^1, x_2^2 \sim U_n} \left[ \prod_{i_1, i_2 \in \{0, 1\}} f(x_1^{i_1}, x_2^{i_2}) \right]
$$

Now let's think of R2 norms of evaluation functions
Fix $\mu$ (over concepts), $\rho$ (over inputs)

**Eval**:
- **Input**: string $r$, string $z$
- Take $\pi_f = \mu(r), x = \rho(z)$
- **Output**: $f(x)$

Equivalent to sampling from $\mu / \rho$ when $r / z$ are uniformly random strings
Definition 1.2 (2-party norm). For \( f : (\{0,1\}^n)^2 \to \{-1,1\} \), the 2-party norm of \( f \) is defined as

\[
R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \left[ \prod_{\epsilon_1, \epsilon_2 \in \{0,1\}} f(x_1^{\epsilon_1}, x_2^{\epsilon_2}) \right]
\]
Definition 1.2 (2-party norm). For $f : \{0, 1\}^n \rightarrow \{-1, 1\}$, the 2-party norm of $f$ is defined as

$$R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \prod_{\epsilon_1, \epsilon_2 \in \{0, 1\}} f(x_1^{\epsilon_1}, x_2^{\epsilon_2}) \tag{2}$$

“Hybrid argument”

Pick a two neighboring hybrid distributions at random

In expectation, over this random choice, Taking the parity of the bits distinguishes them

This is true whenever the 2-party norm of $\text{Eval}$ is large e.g., greater than $1/\text{poly}(n)$

Because the parity of random bits is in expectation 0, of course
Definition 1.2 (2-party norm). For \( f : \{0,1\}^n \rightarrow \{-1,1\} \), the 2-party norm of \( f \) is defined as

\[
R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \left[ \prod_{\xi_1, \xi_2 \in \{0,1\}} f(x_1^{\xi_1}, x_2^{\xi_2}) \right]
\]  

(2)

"Hybrid argument"

- Pick two neighboring hybrid distributions at random.
- In expectation, over this random choice, the parity of the bits distinguishes them.

This is true whenever the 2-party norm of \( \text{Eval} \) is large, e.g., greater than \( 1/poly(n) \).

Because the parity of random bits is in expectation 0, of course.

Can focus on the below 2 anyway.
The expected parity of \( H_4 \) is 0.
Definition 1.2 (2-party norm). For \( f : \{\{0, 1\}\}^2 \rightarrow \{-1, 1\} \), the 2-party norm of \( f \) is defined as

\[
R_2(f) := \mathbb{E}_{x_1, x_2, x_1, x_2 \sim U_n} \left[ \prod_{\xi_1, \xi_2 \in \{0, 1\}} f(x_1^{\xi_1}, x_2^{\xi_2}) \right]
\]  

(2)

"Hybrid argument"

Pick a two neighboring hybrid distributions at random

In expectation, over this random choice,
Taking the parity of the bits distinguishes them

This is true whenever the 2-party norm of Eval is large

\( \text{e.g., greater than } 1/\text{poly}(n) \)

Because the parity of random bits is in expectation 0, of course

Can focus on the below 2 anyway.
The expected parity of \( H_4 \) is 0.

Given these 3 bits, we can predict the next using the distinguisher (or argue directly)
Definition 1.2 (2-party norm). For $f : \{0, 1\}^n \times \{0, 1\}^n \to \{-1, 1\}$, the 2-party norm of $f$ is defined as

$$R_2(f) := \mathbb{E}_{x_1, x_2 \sim U_n} \left[ \prod_{\varepsilon_1, \varepsilon_2 \in \{0, 1\}} f(x_1^{\varepsilon_1}, x_2^{\varepsilon_2}) \right]$$

(2)

Pick two neighboring hybrid distributions at random

In expectation, over this random choice,

Taking the parity of the bits distinguishes them

This is true whenever the 2-party norm of Eval is large

e.g., greater than $1/poly(n)$

Because the parity of random bits is in expectation 0, of course

“Hybrid argument”

Correlation of $f$ with communication protocols with cost $c$ (w.r.t. uniform)

Can focus on the below 2 anyway.
The expected parity of $H_4$ is 0.

How do we know when 2-party norm is big?

**Theorem 1.8** (The correlation bound — [CT93, Raz00, VW07]). For every function $f : \{0, 1\}^n \times \{0, 1\}^n \to \{-1, 1\}$,

$$\text{Cor}(f, \Pi[2, c]) = \max_{\pi \in \Pi[2, c]} \mathbb{E}_x |f(x) \cdot \pi(x)| \leq 2^c \cdot R_2(f)^{1/4}$$

(3)

for $x$ uniformly distributed over $\{0, 1\}^n$.

$H_4 := \begin{cases} \text{Eval}(\pi_1, x_1) = f_{\pi_1}(x_1), \\ \pi_1 \sim U \\ x_1 \sim \rho \end{cases}$

$H_2 := \begin{cases} \text{Eval}(\pi_1, x_1) = f_{\pi_1}(x_1), \\ \pi_1 \sim U \\ x_1 \sim \rho \end{cases}$
Definition 1.2 (2-party norm). For \( f : \{0, 1\}^n \to \{-1, 1\} \), the 2-party norm of \( f \) is defined as

\[
R_2(f) := \mathbb{E}_{x_1^0, x_1^1, x_2^0, x_2^1 \sim U_n} \left[ \prod_{\xi_1, \xi_2 \in \{0, 1\}} f(x_1^{\xi_1}, x_2^{\xi_2}) \right]
\]  

(2)

Correlation of \( f \) with communication protocols with cost \( c \) (w.r.t. uniform)

How do we know when 2-party norm is big?

Theorem 1.8 (The correlation bound — [CT93, Raz00, VW07]). For every function \( f : \{0, 1\}^n \to \{-1, 1\} \),

\[
\text{Cor}(f, \Pi[2, c]) = \max_{\pi \in \Pi[2, c]} \mathbb{E} \left| f(x) \cdot \pi(x) \right| \leq 2^c \cdot R_2(f)^{1/4}
\]

(3)

for \( x \) uniformly distributed over \( \{0, 1\}^n \).
Definition 1.2 (2-party norm). For $f : \{0,1\}^n \rightarrow \{-1,1\}$, the 2-party norm of $f$ is defined as

$$R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \left[ \prod_{\varepsilon_1, \varepsilon_2 \in \{0,1\}} f(x_1^{\varepsilon_1}, x_2^{\varepsilon_2}) \right]$$

How do we know when 2-party norm is big?

Traditionally, Thm. 1.8 is used to prove that certain functions have little correlation with 2-party protocols (w.r.t. the uniform distribution over inputs)

By estimating a (low) $R_2$.

So use contrapositive:

When Eval correlates well with low-communication protocol, $R_2$ is large!
Definition 1.2 (2-party norm). For \( f : \{0,1\}^n \to \{-1,1\} \), the 2-party norm of \( f \) is defined as

\[
R_2(f) := \mathbb{E}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim U_n} \left[ \prod_{\varepsilon_1, \varepsilon_2 \in \{0,1\}} f(x_1^{\varepsilon_1}, x_2^{\varepsilon_2}) \right]
\]

Correlation of \( f \) with communication protocols with cost \( c \) (w.r.t. uniform)

How do we know when 2-party norm is big?

Theorem 1.8 (The correlation bound — [CT93, Raz00, VW07]). For every function \( f : \{0,1\}^n \to \{-1,1\} \),

\[
\text{Cor}(f, \Pi[2, c]) = \max_{\pi \in \Pi[2, c]} \mathbb{E} [f(x) \cdot \pi(x)] \leq 2^c \cdot R_2(f)^{1/4}
\]

for \( x \) uniformly distributed over \( \{0,1\}^n \).

Concept evaluation game

\[
\mathbb{E}_{\pi_f, x} \left[ \text{Eval}(\pi_f, x) \cdot b \right] \geq \gamma
\]

\[
\gamma \cdot 2^{-c} \leq R_2(\text{Eval})
\]

Teacher

\[\pi_f \sim \mu\]

Student

\[x \sim \rho\]
Actualy implementing this in the distributional PAC-Learning model

A randomized predictor with weak advantage $\approx \gamma 2^{-c}$

Where $\mu$ is a fixed distribution over concepts

**INPUT:** $z \sim \rho$

Sample $\langle x, f^{\text{real}}(x) \rangle \sim \text{Ex}(f, \rho)$

Sample $f^{\text{counter}} \sim \mu$.

Compute $f^{\text{counter}}(x), f^{\text{counter}}(z)$

**PREDICT:** $f^{\text{counter}}(x) \cdot f^{\text{counter}}(z) \cdot f^{\text{real}}(x) \cdot 1$
Where $\mu$ is a fixed distribution over concepts.

1) Apply sampling and testing to get a weak distPAC-learning algorithm that prints deterministic hypothesis circuits.

2) Apply boosting (e.g. Schapire, 1990) to derive a “strong” learner over many rounds.

Actually implementing this in the distributional PAC-Learning model

A randomized predictor with weak advantage $\approx \gamma^2 - c$
Example concept distributions

What can be evaluated with low communication?

Super simple example:

Distributions over decision trees given an “Anchor” tree

Evaluation function defined by Anchor tree reads from both the concept representation and the input

Hence, the sampling of the concept representation natural induces a randomized pruning of the anchor, i.e., a distribution over decision trees
Super simple example:

Distributions over decision trees given an “Anchor” tree

Evaluation function defined by Anchor tree reads from both the concept representation and the input

Hence, the sampling of the concept representation natural induces a randomized pruning of the anchor, i.e., a distribution over decision trees
Super simple example:

Distributions over decision trees given an “Anchor” tree

Evaluation function defined by Anchor tree reads from both the concept representation and the input

Hence, the sampling of the concept representation natural induces a randomized pruning of the anchor, i.e., a distribution over decision trees
Example concept distributions

What can be evaluated with low communication?

Super simple example:

Distributions over decision trees
given an “Anchor” tree

Evaluation function defined by
Anchor tree reads from both
the concept representation
and the input

Hence, the sampling of the concept
representation natural induces
a randomized pruning of the anchor,
i.e., a distribution over decision trees

Also:

“Organic” distributions over:

- Depth2 majority circuits
- Intersections of halfspaces
- DNFs
Future directions

Some obvious ones

• What other interesting “organic” distributions over concepts can be learned using this technique?

• Statistical study of distPAC-learning?

• distPAC-learning of AC0[2]?