3.1 PAC Learning

- In the discussion in Statistical Learning Framework, we learned that for a finite hypothesis class, if the ERM rule with respect to that class is applied on a sufficiently large training sample, then the output hypothesis will be probably approximately correct.
- That is the Probably Approximately Correct (PAC) learning

Definition 3.1 (PAC Learnability)

- A hypothesis class H is PAC learnable if there exist:
  - function $m_H : (0,1)^2 \to \mathbb{N}$
  - learning algorithm with the following property:

    For every $\begin{cases} \epsilon, \delta \in (0,1) \\ \text{distribution } D \text{ over } X \\ \text{labeling function } f: X \to \{0,1\} \end{cases}$,

- If the realizability assumption holds with respect to H, D, f, then when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. examples generated by D and labeled by f, the algorithm returns a hypothesis h such that, with probability of at least $1-\delta$ (over the choice of the examples), $L_{D,f}(h) \leq \epsilon$

→ What do ϵ and δ stand for?

These are 2 approximation parameters:

- ϵ : accuracy parameter, determines how far the output classifier can be from the optimal one (approximately - correct)
- δ : confidence parameter, indicating how likely the classifier is to meet that accuracy requirement ("probably" in PAC)

→ continue next page

↳ Why use these approximations (ε and δ) ?

- Since the training set is randomly generated, there is always a chance that the set is noninformative (not giving information about distribution D). For example: it is possible to pick just 1 point in domain X, and sample it over and over again.
- Even if we get a training set S that represents distribution D, since S is finite, there is still some fine details of D that S fails to capture. Thats where our accuracy parameter, ε, allows "forgiving" the learner's classifier for making minor mistakes.

Sample Complexity

How many examples in training set S are required to guarantee a probably approximately correct solution ?

The function $m_H : (0,1)^2 \to \mathbb{N}$ determines the sample complexity of learning H.

Corollary 3.2 (Sample complexity)

Every finite hypothesis class is PAC learnable with sample complexity

$$m_H(\varepsilon, \delta) \leq \left\lceil \frac{\log(|H|/\delta)}{\varepsilon} \right\rceil$$

↳ Note: m (the size) is different from $m_H(\varepsilon, \delta)$ (the threshold)

- Recall from corollary 2.3 of Statistical Learning Framework, we came to the conclusion that the size of training set should be:

$$m \geq \frac{\log(|H|/\delta)}{\varepsilon} = m_H(\varepsilon, \delta)$$

- m_H is defined as the "minimal function" in corollary 3.2. That is, the value of m_H is the smallest possible value that satisfy all the PAC learnability with parameters ε and δ.

3.2 A More General Learning Model

A more generalized model can be achieved by considering 2 aspects:

- Removing the Realizability Assumption
- Learning Problems beyond Binary Classification

Releasing the Realizability Assumption - Agnostic PAC Learning

A more realistic model for the Data-Generating Distribution

- Recall that the realizability assumption said:

$$\exists\, h^* \in H \text{ s.t. } \underset{x \sim D}{P}\left[h^*(x) = f(x) \right] = 1$$

In practical, this is not always true. For our papayas example, we cannot guarantee that 2 papayas with the same color and softness are both tasty.

- Instead, we relax the assumption with a more flexible notion, a data-labels generating distribution, described as follows:
  - D is a joint distribution over $\underbrace{X}_{\text{domain set}} \times \underbrace{Y}_{\text{label set}}$, denoted $D(x, y)$
  - D can be considered to have 2 parts, by chain rule of joint prob:

$$D(x, y) = D(x) \cdot D(y \mid x)$$

- $D(x)$: marginal distribution D_x over unlabeled domain points
- $D(y \mid x)$: conditional distribution $D((x, y) \mid x)$ over labels for each domain point

The empirical and true error revised

Redefined true error

For distribution D over $X \times Y$, prediction rule h:

$$L_D(h) = \mathbb{P}_{(x,y)\sim D}[h(x) \neq y] = D(\{(x,y): h(x) \neq y\})$$

What this means in human language ?

Probability that a data point - label pair (x, y) such that the prediction made by hypothesis h is incorrect.

What is the difference with realizability is assumed ?

Things that contribute to true error

With Realizability Assumption	Without Realizability Assumption
. Representativeness of S	. Representativeness of S
	. Chance of mislabeling S
	. Choice of hypothesis class H

Training error is the same

$$L_S(h) = \frac{|\{i \in [m]: h(x_i) \neq y_i\}|}{m}$$

Why is it the same regardless of realizability assumption.

- Recall that the realizability assumption assumes there is a "perfect" classifier that will make $L_{D,t}(h) = 0$
- Training error measure performance on training set S, so the existence of a "perfect" classifier should not matter.

$$L_S(h) = L_{D(\text{uniform over } S)}(h)$$

- Training loss equals the expected loss under distribution D where each points in S contribute equally to the error.

- Assume "uniform over S" essentially means consider each data points in S will contribute equally to the training error.

→ $L_D(\text{Uniform over } S)(h)$ means ?

- True risk when estimated using sample S, whose each data points are weighted equally
- This is common in ERM, where D is unknown, and the empirical distribution is used as an approximation

The goal

We wish to find some hypothesis, $h: X \to Y$, that minimizes the true risk, $L_D(h)$

The Bayes Optimal Predictor

Given any probability distribution D over $X \times \{0,1\}$, the best predicting function $f: X \to \{0,1\}$ is defined as:

$$f_D(x) = \begin{cases} 1 & \text{if } P(y=1 \mid x) \geq \frac{1}{2} \\ 0 & \end{cases}$$

→ Proof (Exercise 7)

- Let $g: X \to \{0,1\}$ be any other classifier.
  We want to prove that with any g, $L_D(f_D) \leq L_D(g_D)$
- For any classifier g, we can prove that:

$$E_{X,Y}(f(X,Y)) = E_X E_{Y|X=x}(f(X,Y) \mid X=x)$$ Ω

< Law of total expectation, Adam's law >

continue

Therefore: $L_D(g) = E_{(x,y)\sim D}\left(1_{g(x)\neq y}\right)$ < definition >

$= E_{x\sim D_x}\left[E_{y\sim D_{y|x}}\left(1_{g(x)\neq y} \mid X=x\right)\right]$ < proven above >

$= E_{x\sim D_x}\left[P(\{g(x)\neq Y \mid X=x\})\right]$

- We want to prove that, for all $x \in X$, we have:

$$P(\{g(x)\neq Y \mid X=x\}) \geq P(\{f_D(x)\neq Y \mid X=x\})$$

$P(\{g(x)\neq Y \mid X=x\})$

$= 1 - P(\{g(x)=Y \mid X=x\})$ (1)

$= 1 - P(\{g(x)=1, Y=1 \mid X=x\}) - P(\{g(x)=0, Y=0 \mid X=x\})$

LOTP

$P(\{g(x)=1, Y=1 \mid X=x\})$

$= P(\{g(x)=1 \mid X=x\}) \cdot P(\{Y=1 \mid X=x\})$

equals $1_{g(x)=1}$, because

- When $g(x)=1$: $P(\{g(x)=1 \mid X=x\}) = 1$
- When $g(x)=0$: $P(\{g(x)=0 \mid X=x\}) = 0$

LOTP

$P(\{g(x)=0, Y=0 \mid X=x\})$

$= P(\{g(x)=0 \mid X=x\}) \cdot P(\{Y=0 \mid X=x\})$

equals $1_{g(x)=0}$, because:

- When $g(x)=1$: $P(\{g(x)=0 \mid X=x\}) = 0$
- When $g(x)=0$: $P(\{g(x)=0 \mid X=x\}) = 1$

So (1) can be rewritten as:

$$P(\{g(x)=Y \mid X=x\}) = 1_{g(x)=1} P(\{Y=1 \mid X=x\}) + 1_{g(x)=0} P(\{Y=0 \mid X=x\}) \quad (2)$$

Apply similar logic to $P(\{f_D(x)=Y \mid X=x\})$, we have:

$$P(\{f_D(x)=Y \mid X=x\}) = 1_{f_D(x)=1} P(\{Y=1 \mid X=x\}) + 1_{f_D(x)=0} P(\{Y=0 \mid X=x\}) \quad (3)$$

- Let (3) - (2), we have:

$$P(\{ f_D(X) = Y \mid X = x \}) - P(\{ g(X) = Y \mid X = x \})$$

$$= P(\{ Y=1 \mid X=x \}) \left(1_{f_D(x)=1} - 1_{g(x)=1} \right) + P(\{ Y=0 \mid X=x \}) \left(1_{f_D(x)=0} - 1_{g(x)=0} \right)$$

Since:
- $P(\{ Y=0 \mid X=x \}) = 1 - P(\{ Y=1 \mid X=x \})$
- $1_{f_D(x)=0} = 1 - 1_{f_D(x)=1}$
- $1_{g(x)=0} = 1 - 1_{g(x)=1}$

$$= \left[2P(\{ Y=1 \mid X=x \}) - 1 \right] \left(1_{f_D(x)=1} - 1_{g(x)=1} \right) \geq 0$$

Therefore:

$$P(\{ f_D(X) = Y \mid X = x \}) \geq P(\{ g(X) = Y \mid X = x \})$$

(1) $\Leftrightarrow P(\{ f_D(X) \neq Y \mid X = x \}) \leq P(\{ g(X) \neq Y \mid X = x \})$

(2) $\Leftrightarrow E_{x \sim D_X} \left[P(\{ f_D(X) \neq Y \mid X = x \}) \right] \leq E_{x \sim D_X} \left[P(\{ g(X) \neq Y \mid X = x \}) \right]$

(3) $\Leftrightarrow E_{x \sim D_X} \left[E_{y \sim D_{Y|x}} \left(1_{f_D(x) \neq y} \mid X = x \right) \right] \leq E_{x \sim D_X} \left[E_{y \sim D_{Y|x}} \left(1_{g(x) \neq y} \mid X = x \right) \right]$

(4) $\Leftrightarrow E_{(x,y) \sim D} \left[1_{f_D(x) \neq y} \right] \leq E_{(x,y) \sim D} \left[1_{g(x) \neq y} \right]$

(5) $\Leftrightarrow L_D(f_D) \leq L_D(g)$

↳ If Bayes predictor is so good, why not just use it ?

Because we don't know distribution D, so we try to find a predictor whose is as close to Bayes predictor as possible

Definition 3.3 (Agnostic PAC Learnability)

A hypothesis class $\mathcal{H}$ is agnostic PAC learnable if there exist:

- A function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$
- A learning algorithm with the following property:

  For every $\varepsilon, \delta \in (0,1)$ and distribution D over $X \times Y$, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\varepsilon, \delta)$ i.i.d examples generated by D, the algorithm returns a hypothesis h such that, with probability of at least $(1 - \delta)$, over the choice of the m training examples:

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \varepsilon$$

→ How is this difference from Standard PAC Learning ?

Standard PAC

- Best $h \in \mathcal{H}$ has true error = 0

$$L_D(h) = 0$$

- The learning goal is to find $h \in \mathcal{H}$, with high probability $1 - \delta$, such that true error is close to zero, by a margin of ε

$$L_D(h) \leq \varepsilon$$

Agnostic PAC

- Best $h' \in \mathcal{H}$ has true error > 0

$$\min_{h' \in \mathcal{H}} L_D(h') > 0$$

- The learning goal is to find $h \in \mathcal{H}$, with high probability $1 - \delta$, such that true error not much worse than the best possible $h' \in \mathcal{H}$, so:

$$L_D(h) \leq \min_{h' \in \mathcal{H}} L_D(h') + \varepsilon$$

3.2.2 The Scope of Learning Problems Modeled

- Lets extend our model so that it can be applied to a wide variety of learning tasks, some of these tasks are:
  - Multiclass Classification:

    Instead of having binary classification, the label set Y will be the set of categories that is large and finite.
  - Regression:

    In this task, we try to find some simple pattern between domain set X and label set Y
- To accomodate wide range of learning tasks, we generalize our measure of success as follows:

Generalized Loss Functions

Given { any set H (that plays the role of our hypotheses)
      { some domain Z

Let ℓ be any function from $H \times Z$ to the set of nonnegative real numbers,

$$\ell : H \times Z \to \mathbb{R}_+$$

→ Note:

- For prediction task, our $Z = X \times Y$
- But here we generalize beyond prediction task, and therefore allow Z to be any domain of examples.

Risk function

Is the expected loss of a classifier, $h \in H$, with respect to a probability distribution D over Z:

$$L_D(h) = \underset{z \sim D}{E}\left[\ell(h, z) \right]$$

Empirical Risk

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)$$

Some common loss functions, $\ell(h, z)$

- **0-1 loss:** used in binary and multiclass classification problems:

$$\ell_{0-1}(h, (x, y)) = \begin{cases} 0 & \text{if } h(x) = y \\ 1 & \text{if } h(x) \neq y \end{cases}$$

Consequently:

$$L_D(h) = \underset{(x,y)\sim D}{\mathbb{E}} \left[\ell(h, (x, y)) \right]$$

$$= \underset{(x,y)\sim D}{P} \left[h(x) \neq y \right]$$

$$= D(\{(x, y) : h(x) \neq y\})$$

⟨ fundamental bridge: $P(A) = E(I_A)$ ⟩

- **Square loss:** used in regression problems

$$\ell_{sq}(h, (x, y)) = (h(x) - y)^2$$

So, we have completed defining agnostic PAC Learning with general loss function, that is:

Definition 3.4 (Agnostic PAC Learnability for General Loss Functions)

A hypothesis class H is agnostic PAC learnable with respect to a set Z and a loss function $l: H \times Z \to \mathbb{R}_+$, if there exist a function $m_H: (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property:

For every $\varepsilon, \delta \in (0,1)$
distribution D over Z,

When running the algorithm on $m \geq m_H(\varepsilon, \delta)$ i.i.d. examples generated by D, the algorithm returns $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the m training examples)

$$L_D(h) \leq \min_{h' \in H} L_D(h') + \varepsilon$$

where $L_D(h) = \underset{z \sim D}{E}\left[l(h, z) \right]$