Probably Approximately Correct Learning

Pei-Yuan Wu

Dept. Electrical Engineering

National Taiwan University

The model achieves 5% error on 1000 training data, what can we say about the testing error?

The testing error on 500 testing samples is 8%.

Maybe it's just because it gets lucky on the testing data.

How many data is needed to train this model?

Paper X says they successfully train the model with dataset Y of 10000 samples.

Can we have precise statements with theoretical guarantees?

Outline

- PAC Learning Framework
  - ➢ Training error v.s. generalization error
  - ➢ Sample complexity for axis-aligned rectangle concepts.
  - ➢ Sample complexity for finitely many hypotheses (consistent/inconsistent cases)
- Rademacher Complexity
  - ➢ Loss functions associated to hypothesis set
  - ➢ Rademacher complexity and geometrical interpretation
  - ➢ Generalization bounds for binary/multi-class classifiers.
  - ➢ Rademacher complexity for fully-connected neural network
- Growth Function and VC Dimension
  - ➢ Growth function, shattering, VC dimension
  - ➢ Generalization bounds

PAC Learning Framework

Motivation

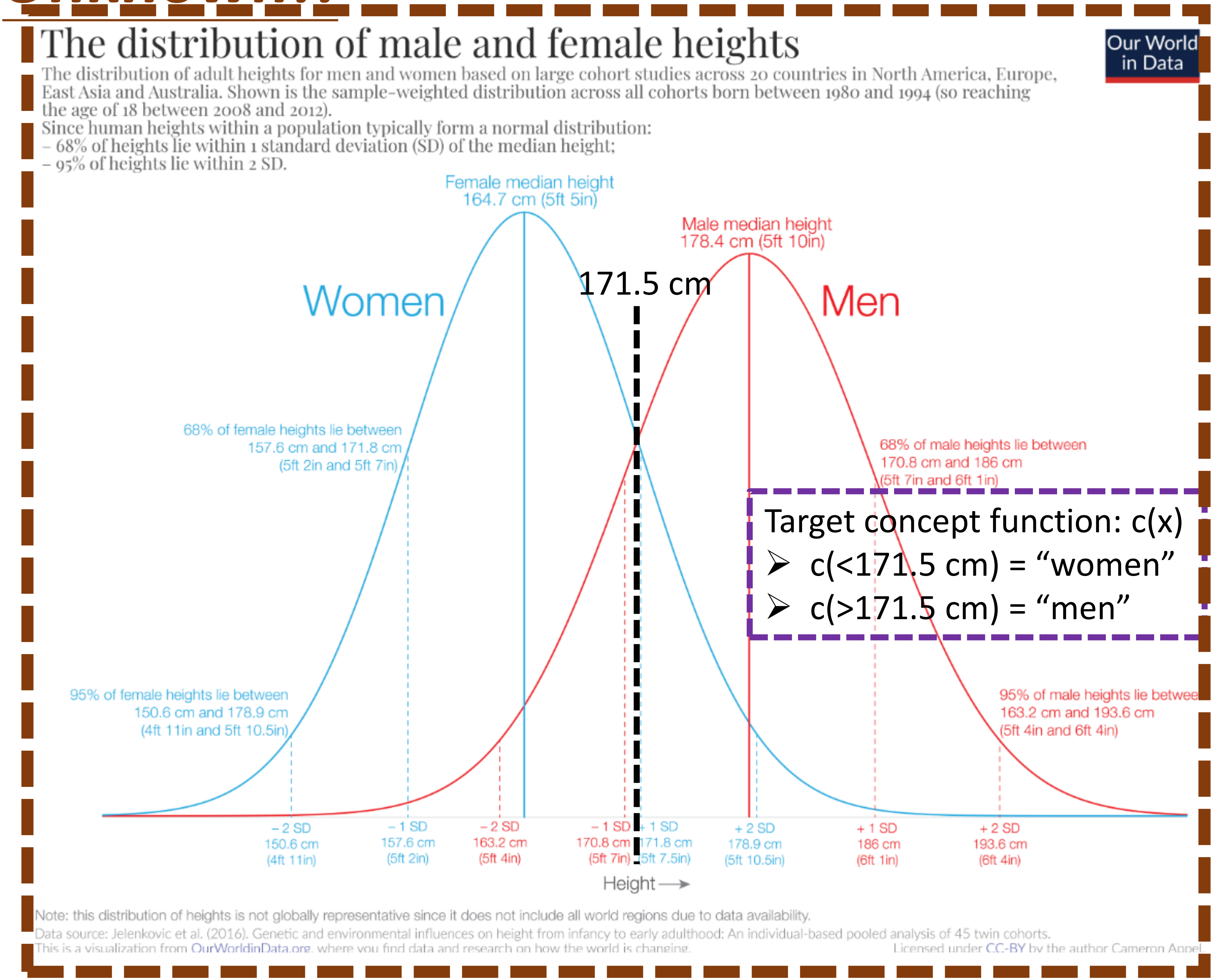
- Given the *training set*, a *learning algorithm* generates a *hypothesis*.
- Run *hypothesis* on the *test set*. The results say *something* about how *good our hypothesis is*.
    - ➢ How much do the *results really tell you*?
    - ➢ Can we be *certain* about how the learning algorithm *generalizes*?
        - ✓ We would have to see *all the examples*. (Not practical)
- Insight: Introduce *probabilities to measure degree of certainty and correctness*. (Valiant 1984)

Computational Learning Theory

- Computational learning theory is a *mathematical* and *theoretical* field related to *analysis* of machine learning *algorithms*.
- We need to seek theory to relate:
  - ➢Probability of successful learning
  - ➢Number of training examples
  - ➢Complexity of hypothesis space
  - ➢Accuracy to which target function is approximated

- Want to use height to distinguish men and women
  - ➢ Training and testing data drawn from the same distribution.
- Can never be absolutely certain that we have learned correctly our target (hidden) concept function.
  - ➢ There is a non-zero chance that, **so far**, we have only seen a sequence of bad examples (E.g., relatively tall women and relatively short men)
- It's generally highly unlikely to see a long series of bad examples!

Unknown!!



The distribution of male and female heights

The distribution of adult heights for men and women based on large cohort studies across 20 countries in North America, Europe, East Asia and Australia. Shown is the sample-weighted distribution across all cohorts born between 1980 and 1994 (so reaching the age of 18 between 2008 and 2012).

Since human heights within a population typically form a normal distribution:
– 68% of heights lie within 1 standard deviation (SD) of the median height;
– 95% of heights lie within 2 SD.

Target concept function: c(x)
- ➢ c(<171.5 cm) = "women"
- ➢ c(>171.5 cm) = "men"

Note: this distribution of heights is not globally representative since it does not include all world regions due to data availability.
Data source: Jelenkovic et al. (2016). Genetic and environmental influences on height from infancy to early adulthood: An individual-based pooled analysis of 45 twin cohorts.
This is a visualization from OurWorldinData.org, where you find data and research on how the world is changing.

https://ourworldindata.org/human-height

Probably Approximately Correct Learning

- The learner receives samples and must select a generalization function (hypothesis) from a certain class of possible functions.
- With high probability an (efficient) learning algorithm will find a hypothesis that is approximately identical to the hidden target concept.
    - ➢ Seriously wrong hypotheses can be ruled out almost certainly (with high probability) using a "small" number of examples
    - ➢ Any hypothesis that is consistent with a significantly large set of training examples is unlikely to be seriously wrong: it must be probably approximately correct (PAC).
    - ➢ Any (efficient) algorithm that returns hypotheses that are PAC is called a PAC-learning algorithm. (Formal definition to be introduced later)

PAC Learning Model

- Denote
  - ➢ $\mathcal{X}$: The set of all possible examples or instances, also referred as input space.
  - ➢ $\mathcal{Y}$: The set of all possible labels or target values.
    - ✓ For introductory purposes, assume $\mathcal{Y} = \{-1, +1\}$ (binary classification)
  - ➢ Concept $c: \mathcal{X} \to \mathcal{Y}$:
    - ✓ If $\mathcal{Y} = \{-1, +1\}$, we can identify c as the subset of $\mathcal{X}$ over which it takes value 1.
  - ➢ Concept class C: A set of concepts.
- Learning problem formulation: A learner
  - ➢ Considers a fixed set H of possible concepts, also referred as hypothesis set.
  - ➢ Receives a sample $S = (x_1, \dots, x_m)$ of m examples drawn i.i.d. according to some fixed but unknown distribution D, as well as the labels $(c(x_1), \dots, c(x_m))$ based on a fixed but unknown target concept $c \in C$.
  - ➢ Uses the labeled sample S to select a hypothesis $h_S \in H$ that has a small generalization error w.r.t. the target concept c.

What do we refer by generalization error?

Generalization Error v.s. Empirical Error

Definition: Generalization error

Given a hypothesis $h \in H$, a target concept $c \in C$, and an underlying distribution D, the generalization error (a.k.a. true error, risk) of h is defined as

$$\mathcal{R}(h) = \mathbb{P}_{x \sim D}[h(x) \neq c(x)] = \mathbb{E}_{x \sim D}\left[1_{h(x) \neq c(x)}\right]$$

Not accessible for the learner

Definition: Empirical error

Given a hypothesis $h \in H$, a target concept $c \in C$, and a sample $S = (x_1, \ldots, x_m)$, the empirical error or risk of h is defined as

$$\hat{\mathcal{R}}_S(h) = \frac{1}{m} \sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}$$

Accessible for the learner

Remark:

Empirical error is an unbiased estimate of generalization error

$$\mathbb{E}_{S \sim D^m}\left[\hat{\mathcal{R}}_S(h)\right] = \mathcal{R}(h)$$

PAC Framework

Definition: PAC-learning

A concept class C is said to be PAC-learnable if there exists an algorithm $\mathbb{A}$ and a polynomial function $poly(\cdot,\cdot)$ such that for any $\epsilon > 0$ and $\delta > 0$, for all distributions D on $\mathcal{X}$, and for any target concept $c \in C$, the following holds for any sample size $m \geq poly\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$

$$\mathbb{P}_{S \sim D^m}[\mathcal{R}(h_S) \leq \epsilon] \geq 1 - \delta$$

where $h_S \in H$ is the hypothesis learned by $\mathbb{A}$ from sample S. We say $\mathbb{A}$ is a PAC-learning algorithm for C.

Remark:

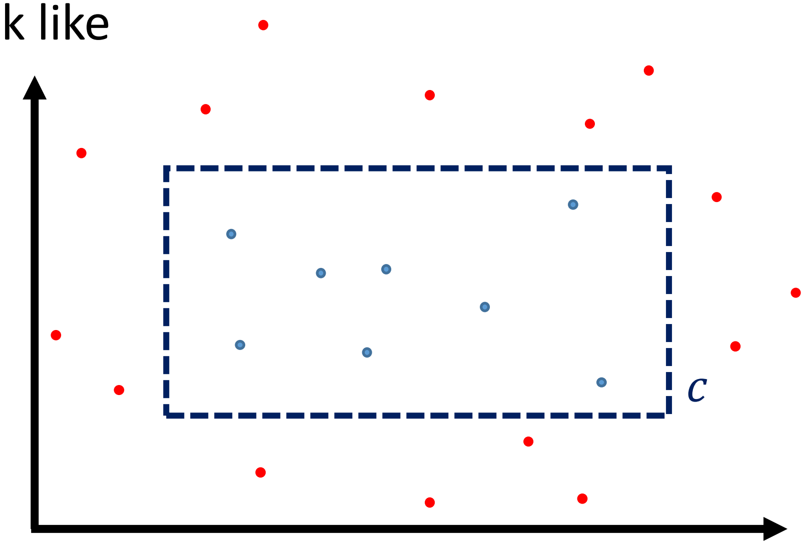
- The hypothesis returned by PAC-learning algorithm $\mathbb{A}$ is
  - ➢ Approximately correct (generalization error at most ϵ), with
  - ➢ High probability (at least $1 - \delta$ confidence), after observing
  - ➢ sufficiently many samples (polynomial in $\frac{1}{\epsilon}$ and $\frac{1}{\delta}$)
- PAC framework is a distribution-free model
  - ➢ No particular assumption on the distribution D from which examples are drawn.
- Stationarity assumption: Training set and test sets are drawn from the same distribution.
- PAC deals with the learnability for a concept class C and not a particular concept c.
  - ➢ Assume concept class C is known to learner, while the target concept $c \in C$ is unknown.

Example: Learning axis-aligned rectangles

- Axis-aligned rectangle concept class:
  - Input space $\mathcal{X} = \mathbb{R}^2$
  - $\mathcal{Y} = \{-1, +1\}$
  - Concept class C: Collection of all axis-aligned rectangles.

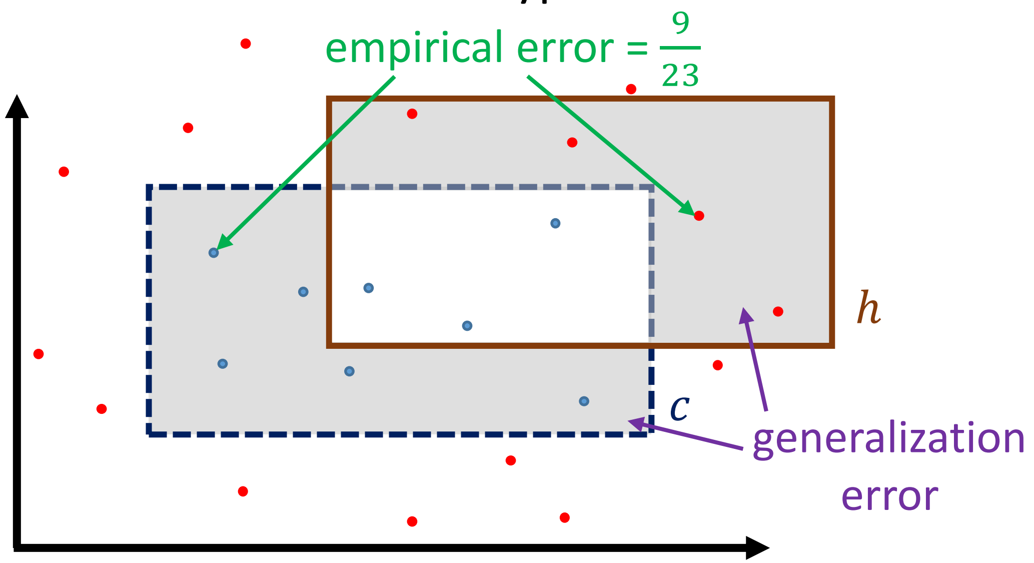
Is C PAC-learnable?

- For a specific concept $c \in C$, a sample S may look like



If only S is observed, how do we guess c?

- Generalization error of hypothesis h

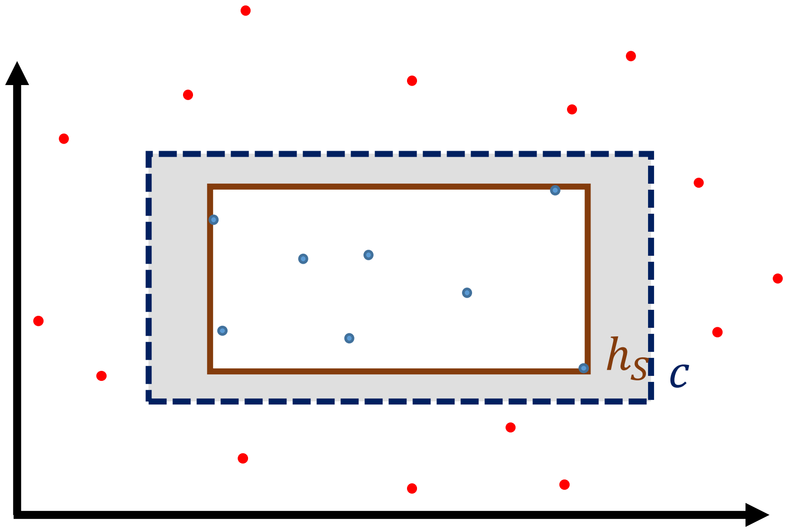


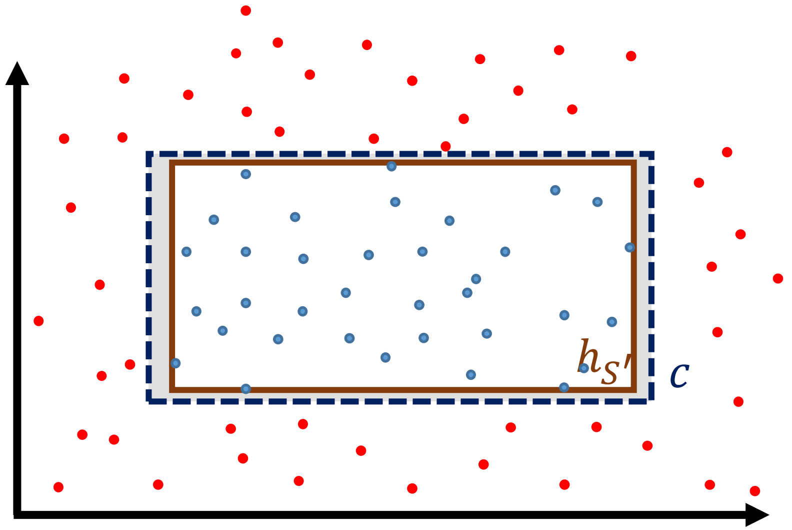
Example: Learning axis-aligned rectangles

- Consider the closure algorithm $\mathbb{A}$:
  - ➢ Given sample S, return h_S as the smallest rectangle consistent with S.
  - ➢ By definition, h_S is a subset of c.

- The generalization error is due to positive instances in S not occupying the inner edge of c (grey area).

- If one takes more instances, new instances may occupy the previously grey areas, leading to smaller generalization error.





If we randomly draw m instances, how unlikely will $R(h_S) > \epsilon$?

Example: Learning axis-aligned rectangles

- If $D(c) < \epsilon$, then $\mathcal{R}(h_S) = D(c - h_S) \leq D(c) < \epsilon$.
- Else, consider four rectangles 🟥🟨🟩🟦 along the inner edges of c

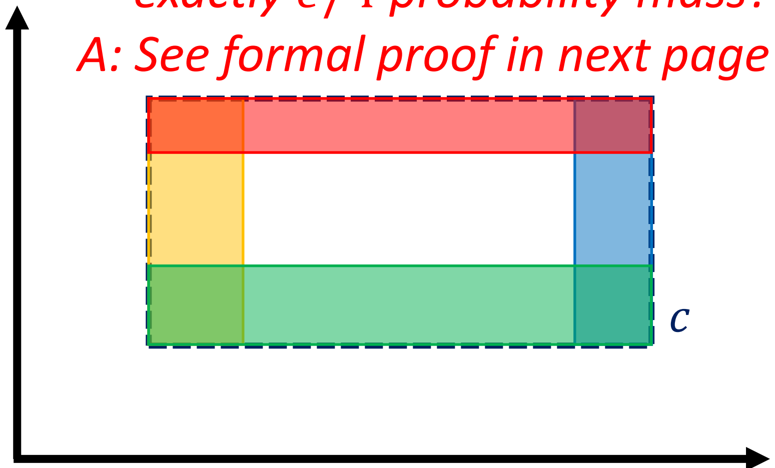
$D($🟥$) = \epsilon/4$

$D($🟩$) = \epsilon/4$

$D($🟨$) = \epsilon/4$ $D($🟦$) = \epsilon/4$

Q: What if you cannot find rectangles with exactly $\epsilon/4$ probability mass?

A: See formal proof in next page



- Let S be a sample of m randomly drawn instances
  - ➢ If S coincides with all four rectangles 🟥🟨🟩🟦, then $\mathcal{R}(h_S) \leq \epsilon$
  - ➢ How likely will things go wrong?

$$\mathbb{P}_{S\sim D^m}[S \cap 🟥 = \emptyset] = (1 - \epsilon/4)^m$$
$$\mathbb{P}_{S\sim D^m}[S \cap 🟨 = \emptyset] = (1 - \epsilon/4)^m$$
$$\mathbb{P}_{S\sim D^m}[S \cap 🟩 = \emptyset] = (1 - \epsilon/4)^m$$
$$\mathbb{P}_{S\sim D^m}[S \cap 🟦 = \emptyset] = (1 - \epsilon/4)^m$$

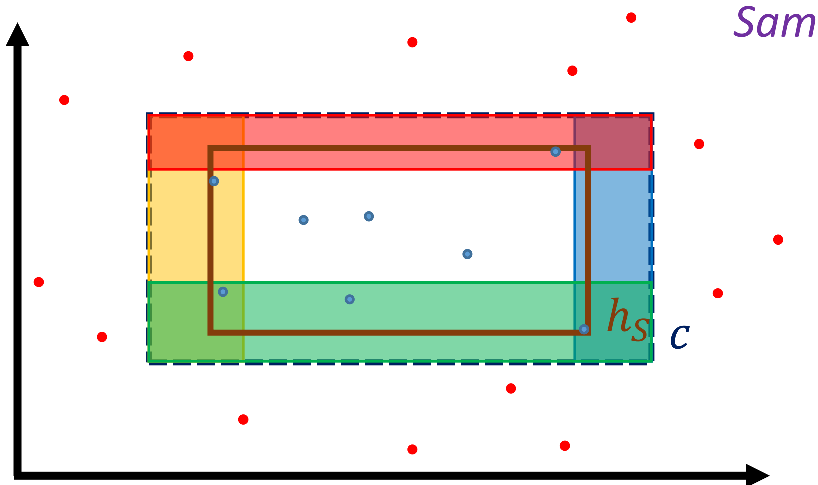
✓ Probability of things going wrong at most

$$4(1 - \epsilon/4)^m \leq 4e^{-m\epsilon/4}$$

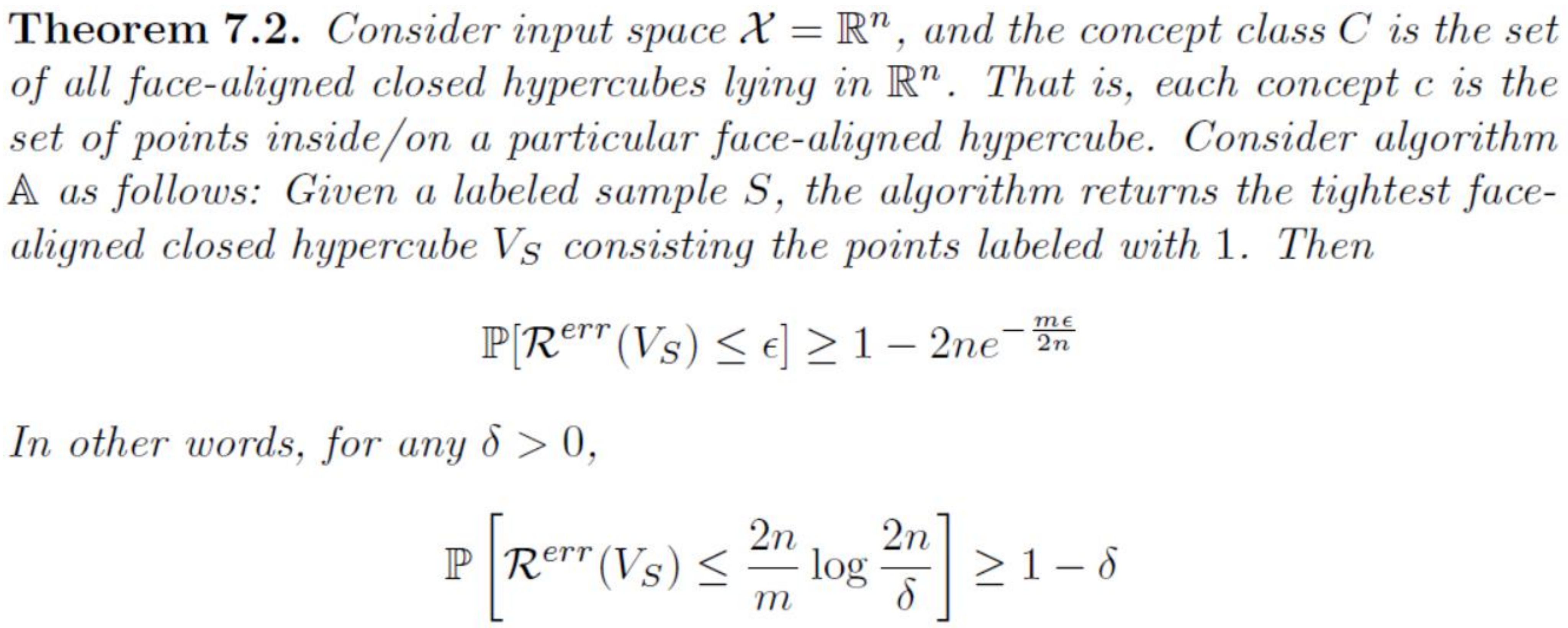
- Hence $\mathbb{P}_{S\sim D^m}[\mathcal{R}(h_S) \leq \epsilon] \geq 1 - 4e^{-\frac{m\epsilon}{4}}$

➔ $\mathbb{P}_{S\sim D^m}[\mathcal{R}(h_S) \leq \epsilon] \geq 1 - \delta$ for $m \geq \boxed{\frac{4}{\epsilon}\log\frac{4}{\delta}}$

Sample complexity



Axis-aligned hyper-cube is PAC-learnable (Formal Proof)



Theorem 7.2. *Consider input space $\mathcal{X} = \mathbb{R}^n$, and the concept class C is the set of all face-aligned closed hypercubes lying in $\mathbb{R}^n$. That is, each concept c is the set of points inside/on a particular face-aligned hypercube. Consider algorithm $\mathbb{A}$ as follows: Given a labeled sample S, the algorithm returns the tightest face-aligned closed hypercube V_S consisting the points labeled with 1. Then*

$$\mathbb{P}[\mathcal{R}^{err}(V_S) \leq \epsilon] \geq 1 - 2ne^{-\frac{m\epsilon}{2n}}$$

In other words, for any $\delta > 0$,

$$\mathbb{P}\left[\mathcal{R}^{err}(V_S) \leq \frac{2n}{m}\log\frac{2n}{\delta}\right] \geq 1 - \delta$$

That is, $\boxed{\mathbb{P}_{S\sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta \text{ for } m \geq \frac{2n}{\epsilon}\log\frac{2n}{\delta}}$

PAC-learnable

Proof. Let $V \in C$ be a target concept, which is a face-aligned closed hypercube defined by $V = \{\mathrm{x} \in \mathbb{R}^n : x^{(k)} \in [a_k, b_k], \forall k = 1, \cdots, n\}$. By definition, $V_S \subset V$. Since $\mathcal{R}^{err}(V_S) \leq \mathbb{P}[\mathrm{x} \in V]$, we may assume $\mathbb{P}[\mathrm{x} \in V] > \epsilon$. Define hypercubes

$$v_{k,1} = \{\mathrm{x} \in V : x^{(k)} \in [a_k, s_k]\}, \quad \bar{v}_{k,1} = \{\mathrm{x} \in V : x^{(k)} \in [a_k, s_k)\}$$
$$v_{k,2} = \{\mathrm{x} \in V : x^{(k)} \in [t_k, b_k]\}, \quad \bar{v}_{k,2} = \{\mathrm{x} \in V : x^{(k)} \in (t_k, b_k]\}$$

where

$$s_k = \inf\{s : \mathbb{P}[\{\mathrm{x} \in V : x^{(k)} \in [a_k, s]\}] \geq \frac{\epsilon}{2n}\}$$
$$t_k = \inf\{t : \mathbb{P}[\{\mathrm{x} \in V : x^{(k)} \in [t, b_k]\}] \geq \frac{\epsilon}{2n}\}$$

Then $\mathbb{P}[\mathrm{x} \in v_{k,\ell}] \geq \frac{\epsilon}{2n}$, $\mathbb{P}[\mathrm{x} \in \bar{v}_{k,\ell}] \leq \frac{\epsilon}{2n}, \forall k = 1, \cdots, n, \ell = 1, 2$. Define $V_0 = \{\mathrm{x} \in \mathbb{R}^n : x^{(k)} \in [s_k, t_k], \forall k = 1, \cdots, n\}$. Then $V_0 \subset V_S \subset V$ implies

$$\mathcal{R}^{err}(V_S) \leq \mathbb{P}\left[\mathrm{x} \in \bigcup_{k=1}^{n} \bigcup_{\ell=1}^{2} \bar{v}_{k,\ell}\right] \leq \sum_{k=1}^{n} \sum_{\ell=1}^{2} \mathbb{P}[\mathrm{x} \in \bar{v}_{k,\ell}] \leq \epsilon$$

Note that

$$\mathbb{P}[V_0 \not\subset V_S] = \mathbb{P}\left[\bigcup_{k=1}^{n} \bigcup_{\ell=1}^{2} (S \cap v_{k,\ell} = \emptyset)\right] \leq \sum_{k=1}^{n} \sum_{\ell=1}^{2} \mathbb{P}[S \cap v_{k,\ell} = \emptyset] \leq 2n \left(1 - \frac{\epsilon}{2n}\right)^m$$

Therefore

$$\mathbb{P}[\mathcal{R}^{err}(V_S) \leq \epsilon] \geq \mathbb{P}[V_0 \subset V_S \subset V] \geq 1 - 2n \left(1 - \frac{\epsilon}{2n}\right)^m \geq 1 - 2ne^{-\frac{m\epsilon}{2n}}$$

Sample complexity for finite hypothesis sets - consistent case

- **Theorem:** Let H be a finite set of binary classifiers on $\mathcal{X}$. Let $\mathbb{A}$ be an algorithm such that for any target concept $c \in H$ and i.i.d. sample S of size m returns a consistent hypothesis $\mathbb{A}(S) \in H$ such that $\hat{\mathcal{R}}_S\big(\mathbb{A}(S)\big) = 0$. Then

$$\mathbb{P}_{S\sim D^m}[\mathcal{R}(\mathbb{A}(S)) \leq \epsilon] \geq 1 - |H|e^{-m\epsilon}$$

where D is the underlying distribution. In other words,

$$\mathbb{P}_{S\sim D^m}\left[\mathcal{R}(\mathbb{A}(S)) \leq \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right)\right] \geq 1 - \delta$$

(Mohri 2012, Theorem 2.1)

Note that the bound holds true regardless of the algorithm $\mathbb{A}$, the target concept c, or the underlying distribution D.

Sample complexity

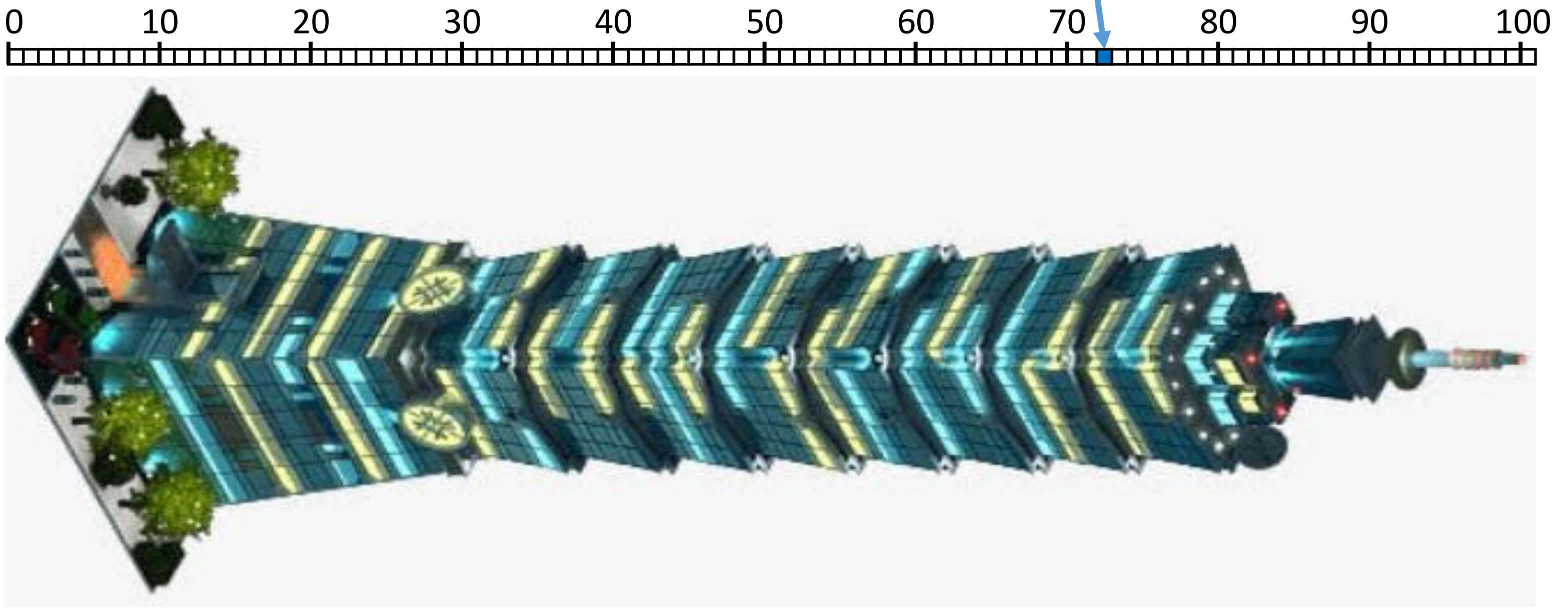
$$\mathbb{P}_{S\sim D^m}\big[\mathcal{R}\big(\mathbb{A}(S)\big) \leq \epsilon\big] \geq 1 - \delta \text{ for } m \geq \boxed{\frac{\log|H| + \log(1/\delta)}{\epsilon}}$$

Example

- 費小清 wishes to predict whether or not i-phone 10 will break if thrown out from the x'th floor at Taipei 101.
  - ➢ $\mathcal{X} = \{1,2,\dots,101\}$ (There are 101 floors)
  - ➢ Hypothesis h_k: The maximum floor thrown out from which i-phone 10 will remain intact is floor k, namely
$$h_k(x) = \begin{cases} \text{intact} & ,\text{if } x \leq k \\ \text{broken} & ,\text{if } x > k \end{cases}$$
  - ➢ Hypothesis set $H = \{h_0, h_1, h_2, \dots, h_{101}\}$.
  - ➢ Target concept c $= h_{k^*} \in H$, where $0 \leq k^* \leq 101$ is unknown to 費小清.
- Suppose 費小清 is interested in the accuracy of the model, should the floors be drawn according to distribution D. The (true) risk function is
$$\mathcal{R}(h) = \mathbb{E}_{X\sim D}\left[1_{h(X)\neq c(X)}\right]$$
Say, if D is the uniform distribution, then
$$\mathcal{R}(h) = \frac{1}{101}\sum_{x=1}^{101} 1_{h(x)\neq c(x)}$$

Example

- 費小清 collects data and train a prediction model
  - The i'th experiment: Randomly choose $X_i \sim D$, throw i-phone 10 from the X_i'th floor, and record the result Y_i (broken/intact).
  - Empirical risk function for sample $S = \big((X_1, Y_1), \dots, (X_m, Y_m)\big)$:
    $$\hat{\mathcal{R}}_S(h) = \frac{1}{m}\sum_{i=1}^{m} 1_{h(X_i) \neq Y_i} = \frac{1}{m}\sum_{i=1}^{m} 1_{h(X_i) \neq c(X_i)}$$
  - Based on the collected sample S, 費小清 applies an algorithm $\mathbb{A}$ to train a model $\mathbb{A}(S) \in H$ that achieves zero empirical risk $\hat{\mathcal{R}}_S\big(\mathbb{A}(S)\big) = 0$, namely $\mathbb{A}(S)(X_i) = c(X_i)$ for all $i = 1, \dots, m$.
- One can guarantee that
  $$\mathbb{P}_{S \sim D^m}\left[\mathcal{R}(\mathbb{A}(S)) \le \frac{1}{m}\left(\log|H| + \log\frac{1}{\delta}\right)\right] \ge 1 - \delta$$
  - Here $|H| = |\{h_0, h_1, h_2, \dots, h_{101}\}| = 102$, so
    $$\mathbb{P}_{S \sim D^m}\left[\mathcal{R}\big(\mathbb{A}(S)\big) \le \frac{1}{m}\left(\log(102) + \log\frac{1}{\delta}\right)\right] \ge 1 - \delta$$



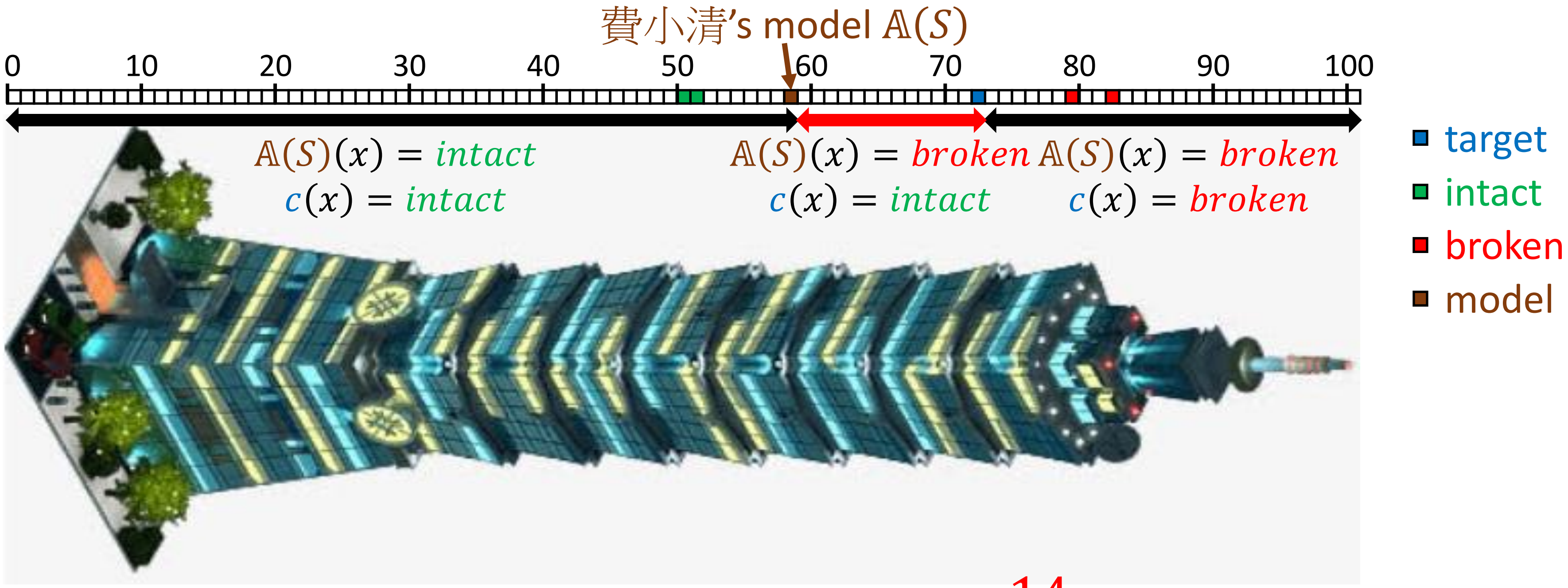
Target concept c (unknown to 費小清)

■ target

$m = 5$

$S = ((X_1, Y_1), \dots, (X_5, Y_5))$

Assume D is uniform distribution



費小清's model $\mathbb{A}(S)$

$\mathbb{A}(S)(x) = \mathit{intact}$
$c(x) = \mathit{intact}$

$\mathbb{A}(S)(x) = \mathit{broken}$
$c(x) = \mathit{intact}$

$\mathbb{A}(S)(x) = \mathit{broken}$
$c(x) = \mathit{broken}$

- 🟦 target
- 🟩 intact
- 🟥 broken
- 🟫 model

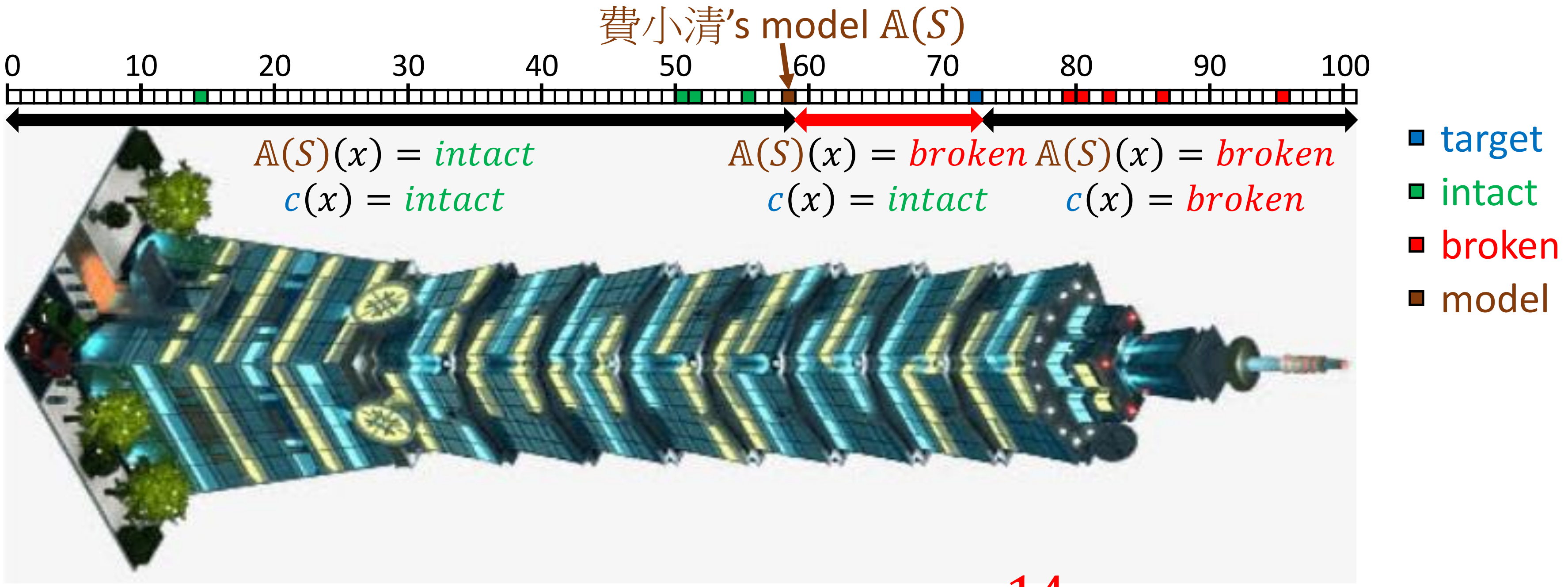
$$\mathcal{R}(\mathbb{A}(S)) = \mathbb{E}_{X\sim D}[1_{h(X)\neq c(X)}] = \frac{14}{101} = 0.1386$$

$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}(\mathbb{A}(S)) \le \frac{1}{5}\underbrace{\left(\log(102) + \log\frac{1}{0.1}\right)}_{1.3855}\right] \ge 0.9$$

$m = 10$

$S = \big((X_1, Y_1), \ldots, (X_{10}, Y_{10})\big)$

Assume D is uniform distribution



費小清's model $\mathbb{A}(S)$
0
10
20
30
40
50
60
70
80
90
100
$\mathbb{A}(S)(x) = intact$
$c(x) = intact$
$\mathbb{A}(S)(x) = broken$
$c(x) = intact$
$\mathbb{A}(S)(x) = broken$
$c(x) = broken$

🟦 target

🟩 intact

🟥 broken

🟫 model

$$\mathcal{R}\big(\mathbb{A}(S)\big) = \mathbb{E}_{X\sim D}\left[1_{h(X)\neq c(X)}\right] = \frac{14}{101} = 0.1386$$

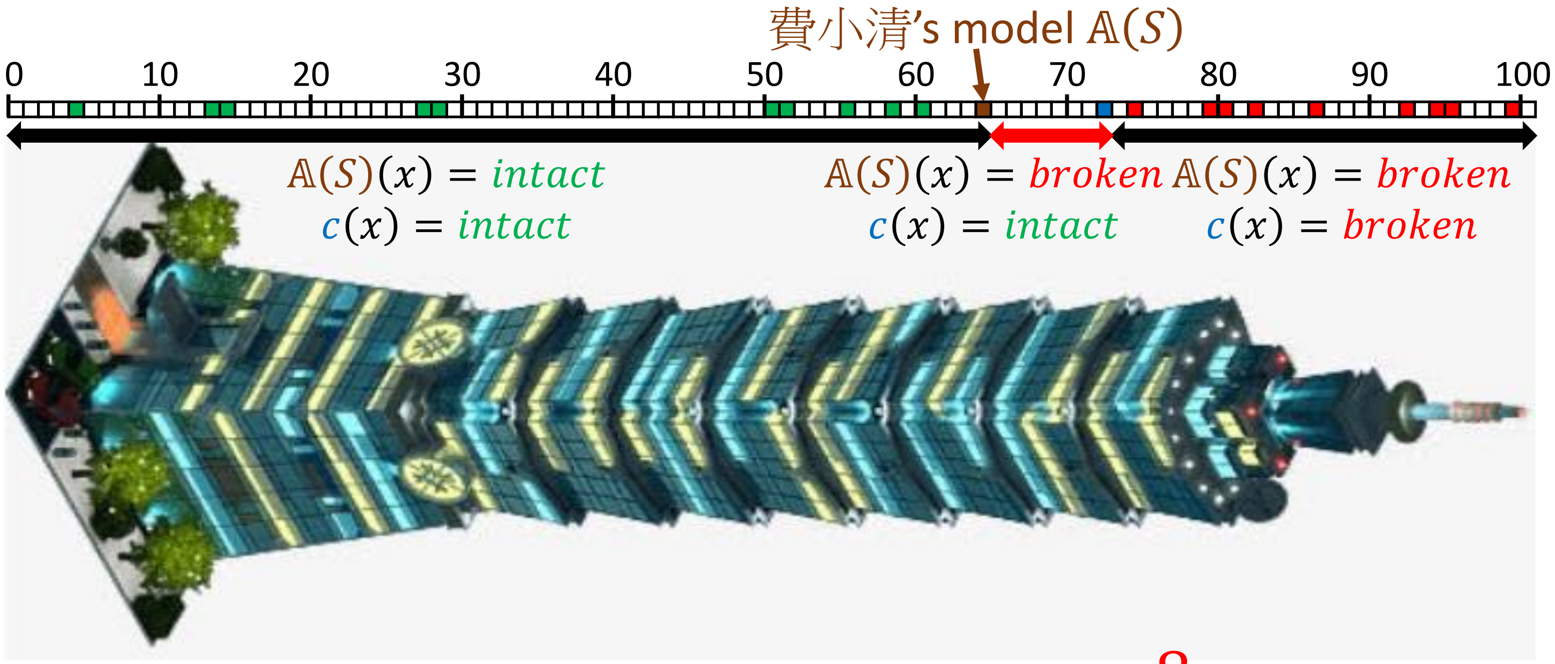
$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}\big(\mathbb{A}(S)\big) \leq \frac{1}{10}\left(\log(102) + \log\frac{1}{0.1}\right)\right] \geq 0.9$$

0.6928

$m = 20$

$S = \left((X_1, Y_1), \dots, (X_{20}, Y_{20})\right)$

Assume D is uniform distribution


費小清's model $\mathbb{A}(S)$

$\mathbb{A}(S)(x) = intact$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = broken$

🟦 target
🟩 intact
🟥 broken
🟫 model

$$\mathcal{R}(\mathbb{A}(S)) = \mathbb{E}_{X\sim D}\left[1_{h(X)\neq c(X)}\right] = \frac{8}{101} = .0792$$

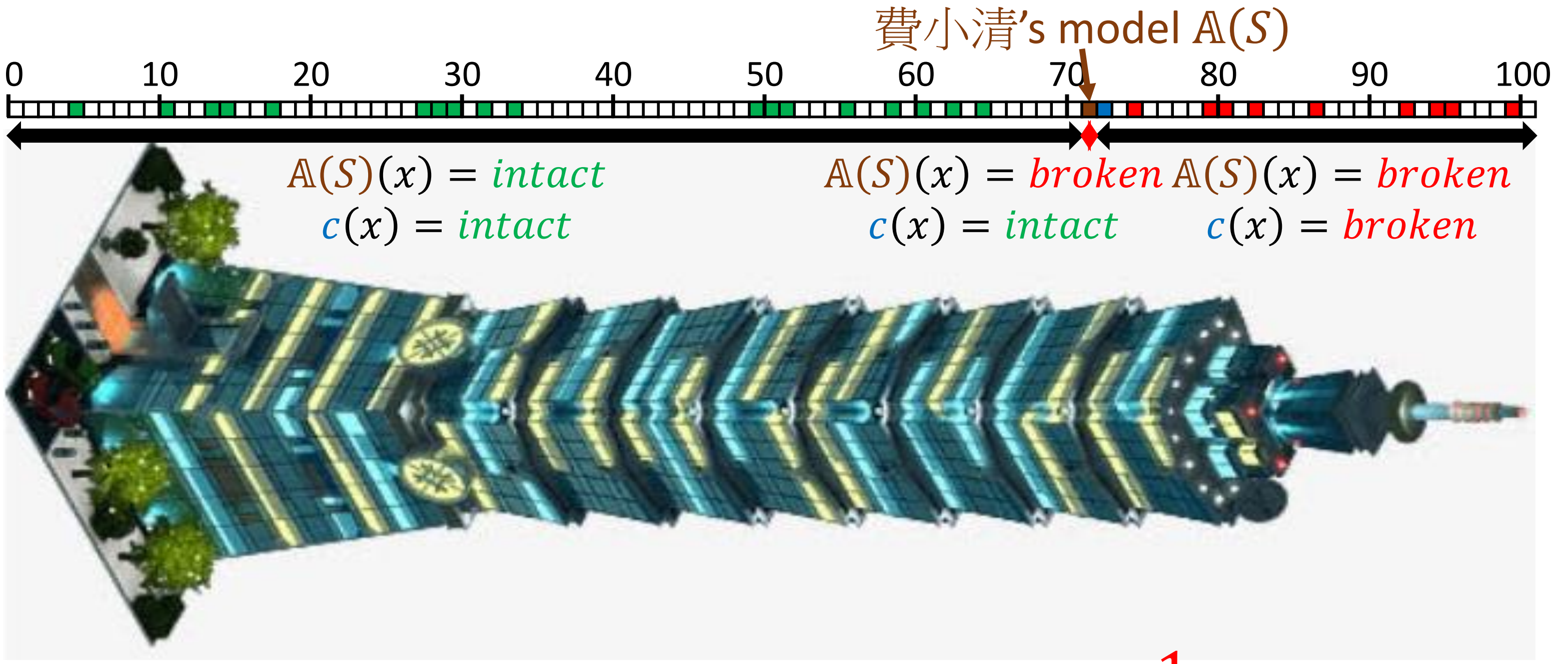
$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}(\mathbb{A}(S)) \leq \frac{1}{20}\left(\log(102) + \log\frac{1}{0.1}\right)\right] \geq 0.9$$

0.3464

$m = 30$

$S = ((X_1, Y_1), \dots, (X_{30}, Y_{30}))$

Assume D is uniform distribution



費小清's model $\mathbb{A}(S)$

$\mathbb{A}(S)(x) = intact$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = broken$

- 🟦 target
- 🟩 intact
- 🟥 broken
- 🟫 model

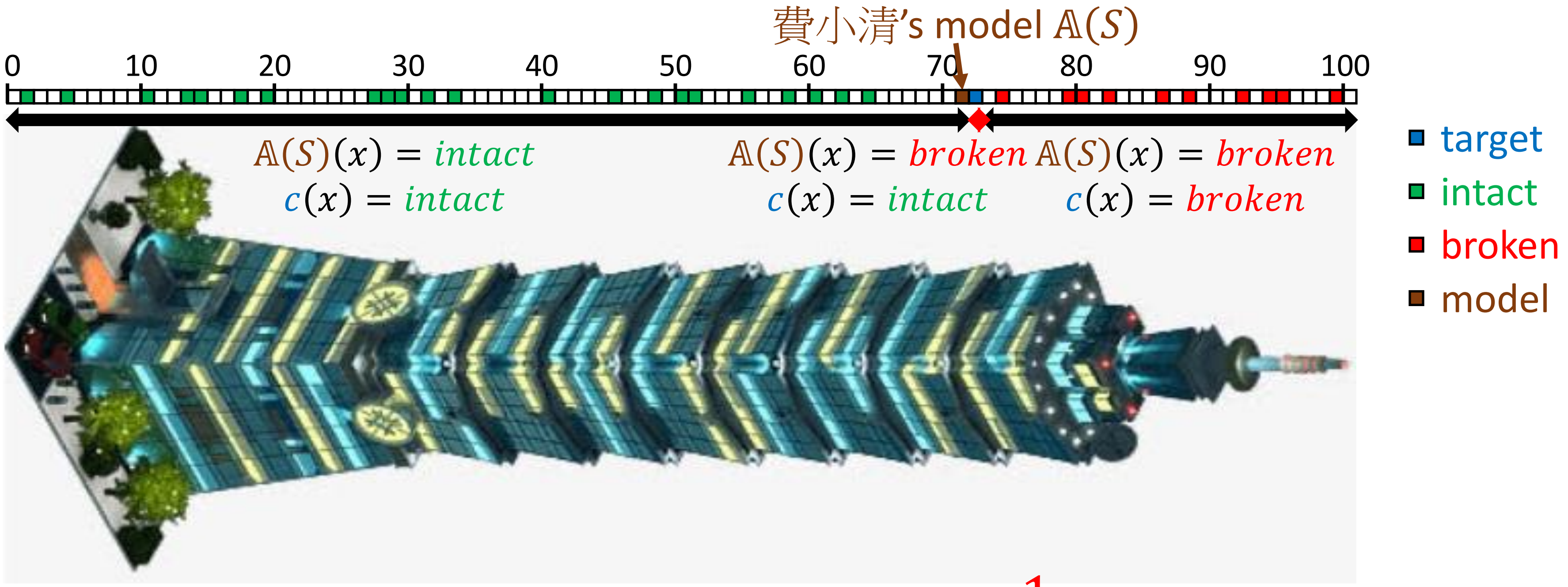
$$\mathcal{R}(\mathbb{A}(S)) = \mathbb{E}_{X\sim D}\left[1_{h(X)\neq c(X)}\right] = \frac{1}{101} = .0099$$

$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}(\mathbb{A}(S)) \le \frac{1}{30}\underset{0.2309}{\left(\log(102) + \log\frac{1}{0.1}\right)}\right] \ge 0.9$$

$m = 40$

$S = ((X_1, Y_1), \ldots, (X_{40}, Y_{40}))$

Assume D is uniform distribution



費小清's model $\mathbb{A}(S)$

$\mathbb{A}(S)(x) = intact$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = intact$

$\mathbb{A}(S)(x) = broken$
$c(x) = broken$

🟦 target
🟩 intact
🟥 broken
🟫 model

$$\mathcal{R}(\mathbb{A}(S)) = \mathbb{E}_{X\sim D}\left[1_{h(X)\neq c(X)}\right] = \frac{1}{101} = .0099$$

$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}(\mathbb{A}(S)) \leq \frac{1}{40}\left(\log(102) + \log\frac{1}{0.1}\right)\right] \geq 0.9$$

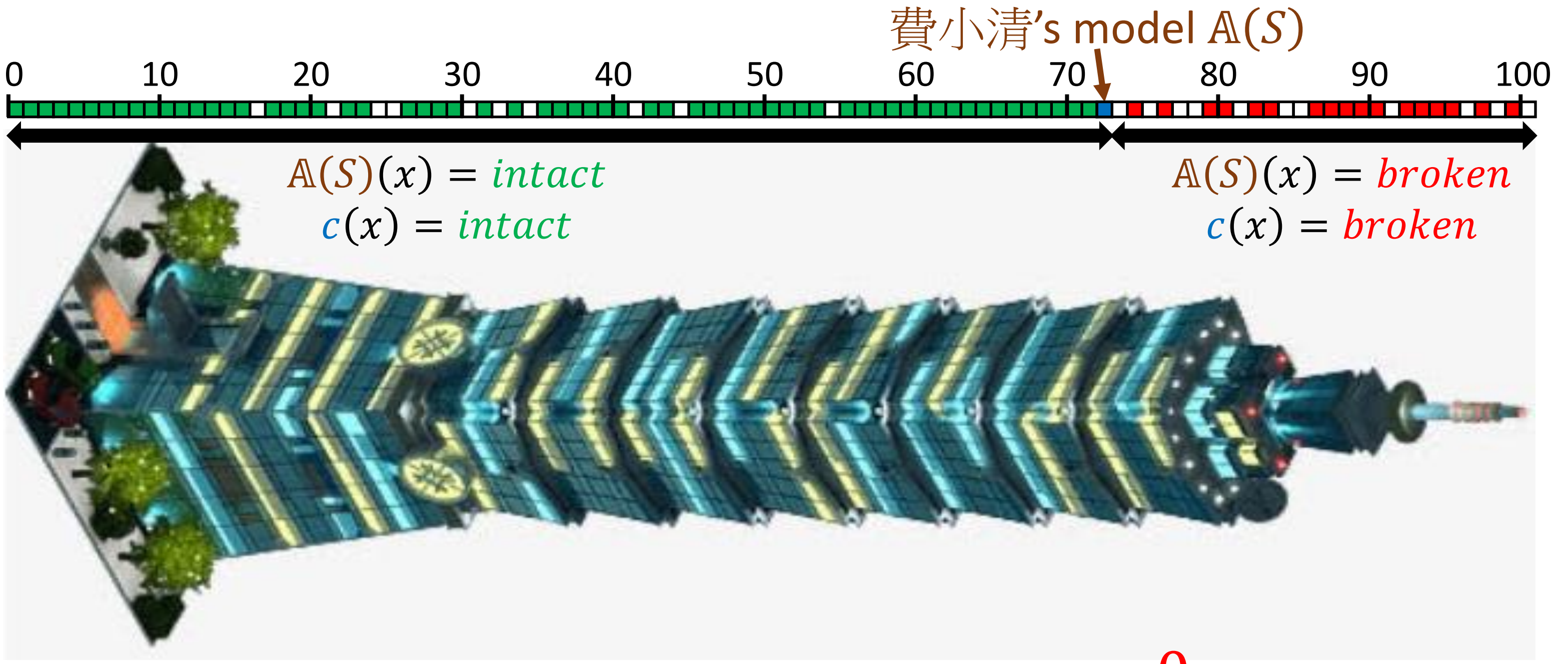
0.1732

4 90 73

$m = 167$

$S = \big((X_1, Y_1), \dots, (X_{167}, Y_{167})\big)$

Assume D is uniform distribution


費小清's model $\mathbb{A}(S)$
0
10
20
30
40
50
60
70
80
90
100
$\mathbb{A}(S)(x) = intact$
$c(x) = intact$
$\mathbb{A}(S)(x) = broken$
$c(x) = broken$

- 🟦 target
- 🟩 intact
- 🟥 broken
- 🟫 model

$$\mathcal{R}\big(\mathbb{A}(S)\big) = \mathbb{E}_{X\sim D}\big[1_{h(X)\neq c(X)}\big] = \frac{0}{101} = 0$$

$$\mathbb{P}_{S\sim D^5}\left[\mathcal{R}\big(\mathbb{A}(S)\big) \le \frac{1}{167}\left(\log(102) + \log\frac{1}{0.1}\right)\right] \ge 0.9$$

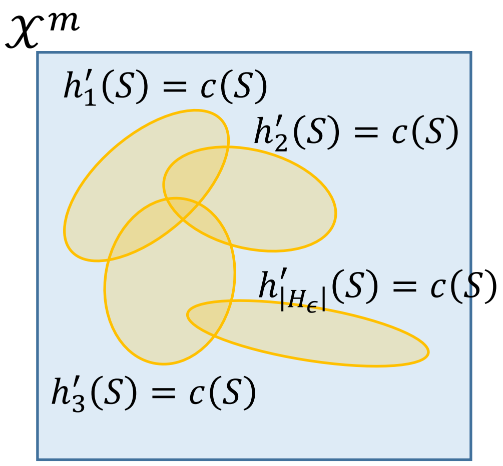
0.0415

Sample complexity for finite hypothesis sets - consistent case (Proof)

- **Theorem:** Let H be a finite set of binary classifiers on $\mathcal{X}$. Let $\mathbb{A}$ be an algorithm such that for any target concept $c \in H$ and i.i.d. sample S of size m returns a consistent hypothesis $\mathbb{A}(S) \in H$ such that $\hat{\mathcal{R}}_S\big(\mathbb{A}(S)\big) = 0$. Then
$$\mathbb{P}_{S\sim D^m}[\mathcal{R}(\mathbb{A}(S)) \leq \epsilon] \geq 1 - |H|e^{-m\epsilon}$$

Proof: Let $H_\epsilon = \{h \in H : \mathcal{R}(h) > \epsilon\}$, then

$$\begin{aligned}
\mathbb{P}_{S\sim D^m}\big[\mathcal{R}\big(\mathbb{A}(S)\big) > \epsilon\big] &= \mathbb{P}_{S\sim D^m}[\mathbb{A}(S) \in H_\epsilon] \\
&\leq \mathbb{P}_{S\sim D^m}[\exists h \in H_\epsilon \ \ s.t.\ h(S) = c(S)] \\
&\leq \sum\nolimits_{h \in H_\epsilon} \mathbb{P}_{S\sim D^m}[h(S) = c(S)] \\
&< \sum\nolimits_{h \in H_\epsilon} (1-\epsilon)^m \leq |H_\epsilon| e^{-m\epsilon}
\end{aligned}$$



$H_\epsilon = \{h'_1, \ldots, h'_{|H_\epsilon|}\}$

Empirical Risk Minimization

- Let H be a family of hypotheses. Let $h^* \in H$ be the optimal hypothesis with the minimum (true) risk among H:

$$h^* \in \operatorname*{argmin}_{h \in H} \mathcal{R}(h)$$

- **Empirical Risk Minimization (ERM)**

  Since one cannot evaluate the risk function $\mathcal{R}(\cdot)$ directly, one may instead approximate $\mathcal{R}$ by the empirical risk $\hat{\mathcal{R}}_S$ evaluated over sample S, and approximate h^* by the hypothesis h_S^{ERM} that minimizes the empirical risk

$$h_S^{ERM} \in \operatorname*{argmin}_{h \in H} \hat{\mathcal{R}}_S(h)$$

h_S^{ERM} may be suboptimal, but what is the gap?

$$\begin{aligned}
\mathcal{R}\left(h_S^{ERM}\right) - \mathcal{R}(h^*) &= \left(\mathcal{R}\left(h_S^{ERM}\right) - \hat{\mathcal{R}}_S\left(h_S^{ERM}\right)\right) + \left(\hat{\mathcal{R}}_S\left(h_S^{ERM}\right) - \mathcal{R}(h^*)\right) \\
&\le \left(\mathcal{R}\left(h_S^{ERM}\right) - \hat{\mathcal{R}}_S\left(h_S^{ERM}\right)\right) + \left(\hat{\mathcal{R}}_S(h^*) - \mathcal{R}(h^*)\right) \\
&\le 2 \boxed{\sup_{h \in H} \left|\hat{\mathcal{R}}_S(h) - \mathcal{R}(h)\right|}
\end{aligned}$$

Can we bound this quantity?

Sample complexity for finite hypothesis sets - inconsistent case

- **Theorem:** Let H be a finite set of binary classifiers on $\mathcal{X}$, then

$$\mathbb{P}_{S\sim D^m}\left[\max_{h\in H}\left|\hat{\mathcal{R}}_S(h)-\mathcal{R}(h)\right|<\epsilon\right]\geq 1-2|H|e^{-2m\epsilon^2}$$

where D is the underlying distribution. In other words,

$$\mathbb{P}_{S\sim D^m}\left[\max_{h\in H}\left|\hat{\mathcal{R}}_S(h)-\mathcal{R}(h)\right|<\sqrt{\frac{\log|H|+\log(2/\delta)}{2m}}\right]\geq 1-\delta$$

(Mohri 2012, Theorem 2.2)

The bound of gap between generalization error and training error over all hypotheses

Note that the bound holds true regardless of the underlying distribution D.

Sample complexity

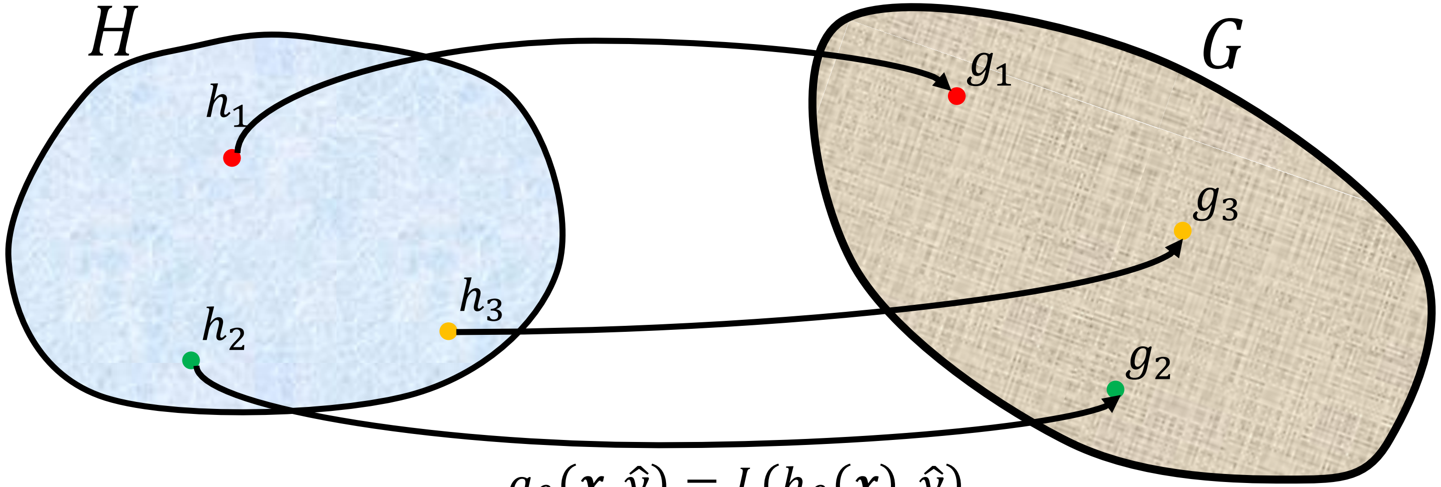
$$\mathbb{P}_{S\sim D^m}\left[\max_{h\in H}\left|\hat{\mathcal{R}}_S(h)-\mathcal{R}(h)\right|<\epsilon\right]\geq 1-\delta \text{ for } \boxed{m\geq\frac{\log|H|+\log(2/\delta)}{2\epsilon^2}}$$

Rademacher Complexity

A useful tool to derive non-trivial generalization bounds when $|H| = \infty$

Loss functions associated to hypothesis set

- Let H be the hypothesis set of functions mapping from input space $\mathcal{X}$ to output space $\mathcal{Y}$.
- Let $L(y, \hat{y})$ be the loss function between prediction $y \in \mathcal{Y}$ and ground truth $\hat{y} \in \mathcal{Y}$.
- To each hypothesis $h \in H$, we can associate a function g that maps $(x, \hat{y}) \in \mathcal{X} \times \mathcal{Y}$ to $L(h(x), \hat{y})$. In other words, *$g(\boldsymbol{x}, \hat{y})$ evaluates the loss h suffers given input $\boldsymbol{x}$ and ground truth $\hat{y}$.*
- Denote G as the collection of all such functions g associated to some $h \in H$.



$$g_\theta(\boldsymbol{x}, \hat{y}) = L(h_\theta(\boldsymbol{x}), \hat{y})$$

Loss functions associated to hypothesis set

- **Example:** The hypothesis set of all linear binary classifiers on $\mathbb{R}^d$ can be written as

$$H = \left\{h_{\boldsymbol{w},b} : \boldsymbol{w} \in \mathbb{R}^d, b \in \mathbb{R}\right\},$$

where each $h_{\boldsymbol{w},b}$ is a binary linear classifier

$$h_{\boldsymbol{w},b}(\boldsymbol{x}) = sign(\boldsymbol{w}^T\boldsymbol{x} + b\ )$$

Suppose we adopt the 0-1 loss function

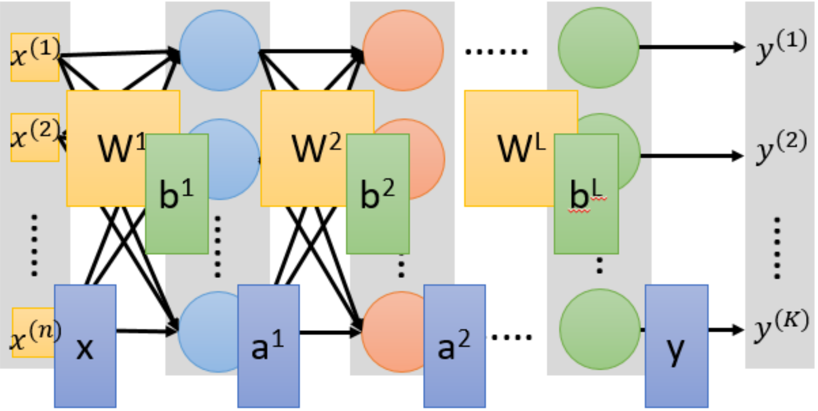
$$L(y, \hat{y}) = 1\{y \neq \hat{y}\}$$

We can associate each hypothesis $h_{\boldsymbol{w},b} \in H$ with $g_{\boldsymbol{w},b}$, as given by

$$g_{\boldsymbol{w},b}(\boldsymbol{x}, \hat{y}) = L\left(h_{\boldsymbol{w},b}(\boldsymbol{x}), \hat{y}\right) = 1\left\{h_{\boldsymbol{w},b}(\boldsymbol{x}) \neq \hat{y}\right\}$$

Loss functions associated to hypothesis set

- **Example:** The hypothesis set pertaining to a neural network



$$H = \{h_\theta : \theta \in \Theta\},$$

where $\theta = \{\boldsymbol{W}^l, \boldsymbol{b}^l\}_{l=1}^L$ is the parameter of all weights and biases, and

$$h_\theta(x) = \sigma(\ W^L\ \cdots\ \sigma(\ W^2\ \sigma(\ W^1\ x\ +\ b^1\ ) + \ b^2\ )\ \cdots + \ b^L\ )$$

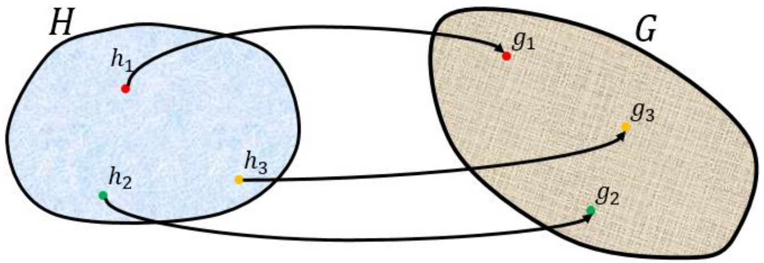
Suppose we consider the cross entropy loss

$$L(y, \hat{y}) = -\sum_{k=1}^{K} \hat{y}^{(k)} \log y^{(k)}$$

We can associate each hypothesis $h_\theta \in H$ with g_θ, as given by

$$g_\theta(\boldsymbol{x}, \hat{y}) = L(h_\theta(\boldsymbol{x}), \hat{y}) = -\sum_{k=1}^{K} \hat{y}^{(k)} \log h_\theta^{(k)}(\boldsymbol{x})$$

Set of Loss Functions and Empirical Loss Minimization



$g_\theta(\boldsymbol{x}, \hat{y}) = L(h_\theta(\boldsymbol{x}), \hat{y})$

- G can be interpreted as the family of loss functions associated to H.
- To minimize the empirical loss evaluated over training data $\{(x_i, \hat{y}_i)\}_{i=1}^m$ is equivalent to

Denote $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ and $z_i = (x_i, \hat{y}_i)$

$$\inf_{h \in H} \sum_{i=1}^{m} L(h(x_i), \hat{y}_i) = \inf_{g \in G} \sum_{i=1}^{m} g(x_i, \hat{y}_i) = \inf_{g \in G} \sum_{i=1}^{m} g(z_i\ )$$

- If G is “big”, it is more likely to achieve small empirical loss, but also more likely to overfit.

- How to measure the “size” of G? *Rademacher complexity*
- How does the “size” of G relates to “overfitting”? *Generalization bound*

Rademacher Complexity

- Let G be a family of functions mapping from $\mathcal{Z}$ to $[a, b]$ and $S = (z_1, \dots, z_m)$ a fixed sample of size m with elements in $\mathcal{Z}$. Then the **empirical Rademacher complexity** of G with respect to the sample S is defined as

$$\widehat{\mathfrak{R}}_S(G) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g \in G} \frac{1}{m} \sum_{i=1}^{m} \sigma_i g(z_i)\right]$$

  where $\boldsymbol{\sigma} = (\sigma_1, \dots, \sigma_m)$, with σ_is being independent uniform random variables taking values in $\{-1, +1\}$. The random variables σ_i are called **Rademacher variables**.

- Let D denote the distribution according to which samples are drawn. For any $m \in \mathbb{N}$, the **Rademacher complexity** of G is the expectation of the empirical Rademacher complexity over all samples of size m drawn according to D:

$$\mathfrak{R}_m(G) = \mathbb{E}_{S \sim D^m}\left[\widehat{\mathfrak{R}}_S(G)\right]$$

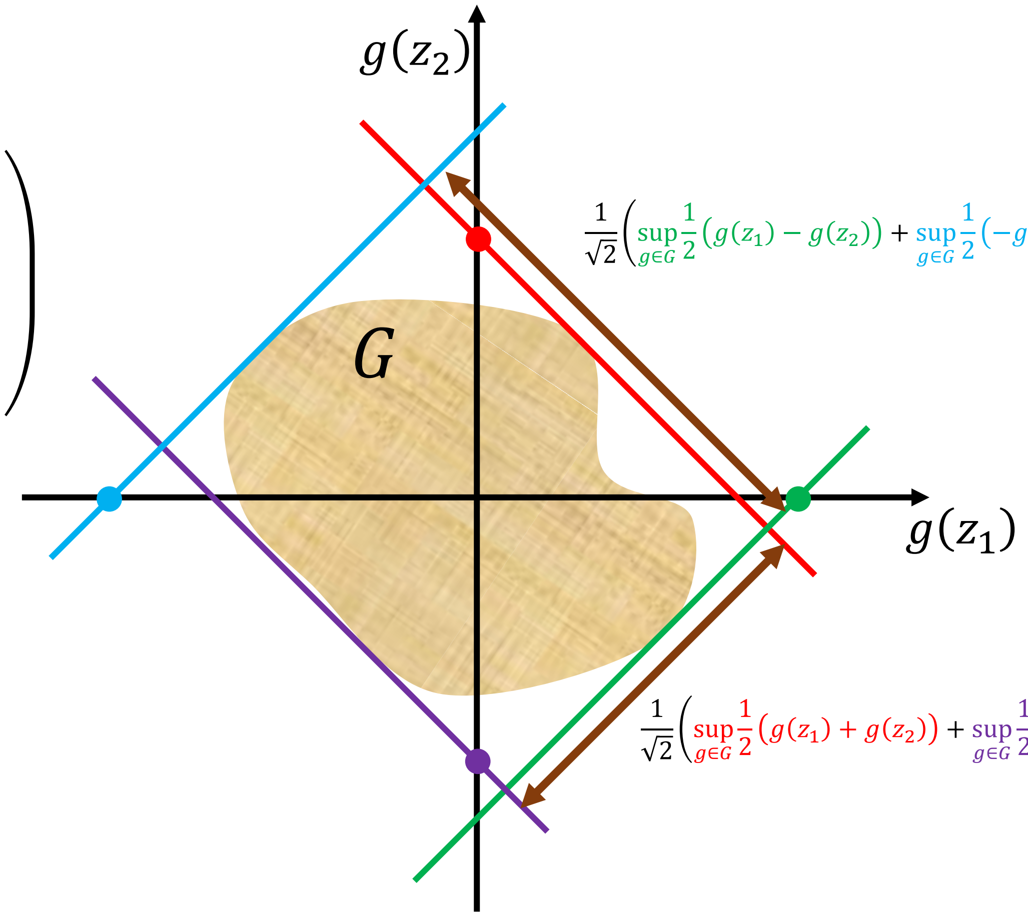
Geometric Interpretation

$$\hat{\mathfrak{R}}_S(G) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in G}\frac{1}{m}\sum_{i=1}^{m}\sigma_i g(z_i)\right]$$

Suppose we have two samples $S = \{z_1, z_2\}$, then

$$\hat{\mathfrak{R}}_S(G) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{g\in G}\frac{1}{2}\left(\sigma_1 g(z_1) + \sigma_2 g(z_2)\right)\right]$$

$$= \frac{1}{4}\begin{pmatrix} \sup_{g\in G}\frac{1}{2}\left(g(z_1)+g(z_2)\right) + \sup_{g\in G}\frac{1}{2}\left(-g(z_1)-g(z_2)\right) \\ +\sup_{g\in G}\frac{1}{2}\left(g(z_1)-g(z_2)\right) + \sup_{g\in G}\frac{1}{2}\left(-g(z_1)+g(z_2)\right) \end{pmatrix}$$



$$\frac{1}{\sqrt{2}}\left(\sup_{g\in G}\frac{1}{2}\left(g(z_1)-g(z_2)\right) + \sup_{g\in G}\frac{1}{2}\left(-g(z_1)+g(z_2)\right)\right)$$

$$\frac{1}{\sqrt{2}}\left(\sup_{g\in G}\frac{1}{2}\left(g(z_1)+g(z_2)\right) + \sup_{g\in G}\frac{1}{2}\left(-g(z_1)-g(z_2)\right)\right)$$

Binary Classifier Generalization Bound

- Let $\mathcal{X}$ be input space, $\mathcal{Y} = \{-1, +1\}$ be output space, H be a hypothesis set. If 0-1 loss is concerned, then

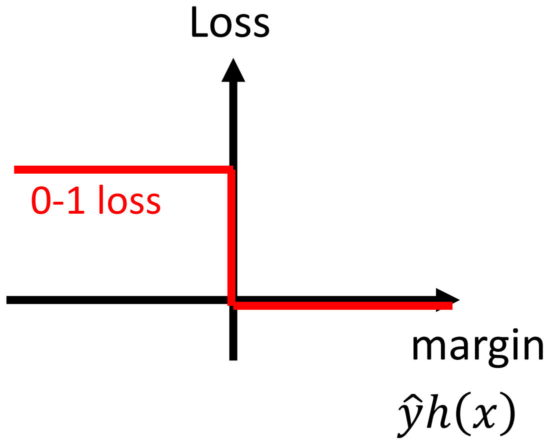
$$\mathbb{P}\left[\sup_{h\in H}\Big(\overset{\text{True loss}}{\mathcal{R}(h)} - \overset{\text{Training loss}}{\hat{\mathcal{R}}_S(h)}\Big) \le \overset{\text{Hypothesis complexity}}{\mathfrak{R}_m(H)} + \sqrt{\frac{\log(1/\delta)}{\underset{\text{Sample size}}{2m}}}\right] \ge 1 - \underset{\text{Confidence}}{\delta}$$

(Mohri 2012, Theorem 3.2)

where $\mathfrak{R}_m(H) = \mathbb{E}_{S\sim D^m}\big[\hat{\mathfrak{R}}_S(H)\big]$, for which D is the underlying distribution on $\mathcal{X}$, and

$$\hat{\mathfrak{R}}_S(H) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in H}\frac{1}{m}\sum_{i=1}^{m}\sigma_i h(x_i)\right]$$

for $S = (x_1, \ldots, x_m)$



- Roughly speaking, Rademacher complexity bounds the gap between training error and true error.

Multi-class Classifier Generalization Bound

- Let $\mathcal{X}$ be input space, $\mathcal{Y} = \{1, \dots, k\}$ be output space, H be a hypothesis set. If hinge loss is concerned, then

$$\mathbb{P}\left[\sup_{h\in H}\left(\overset{\text{True loss}}{\mathcal{R}(h)} - \overset{\text{Training loss}}{\hat{\mathcal{R}}_S(h)}\right) \le \frac{2k^2}{\rho}\overset{\text{Hypothesis complexity}}{\mathfrak{R}_m\big(\psi(H)\big)} + \sqrt{\frac{\log(1/\delta)}{\underset{\text{Sample size}}{2m}}}\right] \ge 1 - \underset{\text{Confidence}}{\delta}$$

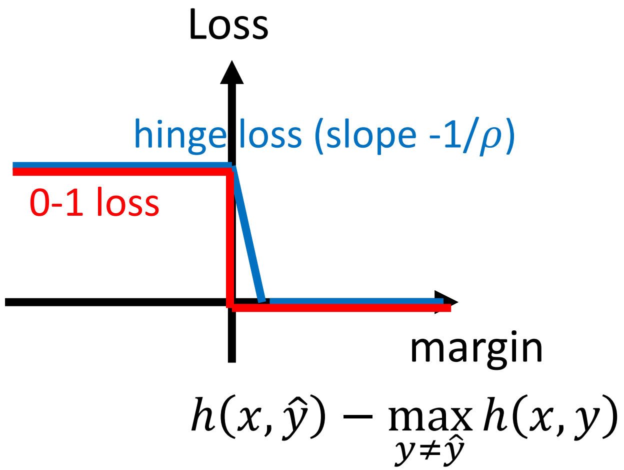
(Mohri 2012, Theorem 8.1)

  where $\psi(H) = \{x \mapsto h(x, y) : h \in H, y \in \mathcal{Y}\}$.

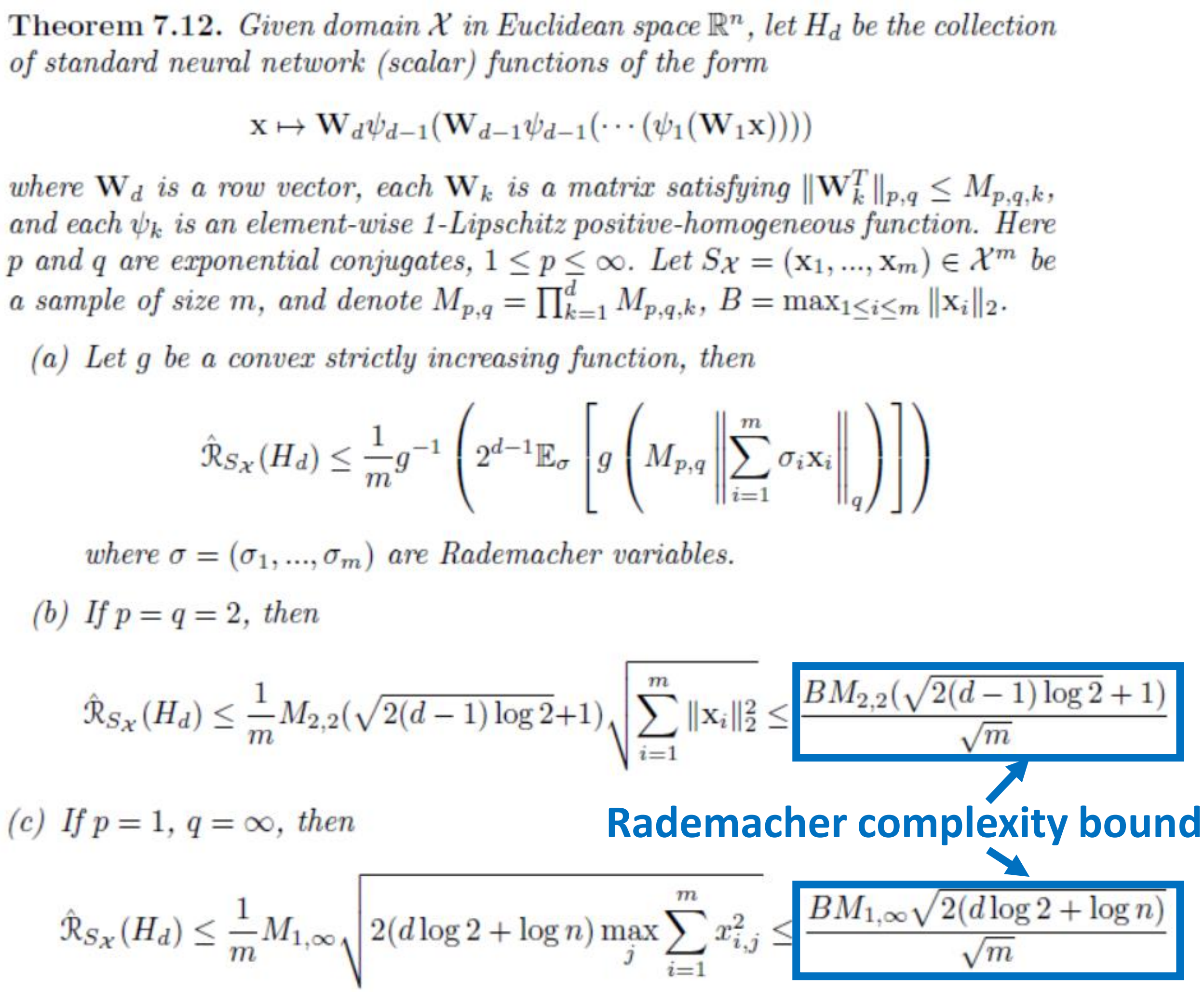
- More elaborately, $\mathfrak{R}_m\big(\psi(H)\big) = \mathbb{E}_{S\sim D^m}\left[\hat{\mathfrak{R}}_S\big(\psi(H)\big)\right]$, for which D is the underlying distribution on $\mathcal{X}$, and

$$\hat{\mathfrak{R}}_S\big(\psi(H)\big) = \mathbb{E}_{\boldsymbol{\sigma}}\left[\sup_{h\in H, y\in\mathcal{Y}} \frac{1}{m}\sum_{i=1}^{m} \sigma_i h(x_i, y)\right]$$

  for $S = (x_1, \dots, x_m)$



Rademacher complexity for Neural Network



Theorem 7.12. *Given domain $\mathcal{X}$ in Euclidean space $\mathbb{R}^n$, let H_d be the collection of standard neural network (scalar) functions of the form*

$$\mathrm{x} \mapsto \mathbf{W}_d\psi_{d-1}(\mathbf{W}_{d-1}\psi_{d-1}(\cdots(\psi_1(\mathbf{W}_1\mathrm{x}))))$$

where $\mathbf{W}_d$ is a row vector, each $\mathbf{W}_k$ is a matrix satisfying $\|\mathbf{W}_k^T\|_{p,q} \le M_{p,q,k}$, and each ψ_k is an element-wise 1-Lipschitz positive-homogeneous function. Here p and q are exponential conjugates, $1 \le p \le \infty$. Let $S_\mathcal{X} = (\mathrm{x}_1, ..., \mathrm{x}_m) \in \mathcal{X}^m$ be a sample of size m, and denote $M_{p,q} = \prod_{k=1}^d M_{p,q,k}$, $B = \max_{1\le i\le m}\|\mathrm{x}_i\|_2$.

(a) Let g be a convex strictly increasing function, then

$$\hat{\mathfrak{R}}_{S_\mathcal{X}}(H_d) \le \frac{1}{m} g^{-1}\left(2^{d-1}\mathbb{E}_\sigma\left[g\left(M_{p,q}\left\|\sum_{i=1}^m \sigma_i \mathrm{x}_i\right\|_q\right)\right]\right)$$

where $\sigma = (\sigma_1, ..., \sigma_m)$ are Rademacher variables.

(b) If $p = q = 2$, then

$$\hat{\mathfrak{R}}_{S_\mathcal{X}}(H_d) \le \frac{1}{m} M_{2,2}(\sqrt{2(d-1)\log 2}+1)\sqrt{\sum_{i=1}^m \|\mathrm{x}_i\|_2^2} \le \boxed{\frac{BM_{2,2}(\sqrt{2(d-1)\log 2}+1)}{\sqrt{m}}}$$

(c) If $p = 1$, $q = \infty$, then

$$\hat{\mathfrak{R}}_{S_\mathcal{X}}(H_d) \le \frac{1}{m} M_{1,\infty}\sqrt{2(d\log 2 + \log n)\max_j \sum_{i=1}^m x_{i,j}^2} \le \boxed{\frac{BM_{1,\infty}\sqrt{2(d\log 2+\log n)}}{\sqrt{m}}}$$

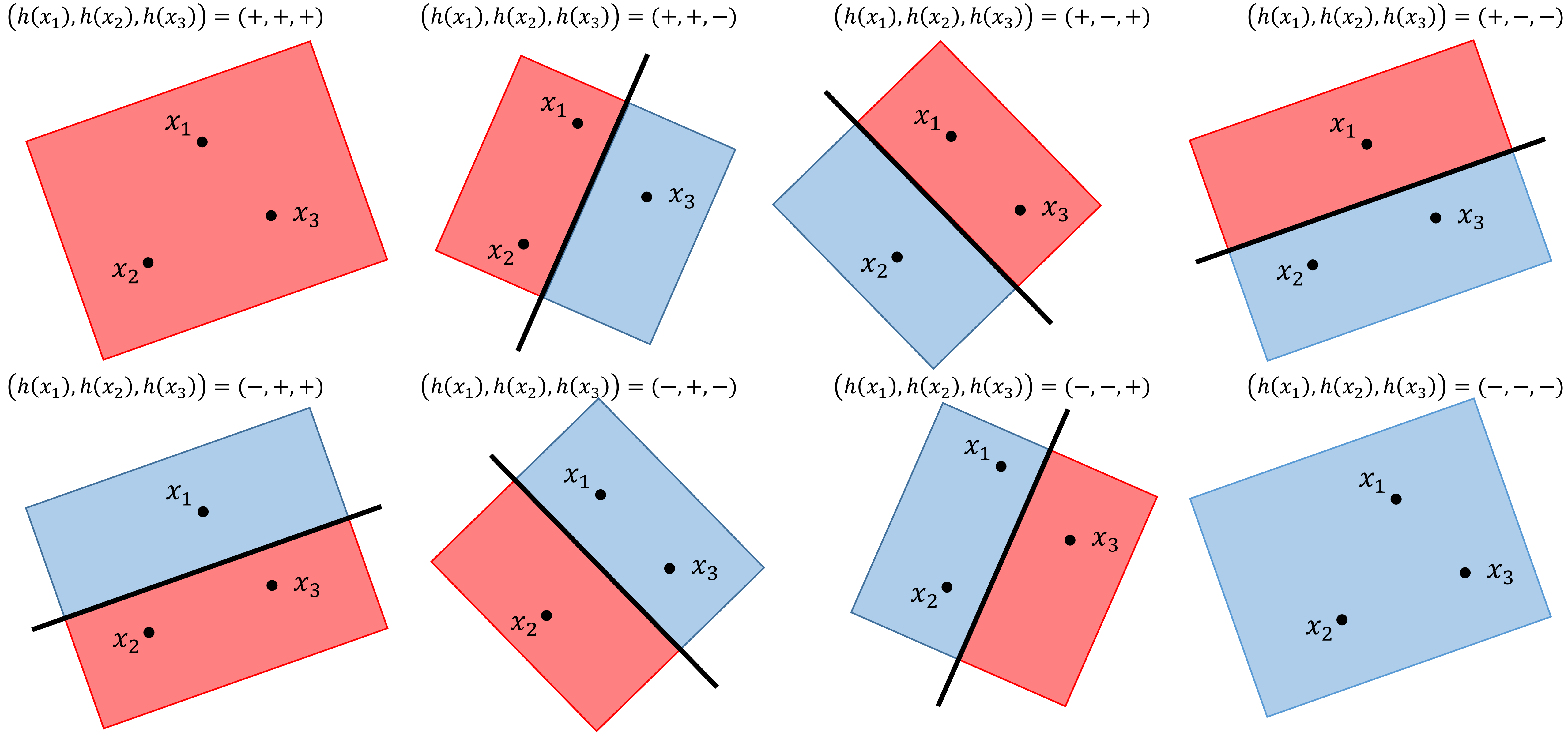
Rademacher complexity bounds

Noah Golowich, Alexander Rakhlin, and Ohad Shamir. "**Size-independent sample complexity of neural networks,"** *Proceedings of the 31st Conference On Learning Theory, PMLR* 75:297-299, 2018.

Growth Function and VC Dimension

Growth Function

- Let H be a family of binary functions mapping from $\mathcal{X}$ to $\{-1,+1\}$.
  - The **growth function** $\Pi_H\colon \mathbb{N} \to \mathbb{N}$ is defined by
  $$\Pi_H(m) = \max_{x_1,\dots,x_m \in \mathcal{X}} \left|\left\{\left(h(x_1),\dots,h(x_m)\right): h \in H\right\}\right|$$
  - A sample $S = (x_1, \dots, x_m) \in \mathcal{X}^m$ is said to be **shattered** by H if
  $$\left|\left\{\left(h(x_1),\dots,h(x_m)\right): h \in H\right\}\right| = 2^m$$
  - The Vapnik–Chervonenkis (VC) dimension of H is the size of the largest set that can be shattered by H, namely
  $$VCdim(H) = \sup\{m \colon \Pi_H(m) = 2^m\}$$


$(h(x_1), h(x_2), h(x_3)) = (+,+,+)$
$(h(x_1), h(x_2), h(x_3)) = (+,+,-)$
$(h(x_1), h(x_2), h(x_3)) = (+,-,+)$
$(h(x_1), h(x_2), h(x_3)) = (+,-,-)$
$(h(x_1), h(x_2), h(x_3)) = (-,+,+)$
$(h(x_1), h(x_2), h(x_3)) = (-,+,-)$
$(h(x_1), h(x_2), h(x_3)) = (-,-,+)$
$(h(x_1), h(x_2), h(x_3)) = (-,-,-)$
x_1
x_2
x_3

Let H be the family of binary linear classifiers on $\mathbb{R}^2$, then $S = (x_1, x_2, x_3)$ can be shattered by H, since

$$\left|\left\{\left(h(x_1), h(x_2), h(x_3)\right): h \in H\right\}\right| = |\{(+,+,+),(+,+,-),(+,-,+),(+,-,-),(-,+,+),(-,+,-),(-,-,+),(-,-,-)\}| = 2^3$$

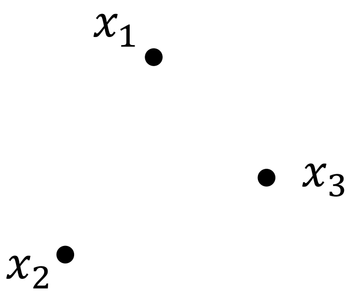
$$\Pi_H(3) = \max_{x_1, x_2, x_3 \in \mathcal{X}} \left|\left\{\left(h(x_1), h(x_2), h(x_3)\right): h \in H\right\}\right| = 8$$

VC Dimension for Binary Classifiers with Hyperplane Decision Boundary

Let H be the family of binary linear classifiers on $\mathbb{R}^2$ ➔$VCdim(H) = 3$

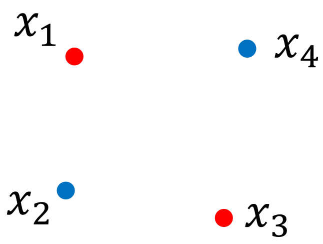
There exists a $S = (x_1, x_2, x_3)$ of size 3 shattered by H

➔ $VCdim(H) \geq 3$



Each sample $S = (x_1, x_2, x_3, x_4)$ of size 4 cannot be shattered by H

➔ $VCdim(H) < 4$



Theorem:

Let H be the family of binary classifiers on $\mathbb{R}^d$ with hyperplane decision boundary, then $VCdim(H) = d + 1$.

(Mohri 2012, Theorem 3.4)

Relation between Rademacher complexity, growth function, and VC dimension

- Let H be a family of binary functions mapping from $\mathcal{X}$ to $\{-1, +1\}$. Then

$$\mathfrak{R}_m(H) \leq \sqrt{\frac{2 \log \Pi_H(m)}{m}}$$

(Mohri 2012, Corollary 3.1)

- If H has VC dimension d, then

$$\Pi_H(m) \leq \left(\frac{em}{d}\right)^d$$

(Mohri 2012, Corollary 3.3)

- Hence with probability at least $1 - \delta$,

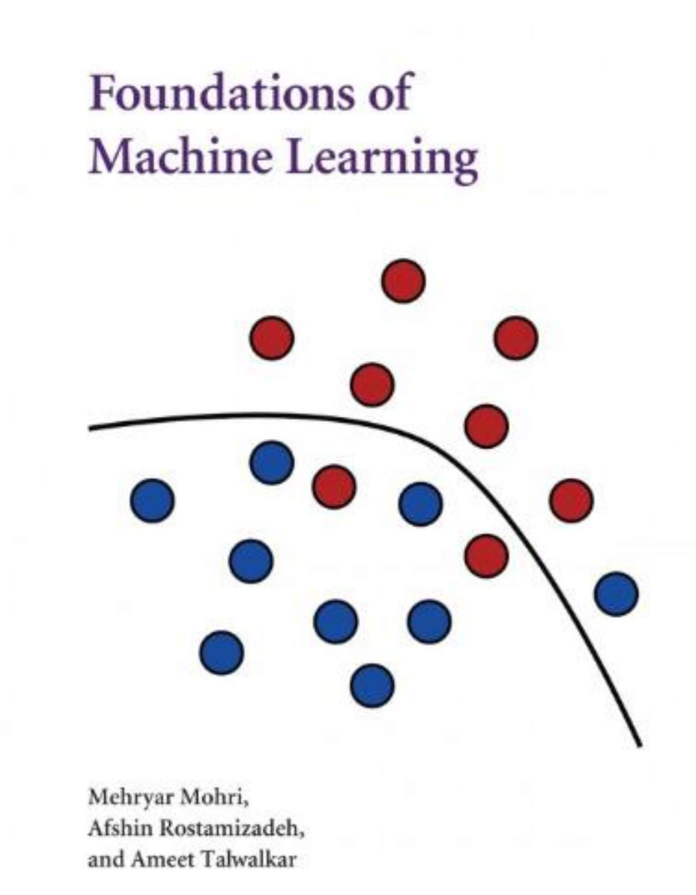
$$\sup_{h \in H}\left(\mathcal{R}(h) - \hat{\mathcal{R}}_S(h)\right) \leq \mathfrak{R}_m(H) + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\leq \sqrt{\frac{2 \log \Pi_H(m)}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

$$\leq \sqrt{\frac{2d \log \frac{em}{d}}{m}} + \sqrt{\frac{\log(1/\delta)}{2m}}$$

Wish to know more?

Foundations of Machine Learning
M. Mohri, A. Rostamizadeh, and A. Talwalkar
MIT Press



Probability in High Dimension
Ramon van Handel
Princeton University (APC 550 Lecture Notes)
https://web.math.princeton.edu/~rvan/APC550.pdf