# Computational complexity of learning

- ERM for inputs where $\widehat{err}_S(h^*) = 0$ is often computationally tractable

- ERM for general inputs is often computationally hard

- Sometimes, ERM is computationally hard for $H$, but there exist $H' \supseteq H$ for which ERM is computationally easy

- Computational hardness of learning $\neq$ Computational hardness of ERM

- Computational hardness of learning often relies on

# Cryptographic assumptions

## ERM when $\widehat{err}_S(h^*) = 0$

- Halfspace with normal $a = (a_1, \ldots, a_d)$

$$h_a(x) = \text{sign}\left(a_1 x_1 + a_2 x_2 + \cdots + a_d x_d\right)$$

- $H_d = \{ h : h_a, a \in \mathbb{R}^d, a \neq 0 \}$

- Suppose we are given a sample $S = ((x_1, y_1) \ldots (x_m, y_m))$ where $x_i \in \mathbb{R}^d$, $y_i \in \{+1, -1\}$. such that $\widehat{err}_S(h^*) = 0$ for some $h^* \in H_d$.

- We can find $\hat{h} \in H_d$ such that $\widehat{err}_S(\hat{h}) = 0$ using linear programming.

- Suppose $\hat{h}$ is parameterized by $a \in \mathbb{R}^d$

- Find $a \in \mathbb{R}^d$

  s.t. $y_1 a^T x_1 > 0$

  $y_2 a^T x_2 > 0$

  $\vdots$

  $y_m a^T x_m > 0$

- Linear program:

Find $a \in \mathbb{R}^d$

so to $y_1 a^T x_1 \geq 1$

$y_2 a^T x_2 \geq 1$

$\vdots$

$y_m a^T x_m \geq 1$

- The choice of right hand side $= 1$ is arbitrary. Any positive real number would work.

- A solution of the linear program can be found in time that is polynomial

in both $d$ and $m$.

---

## ERM in general

- ERM for class $H_d$ is NP-hard
- NP-hard means: if there is polynomial algorithm for ERM for $H_d$ then there exist polynomial algorithms for <u>all</u> NP-complete problems.
- We reduce vertex cover to ERM for $H_d$

- Vertex cover problem:

  Input: Graph $G = (V, E)$ and integer $k$.

  Output: Yes / No : Is there a vertex cover $U \subseteq V$ such that $|U| \leq k$ ?

- Reduction: Given an input to vertex cover problem we construct an input to ERM for $H_d$ problem.
- Let $r = |E|$ and $n = |V|$
- Let $V = \{1, 2, \ldots, n\}$
- We define $m = 2r + n$ and $d = 2n + 1$

and

- Let $e_{i,j} \in \mathbb{R}^{2n}$,

$$e_{i,j} = (0,\dots,0,1,0,\dots,0,1,0\dots0,1)$$

(1 at position $i$, at position $j$, at position $n+i$)

- The labeled sample consists

1) $(e_{i,n+i}, 1)$ for every vertex $i \in V$

($e_{i,n+i}$: features; $1$: label)

2) $(e_{i,j}, 0)$, $(e_{n+i,n+j}, 0)$ for every edge $\{i,j\} \in E$

Lemma: There exists $h \in H_d$ with $\widehat{err}_S(h) \le \frac{k}{2r+m}$ if and

only if there exists vertex cover $U \subseteq V$ of size $|U| \leq k$.

Proof:

- Suppose $h \in H_d$,
  $h(x) = \text{sign}(a^T x)$
- We construct $U \subseteq V$
  as follows
  $U = \{ i \in V : a^T e_{i,n+i} < 0$
  or $\exists j \in V : j > i,\ a^T e_{i,j} \geq 0$
  or $\exists j \in V : j > i,\ a^T e_{n+i,n+j} \geq 0 \}$
- If $\widehat{err}_S(h) \leq \frac{k}{2r+n}$ then
  $|U| < k$

$|U| = k$

- We need to verify that $U$ is a vertex cover.
- Consider an edge $\{i,j\} \in E$, $i < j$
- It suffices to show that if

$$q^T e_{i,n+i} \geq 0 \quad \text{and} \quad q^T e_{j,n+j} \geq 0$$

then $i \in U$.

- $q^T e_{i,n+i} \geq 0$ and $q^T e_{j,n+j} \geq 0$ means that

$$q_i + q_{n+i} + qd \geq 0$$

$$q_j + q_{n+j} + qd \geq 0$$

- Sum the two inequalities:

$$q_i + q_j + q_{n+i} + q_{n+j} + 2qd \geq 0$$

- Thus

$$a_i + a_j + a_d \geq 0$$

or

$$a_{n+i} + a_{n+j} + a_d \geq 0$$

- Thus $a^T e_{i,j} \geq 0$ or $a^T e_{n+i,n+j} \geq 0$

- Since $i < j$, $i \in U$.

---

- Suppose $U \subseteq V$ is a vertex cover of size $|U| \leq k$

- We construct $h \in H_d$ such that $\widehat{err}_S(h) \leq \frac{k}{2r+n}$.

- Define

For $i = 1, \ldots, n$

$-1$ if $i \in U$

$$a_i = a_{i+n} = \begin{cases} \\ +1 \text{ if } i \notin U \end{cases}$$

$$ad = -1$$

- $h(x) = \text{sign}(a^T x)$

- If $\{i,j\} \in E$ is edge then $i \in U$ or $j \in U$.

- Therefore,

$$a^T e_{i,j} = a^T e_{n+i,n+j} = \begin{cases} -3 \text{ if } i \in U, j \in U \\ -1 \text{ if } i \in U, j \notin U \\ -1 \text{ if } i \notin U, j \in U \end{cases}$$

- So $h$ does not make

any mistake on the examples corresponding to edges

- $$a^T e_{i,n+i} = \begin{cases} 1 & \text{if } i \notin U \\ -1 & \text{if } i \in U \end{cases}$$

- So $h$ makes mistake only on examples corresponding to vertices in $U$.

- Therefore
$$\widehat{err_S}(h) = \frac{|U|}{2r+n} \leq \frac{k}{2r+n}.$$

Corrolary: ERM for $H_d$ is an NP-hard problem

## 3-term DNF

- $X = \{0,1\}^d$, $Y = \{0,1\}$
- Conjuction $h: X \to Y$

$$f(x) = x_{i_1} \wedge x_{i_2} \cdots \wedge x_{i_k} \wedge \bar{x}_{j_1} \wedge \cdots \wedge \bar{x}_{j_r}$$

- 3-term DNF

$$h(x) = f_1(x) \vee f_2(x) \vee f_3(x)$$

- $H_{3,d} = \{h : h \text{ is a 3-term DNF}\}$

- ERM for $H_{3,d}$ is computationally hard even when $\widehat{err}_S(h^*) = 0$ for some $h^* \in H_{3,d}$.

- Note $\vee$ distributes over $\wedge$:

$$f_1 \vee f_2 \vee f_3 = \bigwedge_{\substack{a \in f_1 \\ b \in f_2 \\ c \in f_3}} (a \vee b \vee c)$$

where $a, b, c \in \{x_1, \bar{x}_1, x_2, \bar{x}_2, \ldots, x_d, \bar{x}_d\}$ are "literals" (i.e.

Consider a mapping

- Consider a [illegible] $\psi : \{0,1\}^d \to \{0,1\}^{(2d)^3}$

where

$$\psi(x_1 \dots x_d) = (Y_{111} \cdots Y_{2d,2d,2d})$$

$$Y_{i,j,k} = x_i^{c_i} \vee x_j^{c_j} \vee x_k^{c_k}$$

$c_i, c_j, c_k$ is either a negation or not.

- We can think of $\psi$ as "feature mapping" that creates new features from old ones.

- Consider

$$H_{(2d)^3} = \{h : h \text{ is a conjuction}\}$$

- We can use ERM for $H_{(2d)^3}$

- Note that $\hat{h}$ found by ERM for $H_{(2d)^3}$ might not be representable by a 3-term DNF.

- Sample complexity of PAC learning $H_{(2d)^3}$ is roughly

$$O\left(\frac{d^3 \log(1/\delta)}{\varepsilon}\right)$$

- Sample complexity of PAC learning $H_{3,d}$ is only $O\left(\frac{d \log(1/\delta)}{\epsilon}\right)$

- We traded off polynomial time complexity for slightly worse sample complexity.

---

## Cryptography

**Definition:** Family $F = \{f_k : k \in K\}$ of functions, $f_k : \{0,1\}^* \to \{0,1\}^*$, is a trapdoor function family

1) Given $x \in \{0,1\}^n$, it is

computationally easy to compute $f_k(x)$

2) Given $y \in \{0,1\}^n$, it is computationally hard to find $x$ such that $f_k(x) = y$

3) Given a secret $s_k$ and $y \in \{0,1\}^n$ it is easy to find $x$ such that $f_k(x) = y$

---

- Public key cryptography constructs trapdoor functions

- Learning trapdoor function is by definition hard.