ECE595 / STAT598: Machine Learning I
Lecture 24 Probably Approximately Correct

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University

Outline

- Lecture 22 Is Learning Feasible?
- Lecture 23 Probability Inequality
- Lecture 24 Probably Approximate Correct

Today's Lecture:

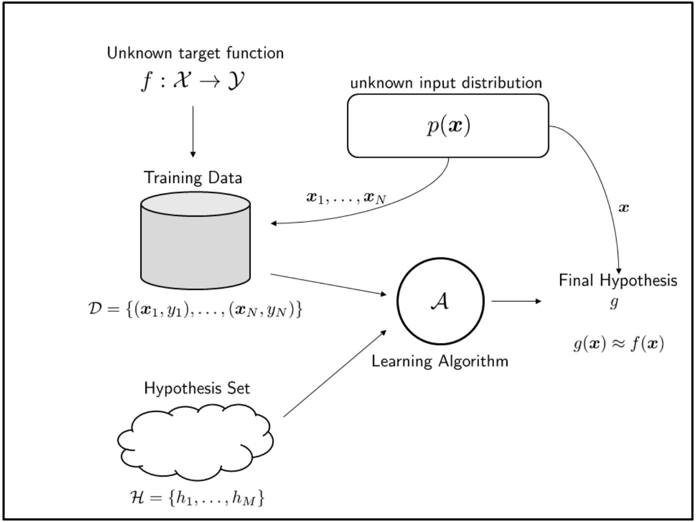
- Two ingredients of generalization
    - Training and testing error
    - Hoeffding inequality
    - Interpreting the bound
- PAC Framework
    - PAC learnable
    - Confidence and accuracy
    - Example

Is Learning Feasible?

- Learning can be **infeasible**.
- Recall the example below.
- Given the training samples, there is no way you can learn and predict.
- You know what you know, and you don't know what you don't know.

$\mathbf{x}_n$			y_n	g	f_1	f_2	f_3	f_4
0	0	0	○	○	○	○	○	○
0	0	1	●	●	●	●	●	●
0	1	0	●	●	●	●	●	●
0	1	1	○	○	○	○	○	○
1	0	0	●	●	●	●	●	●
1	0	1	○	○	○	○	○	○
1	1	0		○/●	○	●	○	●
1	1	1		○/●	○	○	●	●

The Power of Probability



In-Sample Error

- Let $\boldsymbol{x}_n$ be a *training* sample
- h: Your hypothesis
- f: The unknown target function
- If $h(\boldsymbol{x}_n) = f(\boldsymbol{x}_n)$, then say training sample $\boldsymbol{x}_n$ is correctly classified.
- This will give you the **in-sample error**

Definition (In-sample Error / Training Error)

Consider a training set $\mathcal{D} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, and a target function f. The **in-sample error** (or the training error) of a hypothesis function $h \in \mathcal{H}$ is the empirical average of $\{h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)\}$:

$$E_{\text{in}}(h) \overset{\text{def}}{=} \frac{1}{N} \sum_{n=1}^{N} [\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!], \tag{1}$$

where $[\![\cdot]\!] = 1$ if the statement inside the bracket is true, and $= 0$ if the statement is false.

Out-Sample Error

- Let $\boldsymbol{x}$ be a *testing* sample drawn from $p(\boldsymbol{x})$
- h: Your hypothesis
- f: The unknown target function
- If $h(\boldsymbol{x}) = f(\boldsymbol{x})$, then say testing sample $\boldsymbol{x}$ is correctly classified.
- Since $\boldsymbol{x} \sim p(\boldsymbol{x})$, you need to compute the probability of error, called the **out-sample error**

Definition (Out-sample Error / Testing Error)

Consider an input space $\mathcal{X}$ containing elements $\boldsymbol{x}$ drawn from a distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$, and a target function f. The **out-sample error** (or the testing error) of a hypothesis function $h \in \mathcal{H}$ is

$$E_{\text{out}}(h) \overset{\text{def}}{=} \mathbb{P}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})], \tag{2}$$

where $\mathbb{P}[\cdot]$ measures the probability of the statement based on the distribution $p_{\boldsymbol{X}}(\boldsymbol{x})$.

Understanding the Errors

Let us take a closer look at these two error:

$$E_{\text{in}}(h) \stackrel{\text{def}}{=} \frac{1}{N}\sum_{n=1}^{N} [\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!],$$

$$E_{\text{out}}(h) \stackrel{\text{def}}{=} \mathbb{P}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})],$$

- Both error are functions of the hypothesis h
- h is determined by the learning algorithm $\mathcal{A}$
- For every $h \in \mathcal{H}$, there is a different $E_{\text{in}}(h)$ and $E_{\text{out}}(h)$
- The training samples $\boldsymbol{x}_n$ are drawn from $p(\boldsymbol{x})$
- The testing samples $\boldsymbol{x}$ are also drawn from $p(\boldsymbol{x})$
- Therefore, $\mathbb{P}[\cdot]$ in $E_{\text{out}}(h)$ is evaluated over $\boldsymbol{x} \sim p(\boldsymbol{x})$

In-sample VS Out-sample

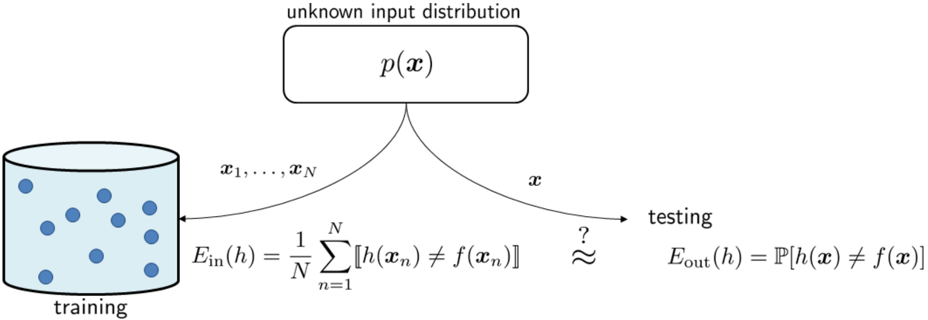
In-Sample Error

$$E_{\text{in}}(h) = \frac{1}{N}\sum_{n=1}^{N} [\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!]$$

Out-Sample Error

$$\begin{aligned} E_{\text{out}}(h) &= \mathbb{P}[h(\boldsymbol{x}) \neq f(\boldsymbol{x})] \\ &= \underbrace{[\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!]}_{=1} \mathbb{P}\Big\{ h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n) \Big\} \\ &\quad + \underbrace{[\![h(\boldsymbol{x}_n) = f(\boldsymbol{x}_n)]\!]}_{=0} \Big(1 - \mathbb{P}\Big\{ h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n) \Big\} \Big) \\ &= \mathbb{E}\Big\{ [\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!] \Big\} \end{aligned}$$

The Role of $p(x)$



- Learning is feasible if $\boldsymbol{x} \sim p(\boldsymbol{x})$
- $p(\boldsymbol{x})$ says: Training and testing are related
- If training and testing are unrelated, then hopeless – the deterministic example shown previously
- If you draw training and testing samples with different bias, then you will suffer

A Mathematical Tool

Beside in-sample and out-sample error, we also need a mathematical tool.

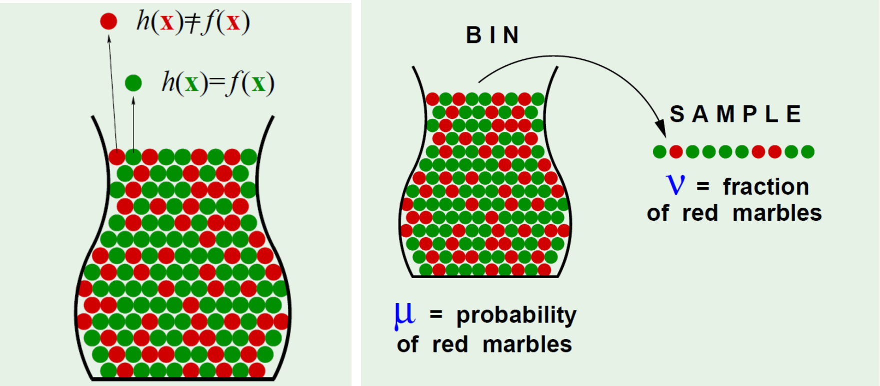
Theorem (Hoeffding Inequality)

Let $X_1, \ldots, X_N$ be random variables with $0 \le X_n \le 1$, then

$$\mathbb{P}\left[|\nu - \mu| > \epsilon\right] \le 2e^{-2\epsilon^2 N}$$

- We will use Hoeffding inequality to analyze the generalization error
- There are many other inequalities that can serve the same purpose
- Hoeffding requires $0 \le X_n \le 1$
- $\nu = \frac{1}{N}\sum_{n=1}^{N} X_n$ is the empirical average
- Probability of how close ν compared to μ
- $\epsilon =$ tolerance level
- $N =$ number of samples

Applying Hoeffinding Inequality to Our Problem



- $X_n = [\![h(\mathbf{x}_n) \neq f(\mathbf{x}_n)]\!]$ = one sample training error = either 0 or 1
- $\nu = E_{\text{out}} = \frac{1}{N}\sum_{n=1}^{N} X_n$ = training error
- $\mu = E_{\text{in}}$ = testing error

Applying Hoeffinding Inequality to Our Problem

- Therefore, the inequality can be stated as

$$\mathbb{P}\left[|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\right] \leq 2e^{-2\epsilon^2 N}.$$

- N = number of training samples
- ϵ = tolerance level
- Hoeffding is applicable because $[\![h(\boldsymbol{x}) \neq f(\boldsymbol{x})]\!]$ is either 1 or 0.
- If you want to be more explicit, then

$$\mathop{\mathbb{P}}_{\boldsymbol{x}_n \sim \mathcal{D}}\left[\left|\frac{1}{N}\sum_{n=1}^{N}[\![h(\boldsymbol{x}_n) \neq f(\boldsymbol{x}_n)]\!] - E_{\text{out}}(h)\right| > \epsilon\right] \leq 2e^{-2\epsilon^2 N}.$$

- The probability is evaluated with respect to $\boldsymbol{x}_n$ drawn from the dataset $\mathcal{D}$

Interpreting the Bound

- Let us look at the bound again:

$$\mathbb{P}\left[|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\right] \leq 2e^{-2\epsilon^2 N}.$$

- **Message 1**: You can bound $E_{\text{out}}(h)$ using $E_{\text{in}}(h)$.
- $E_{\text{in}}(h)$: You know. $E_{\text{out}}(h)$: You don't know, but you want to know.
- They are close if N is large.

- **Message 2**: The right hand side is independent of h and $p(\boldsymbol{x})$
- So it is a universal upper bound
- Works for any $\mathcal{A}$, any $\mathcal{H}$, any f, and any $p(\boldsymbol{x})$

Outline

- Lecture 22 Is Learning Feasible?
- Lecture 23 Probability Inequality
- Lecture 24 Probably Approximate Correct

Today's Lecture:

- Two ingredients of generalization
  - Training and testing error
  - Hoeffding inequality
  - Interpreting the bound
- PAC Framework
  - PAC learnable
  - Confidence and accuracy
  - Example

Accuracy and Confidence

Recall the equation

$$\mathbb{P}\Big[\,|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\Big] \le 2e^{-2\epsilon^2 N}$$

- $\delta = 2e^{-2\epsilon^2 N}$. **confidence**: $1-\delta$.
- $\epsilon = \sqrt{\frac{1}{2N}\log\frac{2}{\delta}}$. **accuracy**: $1-\epsilon$.
- Then the equation becomes

$$\mathbb{P}\Big[\,|E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\Big] \le \delta$$

- which is equivalent to

$$\mathbb{P}\Big[\,|E_{\text{in}}(h) - E_{\text{out}}(h)| \le \epsilon\Big] > 1-\delta$$

Probably Approximately Correct

- **Probably**: Quantify error using probability:

$$\mathbb{P}\left[\,|E_{\text{in}}(h) - E_{\text{out}}(h)| \le \epsilon\right] \ge 1 - \delta$$

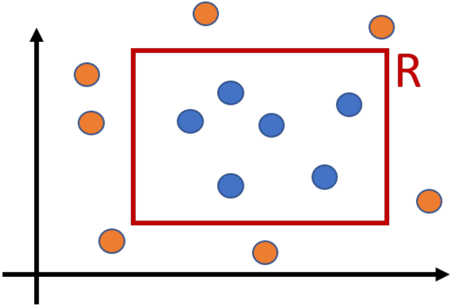
- **Approximately Correct**: In-sample error is an approximation of the out-sample error:

$$\mathbb{P}\left[|E_{\text{in}}(h) - E_{\text{out}}(h)| \le \epsilon\right] \ge 1 - \delta$$

- If you can find an algorithm $\mathcal{A}$ such that for any ϵ and δ, there exists an N which can make the above inequality holds, then we say that the target function is **PAC-learnable**.
- The following example is taken from Mohri et al. Foundation of Machine Learning, Example 2.4.

Example: Rectangle Classifier

Consider a set of 2D data points.



- The target function is a rectangle R
- Inside R: blue. Outside R: orange. Data is intrinsically separable.
- Goal: Pick a hypothesis rectangle R' using the available data point
- Question: Is this problem PAC learnable?

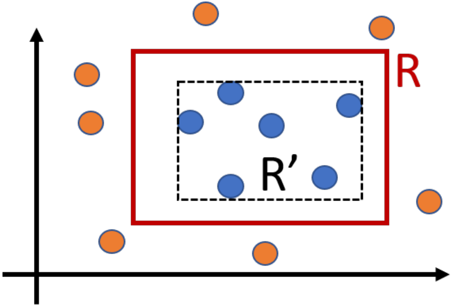
What Shall We Do?

- This question is very general.
- It asks about the nature of the problem.
- We want to show that this problem is indeed PAC learnable.
- To do so, we need to propose an **algorithm** $\mathcal{A}$ which takes the training data and returns an R', such that for any $\epsilon > 0$ and $\delta > 0$, there exists an N (which is a function of ϵ and δ) with

$$\mathbb{P}[|E_{\text{in}}(R') - E_{\text{out}}(R')| > \epsilon] \leq \delta.$$

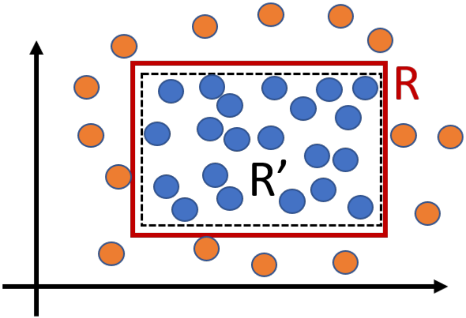
- If we find such algorithm, then the problem is PAC learnable.

- $\mathcal{A}$: Give me the data point points, find the **tightest rectangle** that covers the blue circles.



Intuition

- As N grows, we can find a R' which is getting closer and closer to R.
- So for any $\epsilon > 0$ and $\delta > 0$, it seems possible that as long as N is large enough we will be able to make training error close to testing error.
- See Appendix for proof.



Summary

- Not all problems are learnable.
- Those that are learnable require **training and testing** samples are correlated.
- Then **Hoeffding inequality** applies

$$\mathbb{P}[|E_{\text{out}}(R') - E_{\text{in}}(R')| > \epsilon] \leq \delta.$$

- For any accuracy ϵ and any confidence δ, if you can find an algorithm $\mathcal{A}$ such that as long as N is large enough the above inequality can be proved, then the target function is PAC learnable.
- Next time: Look at the hypothesis set $\mathcal{H}$.

Reading List

- Yasar Abu-Mustafa, Learning from Data, Chapter 1.3, 2.1.
- Mehryar Mohri, Foundations of Machine Learning, Chapter 2.1.
- Martin Wainwright, High Dimensional Statistics, Cambridge University Press 2019. (Chapter 2)
- CMU Note https://www.cs.cmu.edu/~mgormley/courses/10601-s17/slides/lecture28-pac.pdf
- Iowa State Note http://web.cs.iastate.edu/~honavar/pac.pdf
- Princeton Note https://www.cs.princeton.edu/courses/archive/spring08/cos511/scribe_notes/0211.pdf
- Stanford Note http://cs229.stanford.edu/notes/cs229-notes4.pdf

Appendix

How to Prove PAC for the Example?

- First, realize that by the construction of the algorithm, $E_{\text{in}}(R') = 0$.
- No training error, because $\mathcal{A}$ ensures that all blue circles are inside.
- So the probability inequality is simplified from

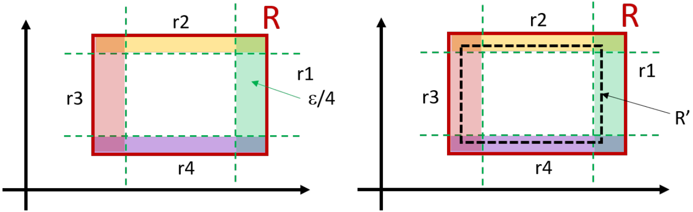
$$\mathbb{P}[|\cancel{E_{\text{in}}(R')} - E_{\text{out}}(R')| > \epsilon] \leq \delta.$$

to (using a different δ):

$$\mathbb{P}[E_{\text{out}}(R') > \epsilon] \leq \delta.$$

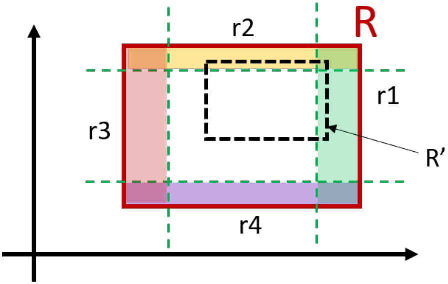
- So just need to evaluate $\mathbb{P}[E_{\text{out}}(R') > \epsilon]$.

Geometric Arguments



- Suppose you give me $\epsilon > 0$. Let us create 4 segments r_1, r_2, r_3, r_4.
- area$(r_i) > \frac{\epsilon}{4}$.
- If R' overlaps with all the four segments, then there exists a ring such that the hypothesis R' will fail to predict.
- Since sum of areas $> \epsilon$, it then follows that $E_{\text{out}}(R') < \epsilon$. (For $E_{\text{out}}(R') > \epsilon$, the hypothesis R' cannot overlap with all four segments.)
- So to analyze $E_{\text{out}}(R') > \epsilon$, we should consider the case where not all segments are overlapped.

Geometric Arguments



- $\mathbb{P}[E_{\text{out}}(R') > \epsilon]$ = Probability that at least one segment does not intersect with R'
- This could mean r_1 or r_2 or r_3 or r_4.
- So

$$\mathbb{P}[E_{\text{out}}(R') > \epsilon] = \mathbb{P}\left[\bigcup_{i=1}^{4}\{R' \cap r_i = \emptyset\}\right].$$

Bounding Out-sample Error

We can evaluate the probability as follows.

$$
\begin{aligned}
\mathbb{P}[E_{\text{out}}(R') > \epsilon] &\leq \mathbb{P}\left[\bigcup_{i=1}^{4}\{R' \cap r_i = \emptyset\}\right] \\
&\leq \sum_{i=1}^{4} \mathbb{P}\left[\{R' \cap r_i = \emptyset\}\right] \qquad \text{union bound} \\
&= \sum_{i=1}^{4} \mathbb{P}\left[\text{all } \boldsymbol{x}_n \text{ are outside } r_i\right] \qquad \text{because } x_n \text{ are covered by } R' \\
&= \sum_{i=1}^{4} (1 - \frac{\epsilon}{4})^N \\
&= 4(1 - \frac{\epsilon}{4})^N \leq 4e^{-\frac{N\epsilon}{4}}. \qquad \text{because} 1 - x < e^{-x}.
\end{aligned}
$$

PAC Learnable!

- Therefore,
$$\mathbb{P}[E_{\text{out}}(R') > \epsilon] \leq 4e^{-\frac{N\epsilon}{4}}$$
- $4e^{-\frac{N\epsilon}{4}} \leq \delta$ if and only if $N \geq \frac{4}{\epsilon}\log\frac{4}{\delta}$.
- So we have found an algorithm $\mathcal{A}$!
- This $\mathcal{A}$ ensures that for any $\epsilon > 0$ and $\delta > 0$, as long as N is larger than $\frac{4}{\epsilon}\log\frac{4}{\delta}$, then we can guarantee
$$\mathbb{P}[E_{\text{out}}(R') > \epsilon] \leq \delta$$
- If you want the two sided bound, we can show that
$$\mathbb{P}[|E_{\text{out}}(R') - E_{\text{in}}(R')| > \epsilon] \leq 2\delta.$$
- Therefore, the problem is PAC learnable.