LAST TIME:

- Definition of Weak Learning
- ADABoost: Training: $\{(x_1, y_1), \dots, (x_m, y_m)\}$.

$D_1(i) = 1/m$

For $t = 1, \dots, T$,

- Use WL under D_t to get h_t
- Let $\varepsilon_t = \mathbb{P}_{x \sim D_t}[h_t(x) \neq y_t]$; let $\gamma_t = 1/2 - \varepsilon_t$ // $\gamma_t \geq \gamma$ (or $\varepsilon_t \leq \frac{1}{2} - \gamma$)
- Choose $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
- $D_{t+1}(i) = D_t(i) e^{-\alpha_t y_t h_t(x_i)} / Z_{t+1}$.

Output: $\cancel{H} = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(\cdot)\right)$ as classifier.

Theorem: Provided $T \geq \frac{1}{2\gamma^2} \log(2m)$, then training error of H on the sample is 0.

Suppose that WL (weak learning algorithm) always outputs a hypothesis from some class H, whose VC-dimension is d.

$$TH_{n,T} = \left\{ \text{sign}\left(\sum_{i=1}^{T} \alpha_i h_i(\cdot)\right) \,\middle|\, h_i \in H_n,\ \alpha_i \in \mathbb{R} \right\}.$$

what is VCD$(TH_{n,T})$?

(exercise): $\text{VCD}(TH_{n,T}) \leq c \cdot T \cdot d \log T$.

(Compute the growth function of the composition)

[Provided $m \geq \frac{k_0}{\varepsilon}\left(\log \frac{1}{\delta} + c \cdot T d \log T \log \frac{1}{\varepsilon}\right)$ then the hypothesis output by AdaBoost has error $\leq \varepsilon$ w.p. $\geq 1 - \delta$.

[Appealing to VC Theorem].

$$m \geq \left(\frac{k_0}{\varepsilon} \cdot \frac{1}{2\gamma^2} \log(2m) d \log\left(\frac{1}{2\gamma^2} \log(2m) d\right)\right) + \frac{k_0}{\varepsilon} \log \frac{1}{\delta}$$

$\frac{m}{2} \geq a \cdot \log(bm)$

$f(x) = x - a \log bx \geq 0 \qquad x \geq c \cdot a \log a$.

PAC Learning:

- Several concept classes that are PAC learnable.
  - CONJUNCTIONS, HALFSPACES, 3-CNF, ...
- Occam's Razor / VC Theory

  Consistent Learning $\Rightarrow$ PAC learning

  (Settles information-theoretic / statistical complexity of learning).
- Weak & strong learning are equivalent.
- Are there hard learning problems for computational reasons?
  - If $NP \neq RP$, then the class 3-term DNF is not (properly) PAC-learnable using 3-term DNF.
- Is there a concept class that is hard to learn even improperly (algorithm can output a hypothesis from a larger class.).

  If a certain assumption related to RSA holds, then then there exists C that is not efficiently PAC-learnable.

Let p, q be two large primes, both of them 2048 bits long, and of the form $3k+2$.

Let $N = pq$ [Factoring N is believed to be hard.]

(All arithmetic is mod N).

$\varphi(N) = (p-1)(q-1)$ (Euler's totient function)

$\mathbb{Z}_N^* = \{ i \mid 0 < i < N, \gcd(i, N) = 1 \}$, $|\mathbb{Z}_N^*| = \varphi(N)$.

(Group under multiplication mod N)

We know that $3 \nmid \varphi(N)$

Let $f_N : \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^*$, $f_N(y) = y^3 \pmod N$

<u>Claim</u> : f_N is a bijection.

Euler's theorem : $\forall a \in \mathbb{Z}_N^*$, $a^{\varphi(N)} \equiv 1 \pmod N$

<u>Proof</u> : Since $3 \nmid \varphi(N)$, $\gcd(3, \varphi(N)) = 1$

$\exists d \in \mathbb{Z}$, s.t. $3d \equiv 1 \pmod{\varphi(N)}$ $\longrightarrow$ $3d = 1 + d'\varphi(N)$

$\boxed{\gcd(a,b) = g : \exists m,n \in \mathbb{Z},\ g = m.a + n.b.}$

$$(f_N(y))^d \equiv (y^3)^d \equiv y^{3d} \equiv y^{1+d'\varphi(N)} \equiv y \pmod N$$

$$f_N^{-1} : \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^*, \quad f_N^{-1}(x) = x^d \pmod N$$

---

f_N is easy to compute using N. $\longrightarrow$ [PUBLIC KEY]

f_N^{-1} is hard to compute without knowing d $\rightarrow$ [PRIVATE KEY]

<u>Discrete Cuberoot Problem</u> : Two primes of the form $(3k+2)$, p & q are chosen, $N = p \cdot q$, $\varphi(N) = (p-1)(q-1)$, $3 \nmid \varphi(N)$. Given N and $x \in \mathbb{Z}_N^*$ as input, output y, s.t. $y^3 \equiv x \pmod N$.

(Easy to solve if we can factor N)

DCRA Assumption :

For every polynomial $r(\cdot)$, $\not\exists$ algorithm that runs in time $r(n)$ and that on input N & x, where N is as defined above, $x \in \mathbb{Z}_N^*$ at random and n = # bits in N) outputs y s.t. w.p. $\geq 1/r(n)$, y satisfies $y^3 \equiv x \pmod N$.

The probability is over random draws of p, q, x and any internal randomization of the algorithm.

Given $(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)$, $x_i, y_i \in \mathbb{Z}_N^*$, $x_i \sim \mathrm{Unif}(\mathbb{Z}_N^*)$ and y_i s.t. $y_i^3 \equiv x_i \pmod N$.

Can we output $h : \mathbb{Z}_N^* \to \mathbb{Z}_N^*$ s.t. w.p. $\geq 1-\delta$

$$\mathbb{P}_{x \sim \mathrm{Unif}(\mathbb{Z}_n^*)}\left[h(x) \neq y, \text{ where } y^3 \equiv x \pmod N\right] \leq \varepsilon.$$

---

y is also uniform on $\mathbb{Z}_N^*$, as f_N is a bijection.

Generating data is easy ; Pick $y_i \sim \mathrm{Unif}(\mathbb{Z}_N^*)$, $x_i \equiv y_i^3 \pmod N$

$$f_N^{-1} : \mathbb{Z}_N^* \longrightarrow \mathbb{Z}_N^* \;;\quad f_N^{-1}(x) \equiv y \quad \text{s.t. } y^3 \equiv x \pmod N$$

n = # bits in N.

$f_{N,i}^{-1} : \mathbb{Z}_N^* \to \{0,1\}$ i^{th} bit of y

$\{f_{N,1}^{-1}, \ldots, f_{N,n}^{-1}\} \longleftarrow$ at least one of these functions must be hard to learn.

Otherwise set $\varepsilon = 1/n^2$, $\delta = 1/n^2$.

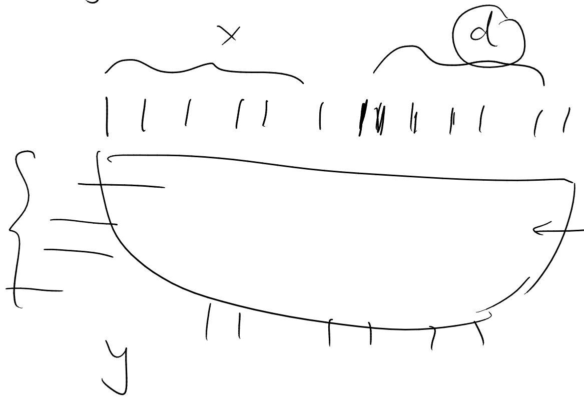
suppose we learn $h_1, \ldots, h_n$, w.p. $\geq 1 - n.\delta \geq 1 - \frac{1}{n}$
all h_i satisfy $err(h_i) \leq 1/n^2$

For x is picked at random, w.p. $\geq 1 - \frac{1}{n}$,

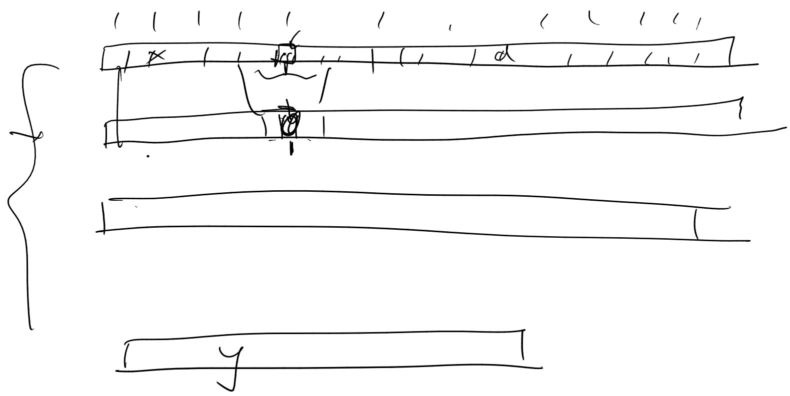
$$h_1(x)\, h_2(x) \cdots h_n(x) = y, \quad \text{s.t. } y^3 \equiv x \pmod N$$

$f_N^{-1} \leftarrow d$

Turing Machine: $M(x, d) \to y$ in polynomial time s.t. $y^3 \equiv x \pmod N$



Polynomial-size circuit.



$\exists$ poly-nomial-size circuit that computes $f_N^{-1}(x)$