Learning Noisy Parities

\[ \mathcal{E}_{0,1}^m \]

Unknown \( S \subseteq \{0,1\}^n \), we \( \mathcal{E}_{0,1}^m \)

**First NOISELESS**

Get several random

\[ x \in \mathcal{E}_{0,1}^m \] (uniformly)

along with

\[ f(x) = \bigoplus x_i = \langle x, w \rangle \in \mathbb{F}_2 \]

\[ \text{for } i \in S \]

Find \( S \).

with membership queries

ask for \( f(e_i) \) \( e_i = (0 \ldots 010\ldots 0) \)

\[ m \] queries

with random examples
If we get enough $x$'s so that
span of these $x$'s $= \mathbb{F}_2^n$
then can recover $w$ from
all the $\langle w, x \rangle$
(Gaussian elimination).

\[
\begin{bmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_n 
\end{bmatrix}
\begin{bmatrix}
  w \\
  1 
\end{bmatrix}
= \begin{bmatrix}
  b_1 \\
  b_2 \\
  \vdots \\
  b_n 
\end{bmatrix}
\]

Want it to have rank $n$.

\[
\Pr \left[ \text{random non-zero matrix in } \mathbb{F}_2^n \text{ has rank } n \right]
= \Pr \left[ \det(M) \neq 0 \right]
\]

\[
\text{Harsha: } 0.5^1 \\
\text{Vishnu: } \frac{\sqrt{n}}{2^{1/2}} \\
\text{Chaitanya: } 1 - \frac{n(n+1)}{2^{n+1}}
\]
\[
P_n\left[ \det(M) \neq 0 \right] = P_n\left[ \forall i = 1 \ldots n, \ni^{th} \text{ column } \notin \text{span}\left( \text{cols } 1 \ldots i-1 \right) \right]
\]
\[
= \left( 1 - \frac{1}{2^n} \right) \left( 1 - \frac{2}{2^n} \right) \left( 1 - \frac{4}{2^n} \right) 
\cdots \left( 1 - \frac{2^{i-1}}{2^n} \right) \cdots \left( 1 - \frac{2^{n-1}}{2^n} \right)
\]
\[
\approx 0.36 \text{ (possibly)}
\]

\[
P_n\left[ x_1, \ldots, x_m \text{ span dim } \leq n-1 \right]
\]

\[
\bigoplus = P_n\left[ \exists \text{ n-1 dim subspace } W \text{ s.t. all } x_i \in W \right] 
\leq \sum_{\text{dim}(n-1)} P_n\left[ \text{all } x_i \in W \right]
\]
\[ \leq \left( \frac{1}{2^m} \right) \cdot \#(n\text{-dim spaces}) \]

\[ \leq \frac{2^n}{2^m} \]

Taking \( m = n + O(1) \) makes this \( p_n = O(1) \).

**Noisy Case**

Unknown \( w \in \{0,1\}^n \).

Given many random examples \( x \in \{0,1\}^n \) and \( \langle x, w \rangle + b \in \mathbb{F}_2 \)

where \( b \in \{0,1\} \)

with \( p_n \{ b = 1 \} \leq 0.49 \).

How quickly can we find \( w \)?

How many examples are needed?
Coming up

1. Trivial $2^n \cdot \text{poly}(n)$ time alg.
2. poly$(n)$ examples suffice
3. $O(2^{n/\log n})$ time algorithm. (Blum-Kalai-Wasserman)

Won't see

poly$(n)$ examples + $2^{n/\log \log n}$ time suffice (Cubanski).

---

1 + 2:

Pick $M = \text{poly}(n)$

Get $x_i$ and

$y_i = \langle x_i, w \rangle + b_i$ for $i = 1, \ldots, M$.

For each $v \in \mathcal{F}^n$

check $N_v = \left( \# i \text{ s.t. } \langle x_i, v \rangle = y_i \right)$

claim $\exists$ exactly one $v$ for which
This number $N_V > (0.501) \cdot m$.

**Part 1**

For $V = W$

\[
\text{w.h.p. } N_W \geq 0.501 \cdot m.
\]

Proof

\[
N_W = m \cdot \left( \text{# i s.t. } b_i = 1 \right)
\]

\[
\geq m - 0.499m \quad \text{w.h.p.}
\]

**Part 2**

For $V \neq W$.

\[
P_n \left[ N_V \geq 0.501m \right] \ll \frac{1}{2^{2n}}.
\]

\[
\Rightarrow P_n \left[ \exists V \neq W \text{ s.t. } N_V > 0.501m \right] < \frac{2^n}{2^{2n}} = \frac{1}{2^n}.
\]

Proof

\[
P_n \left[ \text{#(i s.t. } \langle x_i, V \rangle = \langle x_i, W \rangle + b_i \rangle > 0.501m \right]
\]

\[
= P_n \left[ \text{#(i s.t. } \langle x_i, V - W \rangle = b_i \rangle > 0.501m \right]
\]

\[
= P_n \left[ \text{m uniform coin tosses give } > 0.501m \text{ heads} \right]
\]
\[ e^{-\Delta(m)} \]

**BKW plan**

Find \( x^* \in \mathcal{E}_0 \setminus B^m \)

and many different subsets

\( U_1, U_2, \ldots, U_t \subseteq [m] \)

such that \( |U_i| \leq t \)

\[ \sum_{i \in U_i} x_i = x^* \]

Then we get many noisy opinions about \( \langle x^*, w \rangle \)

and so we can get \( \langle x^*, w \rangle \) exactly.

---

Need \( t \) small, \( t \) big.

**Claim** If \( b_1, b_2, \ldots, b_t \in \mathcal{E}_0 \setminus B^m \)

with \( b_i \) all independent
\[
\ln(b_i = 1) = \eta
\]

then.

\[
P_{\eta} \left( \bigoplus_{i=1}^t b_i = 1 \right) = \frac{1 - (1 - 2\eta)^t}{2}
\]

Proof

\[
\sum_{J \in [t] \atop |J| \text{ odd}} \eta^{|J|} (1 - \eta)^{|J| - 1}
\]

= \sum_{n, \text{ odd}} \eta^n (1 - \eta)^{t - n}

= \frac{(\eta + (1 - \eta))^t - (\eta - (1 - \eta)^t)}{2}

If \( \eta = \frac{1 - e}{2} \), then noise for

sum of \( \sum_{i=1}^t b_i = \frac{1 - et}{2} \).

Suppose we could find for every

\( x^* = 50, 13^n \) 2 different ways
of summing up $t$-subsets of the $x_i$ to $x^*$, then we get a noiseless evaluation of $\langle x^*, w \rangle$ provided

\[ l \gg \text{poly} \left( \frac{1}{(-2x)^t} \right) \]

\[ N = 0.49 \Rightarrow 2^{N(t)} \]

\[ l \gg 2^{N(t)} \]

---

**BKW** Given $x_1, \ldots, x_{2^{\frac{n}{3} \log n}}$ uniformly random

we can find $\leq n/\log n$ vectors in time $2^{O(1/\log n)}$ that sum up to any desired vector.

\[ \text{Noise prob goes from } O(1) \text{ to } \frac{1}{2} - 2^{-n/\log n} \]

---

**Fact** Given $x_1, \ldots, x_{2^m}$ random vectors
for any $x^* \in \{0,1\}^n$, is some subset of $\leq 5n$ of the $x_i$ that seem up to $x^*$.

Don't know how to find this in time $2^{O(n)}$.

NICE QUESTION!

Take all the $x_1, \ldots, x_m$.

Look at the first $k$ bits of each of them.

This puts the $x_i$ into $2^k$ buckets: $S_1, S_2, \ldots, S_{2^k}$.

For each $j \in \{2^k\}$, let $Z_j$ be some element of $S_j$.

Produce the vectors $T_j = \{Z_j + w : w \in S_j \setminus \mathbb{Z}_2^k\}$

EltS of $T_j$:
Independent
First k bits = 0.
Last n-k bits uniform.
Each elt is a sum of 2 of the x_i's.

\( U \) of \( m \) examples

we get \( m - 2^k \) new examples

each new example is a sum of two old examples, and is uniform conditioned on the first \( k \) bits being all 0.

Can repeat this \( \frac{n-1}{k} \) times to get, from \( m \) examples, \( m - \frac{n-1}{k} \) examples whose first \( n-1 \) bits are 0 and the last bit is uniform.
each new example is a sum of $\frac{n}{k}$ original examples.

Want $2^{\frac{n}{k}} = \frac{n}{\log n}$

\[
\frac{n}{k} \approx \log n - \log \log n
\]

\[
k \approx \frac{n}{\log n - \log \log n}
\]

\[
\frac{n}{k} \approx \log n
\]

\[
k \approx \frac{n}{\log n}
\]

Take $m \gg \frac{n-1}{k} \cdot 2^k \approx 2^{\frac{(n-1)}{\log n}}$

Then we get $\frac{n}{\log n}$ original examples summing up to have first $n-1$ bits 0.
So we express $en$ as a sum of \( \frac{n}{\log n} \) vectors in time 

\[ \text{poly} \left( 2^k \cdot \frac{n}{k} \right) = 2^{O(n/\log n)} \]