Lecture 2: PAC Learning

Xiugang Wu

University of Delaware

Fall 2021

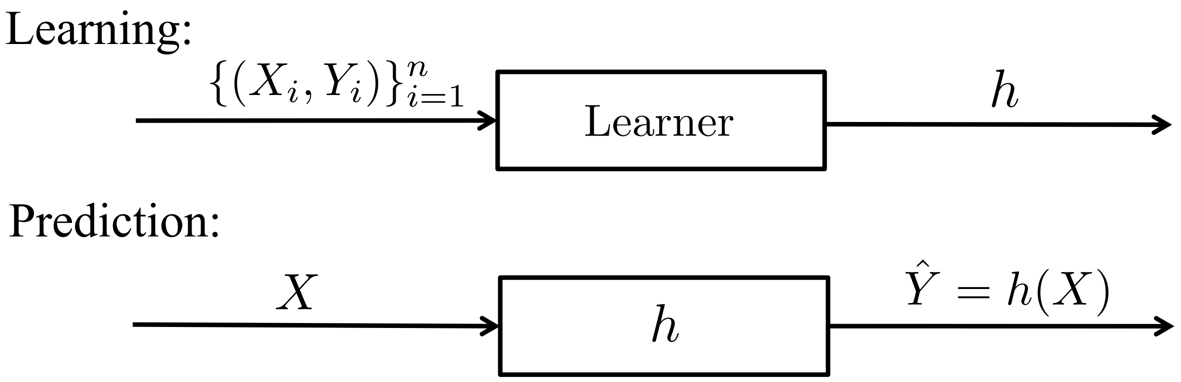
Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

A Simplified Learning Model



Learning:

$\{(X_i, Y_i)\}_{i=1}^n$

Learner

h

Prediction:

X

h

$\hat{Y} = h(X)$

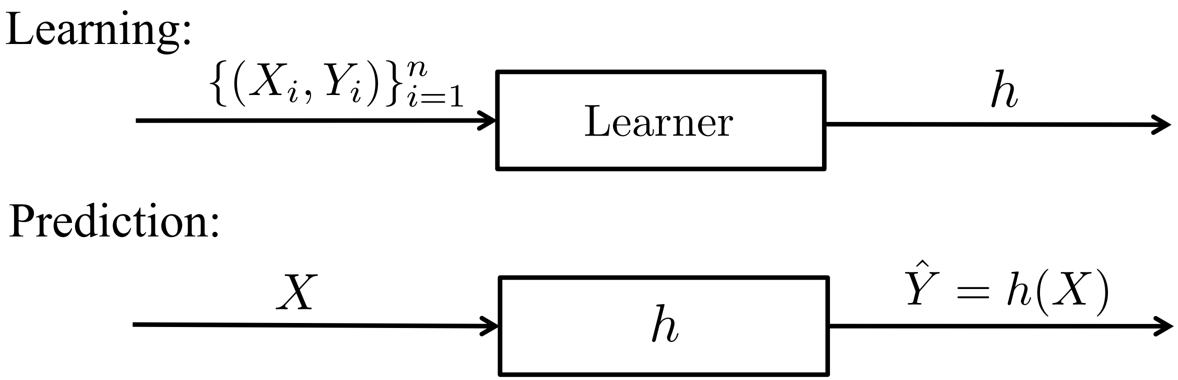
Learner's Input. The learner has access to the following:

- Domain set: An arbitrary set, denoted by $\mathcal{X}$, which contains all the possible inputs. Usually, a domain point (or an instance) x is represented by a vector of *features*.

- Label set: The set of possible outputs, denoted by $\mathcal{Y}$. For our current discussion, we restrict the label set to be $\mathcal{Y} = \{0, 1\}$.

- Training data: The training data $\{(X_i, Y_i)\}_{i=1}^n$ is a finite sequence of (domain point, label) pairs in the product set $\mathcal{X} \times \mathcal{Y}$. For notational convenience, we also write $\mathcal{X} \times \mathcal{Y}$ as $\mathcal{Z}$, and denote the training data by $Z^n = \{(X_i, Y_i)\}_{i=1}^n$.

A Simplified Learning Model



Learning:

$\{(X_i, Y_i)\}_{i=1}^n$

Learner

h

Prediction:

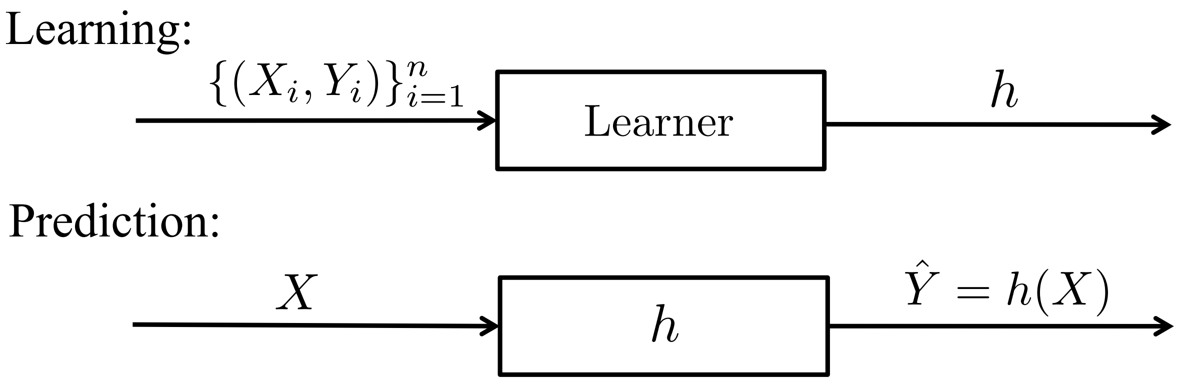
X

h

$\hat{Y} = h(X)$

Learner's Output. The learner outputs a prediction rule $h : \mathcal{X} \to \mathcal{Y}$. This function h is also called a predictor, a hypothesis, or a classifier.

A Simplified Learning Model



Learning:

$\{(X_i, Y_i)\}_{i=1}^n$

Learner

h

Prediction:

X

h

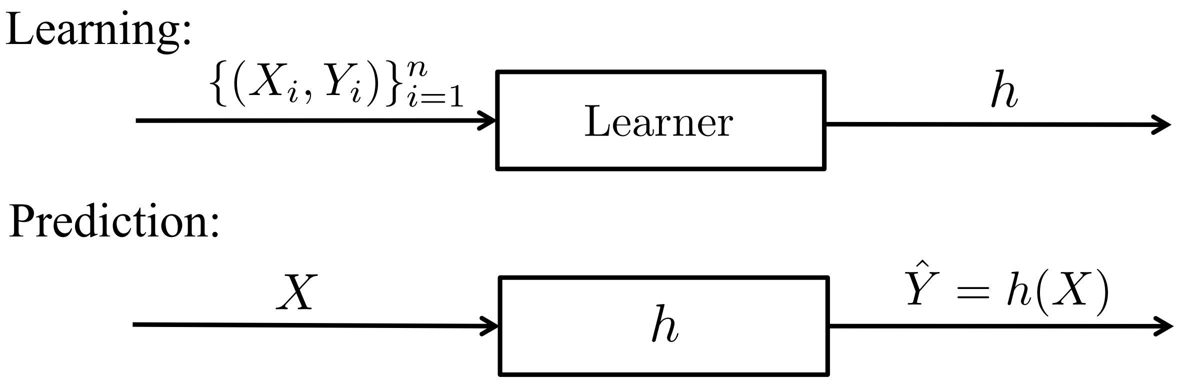
$\hat{Y} = h(X)$

Data-Generation Mechanism.

- Training data: First, instances $\{X_i\}_{i=1}^n$ are i.i.d. generated according to some probability distribution P over $\mathcal{X}$. Then, each instance X_i is labeled according to some labelling function f so that $Y_i = f(X_i)$.
- Test data: The test data point X is generated independently of the training data Z^n, according to the same distribution P.

Note that both P and f are unknown to the learner — in fact, f is exactly what the learner is trying to figure out and for this reason we will also call f the target function.

A Simplified Learning Model



Performance Measure of a Classifier. The probability of error associated with classifier h is given by

$$L(h, P, f) \triangleq \mathbb{P}_{X \sim P}(h(X) \neq f(X)) = P(\{x : h(x) \neq f(x)\}).$$

We will call $L(h, P, f)$ the true error (the true risk, or the test error, interchangeably throughout this course) associated with a classifier h under the distribution P and target labelling function f.

Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

Empirical Risk Minimization

The goal of the learner is to find h_{Z^n} that achieves small error $L(h_{Z^n}, P, f)$.

The learner cannot directly calculate the true error $L(h, P, f)$, but can evaluate the training error $L(h, Z^n)$ defined as

$$L(h, Z^n) = \frac{|\{i \in [1:n] : h(X_i) \neq Y_i\}|}{n}.$$

Note that this training error is in fact the error $L(h, P_n, f_n)$ where

$$P_n(X = x) \triangleq \frac{|\{i \in [1:n] : X_i = x\}|}{n}$$

$$f_n(x) \triangleq \begin{cases} Y_i & \text{if } \exists i \in [1:n] \text{ s.t. } X_i = x \\ 0 & \text{otherwise} \end{cases}.$$

The learning paradigm of coming up with a predictor h that minimizes the empirical risk $L(h, Z^n)$, is called *Empirical Risk Minimization* or simply ERM.

Problem with ERM

Although the ERM approach seems very natural, without being careful, it may fail miserably.

- For example, think of the predictor f_n. Clearly, no matter what the training sample is, f_n results in a training error $L(f_n, P_n, f_n) = 0$ and therefore f_n may be chosen by an ERM algorithm
- However, such f_n may perform very poorly on test data! (Can you think of an example here?)

That said, ERM may lead to a predictor whose performance on the training set is excellent, yet its performance in the true world is very poor. This phenomenon is called *overfitting*.

ERM with Inductive Bias

A solution to the above overfitting problem is to apply ERM learning rule over a restricted search space. In particular, the learner should choose in advance (before seeing the data) a set of predictors. This set is called a hypothesis class and is denoted by $\mathcal{H}$.

$$\mathrm{ERM}_{\mathcal{H}}(Z^n) \in \arg\min_{h\in\mathcal{H}} L(h, Z^n).$$

- By restricting the learner to choosing a predictor from $\mathcal{H}$, we bias it toward a particular set of predictors. Such restrictions are often called an *inductive bias*.
- Since the choice of such a restriction is determined before the learner sees the training data, it should ideally be based on some prior knowledge about the problem to be learned.
- Try to appreciate the formula: "Data + Prior Knowledge = Generalization", if you haven't heard of it or haven't realized its importance. We'll come back to this when introducing "no free lunch theorem".

ERM with Inductive Bias

A solution to the above overfitting problem is to apply ERM learning rule over a restricted search space. In particular, the learner should choose in advance (before seeing the data) a set of predictors. This set is called a hypothesis class and is denoted by $\mathcal{H}$.

$$\mathrm{ERM}_{\mathcal{H}}(Z^n) \in \arg\min_{h\in\mathcal{H}} L(h, Z^n).$$

A fundamental question in learning theory is, over which hypothesis classes $\mathrm{ERM}_{\mathcal{H}}$ learning will not result in overfitting. We will study this question later in the course, in particular when we introduce the VC theory.

Also, intuitively, choosing a more restricted hypothesis class better protects us against overfitting but at the same time might cause us a stronger inductive bias. We will get back to this fundamental (so-called complexity-bias) tradeoff later as well.

Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

Finite Hypothesis Class: Realizable Case

In this lecture, we consider perhaps the simplest type of restriction on a hypothesis class, i.e. imposing an upper bound on its size. We will show that if $\mathcal{H}$ is a finite class then $\mathrm{ERM}_{\mathcal{H}}$ will not overfit if the training sample is sufficiently large.

Particularly, for now also assume that $\mathcal{H}$ satisfies the realizability assumption: there exists $h^* \in \mathcal{H}$ such that $L(h^*, P, f) = 0$. Note that the realizability assumption implies that the training error $L(h_{Z^n}, Z^n)$ using $\mathrm{ERM}_{\mathcal{H}}$ algorithm always equals to 0.

Formalize "Successful Learning"

Since the training set Z^n is randomly generated, there is randomness in the choice of h_{Z^n} and hence the true risk $L(h_{Z^n}, P, f)$ is a random variable depending on the training set Z^n. What we desire to show is that for sufficiently large training sample, we can achieve

$$P^n(L(h_{Z^n}, P, f) \leq \epsilon) \geq 1 - \delta,$$

where ϵ is called the *accuracy parameter* and δ is called the *confidence parameter*.

- First it is not realistic to hope to find "exactly" correct h_{Z^n} such that

$$L(h_{Z^n}, P, f) = 0.$$

- Even relaxing to "approximately" correct h_{Z^n} such that $L(h_{Z^n}, P, f) \leq \epsilon$, it is not realistic to expect that with full certainty Z^n will suffice to direct the learner toward a good classifier, as there is always some probability that the sampled training data happens to be very non-representative of the underlying P.

Probability of Failure of ERM Learner

We interpret the event $L(h_{Z^n}, P, f) > \epsilon$ as a failure of the learner, while if $L(h_{Z^n}, P, f) \leq \epsilon$ we view the output of the algorithm as an *approximately correct* predictor. We are interested in upper bounding the probability of encountering such training sample Z^n that leads to failure of the learner, i.e. $P^n(L(h_{Z^n}, P, f) > \epsilon)$.

For this, let $\mathcal{H}_B$ be the set of bad hypothesis that incurs a high test error, i.e.,

$$\begin{aligned}\mathcal{H}_B &= \{h \in \mathcal{H} : L(h, P, f) > \epsilon\} \\ &= \{h \in \mathcal{H} : P(h(X) \neq f(X)) > \epsilon\}.\end{aligned}$$

In addition, let $\mathcal{M}$ be the set of misleading training samples, under which there is some bad hypothesis that looks like a good hypothesis, i.e.,

$$\begin{aligned}\mathcal{M} &= \{z^n : \exists h \in \mathcal{H}_B \text{ s.t. } L(h, z^n) = 0\} \\ &= \bigcup_{h \in \mathcal{H}_B} \{z^n : L(h, z^n) = 0\}.\end{aligned}$$

Probability of Failure of ERM Learner

Note that the failure of ERM learner, i.e. the event $L(h_{Z^n}, P, f) \geq \epsilon$, can only happen if Z^n falls into the set $\mathcal{M}$ of misleading samples. Therefore,

$$\begin{aligned} P^n(L(h_{Z^n}, P, f) \geq \epsilon) &\leq P^n(Z^n \in \mathcal{M}) \\ &\leq \sum_{h \in \mathcal{H}_B} P^n(L(h, Z^n) = 0). \end{aligned}$$

Since $L(h, Z^n) = 0$ iff $h(X_i) = f(X_i), \forall i \in [1:n]$, we have for any $h \in \mathcal{H}_B$ that

$$\begin{aligned} P^n(L(h, Z^n) = 0) &= P^n(h(X_i) = f(X_i), \forall i \in [1:n]) \\ &= \prod_{i=1}^{n} P(h(X_i) = f(X_i)) \\ &\leq (1-\epsilon)^n. \end{aligned}$$

Combining the above we have

$$P^n(L(h_{Z^n}, P, f) \geq \epsilon) \leq |\mathcal{H}_B|(1-\epsilon)^n \leq |\mathcal{H}|e^{-n\epsilon}.$$

Summary

Corollary: Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0, 1)$ and $\epsilon > 0$ and let n be an integer that satisfies

$$n \geq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}.$$

Then, for any labeling function f and distribution P, for which the realizability assumption holds (that is, for some $h \in \mathcal{H}, L(h, P, f) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample Z^n, we have that for every ERM returned predictor, h_{Z^n}, it holds that

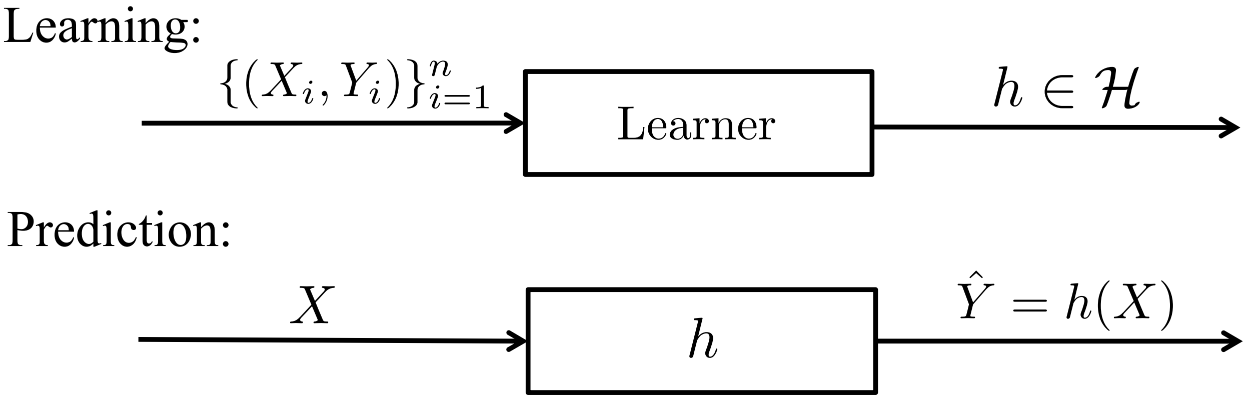
$$L(h_{Z^n}, P, f) \leq \epsilon.$$

The preceding corollary tells us that for a sufficiently large sample size n, the $\mathrm{ERM}_{\mathcal{H}}$ rule over a finite hypothesis class will be probably (with confidence $1-\delta$) approximately (up to an error of ϵ) correct. Next we formally define the model of Probably Approximately Correct (PAC) learning.

Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

PAC Learning Model



Learning:

$\{(X_i, Y_i)\}_{i=1}^{n}$

Learner

$h \in \mathcal{H}$

Prediction:

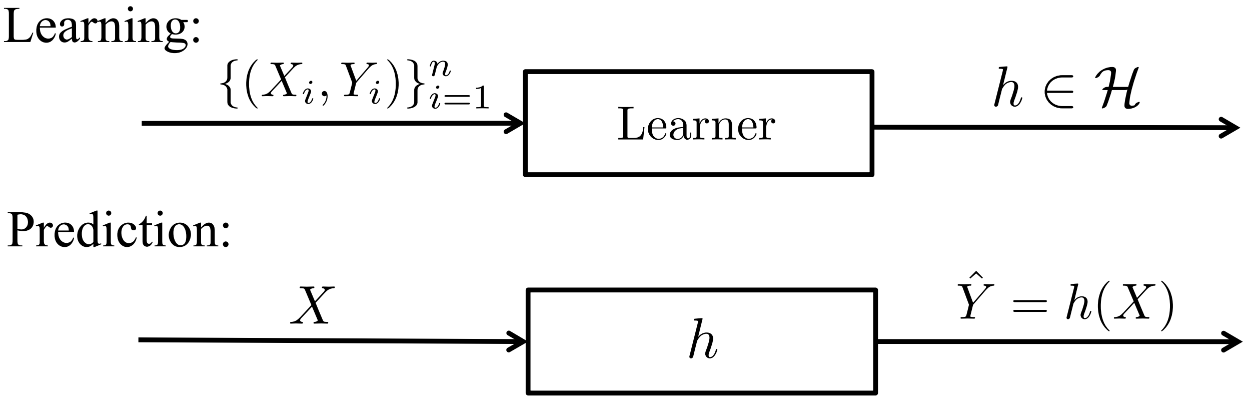
X

h

$\hat{Y} = h(X)$

Label set. We now extend our model by relaxing the binary label set $\mathcal{Y} = \{0, 1\}$ to be the set of real vectors or the set of multiple labels. This allow us to include *regression* or *multiclass classification* problems. The generalized $\mathcal{Y}$ set is also often referred to as the *target* set.

PAC Learning Model



Data-Generation Mechanism. We will consider a joint distribution P_{XY}, or simply P, over $\mathcal{X} \times \mathcal{Y}$. One can view such a distribution as being composed of two parts: a distribution P_X over unlabeled domain points and a conditional distribution over labels for each domain point, $P_{Y|X}$.

PAC Learning Model

Performance Measure of a Predictor. We now introduce a general framework of quantifying the performance of a predictor. Given a target set $\mathcal{Y}$ and a reconstructed target set $\hat{\mathcal{Y}}$, let ℓ be any function from $\mathcal{Y} \times \hat{\mathcal{Y}}$ to the set of non-negative real numbers, $\ell : \mathcal{Y} \times \hat{\mathcal{Y}} \to \mathbb{R}_+$. Such functions are referred to as *loss functions* as they are used to quantify how bad we feel about our reconstruction $\hat{y}$ once we find out the ground truth y. Define the risk $L(h, P)$ associated with a predictor h under data-generating distribution P as the expected loss when applying h to X, i.e.

$$L(h, P) \triangleq \mathbb{E}_{(X,Y)\sim P}[\ell(Y, h(X))].$$

This risk is also called the true risk as it statistically measures the true performance of predictor h on unseen data. In contrast, one can also consider the empirical risk that the predictor h incurs over the training sample,

$$\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, h(x_i)).$$

It can be readily seen that the the empirical risk is simply the risk of h evaluated under the empirical distribution P_n, i.e. $L(h, P_n)$.

Loss Functions

0-1 loss: The 0-1 loss, widely used in classification, is defined as

$$\ell_{0-1}(y, \hat{y}) = 1 - 1_y(\hat{y}).$$

The risk of h under distribution P and loss function ℓ_{0-1} is simply the probability of error,

$$\mathbb{E}_P[\ell_{0-1}(Y, h(X))] = P(Y \neq h(X)).$$

Square loss: The square loss, also known as ℓ_2 loss or quadratic loss, is usually used in the regression problem and is defined as

$$\ell_{\text{sq}}(y, \hat{y}) = \|y - \hat{y}\|^2.$$

The risk of h under distribution P and loss function ℓ_{sq} is generally known as the Mean Square Error (MSE),

$$\mathbb{E}_P[\ell_{\text{sq}}(Y, h(X))] = \mathbb{E}_P[\|Y - h(X)\|^2].$$

Loss Functions

Logarithmic loss: The logarithmic loss (or log loss in short), also known as the cross entropy loss, is a loss function widely used in classification when the reconstruction is "soft" and $\hat{y}$ represents a distribution over $\mathcal{Y}$,

$$\ell_{\log}(y, \hat{y}) = \log \frac{1}{\hat{y}(y)} = H(1_y, \hat{y}),$$

where $H(p, q)$ is the cross entropy between two distributions p and q:

$$H(p, q) \triangleq \sum_{y \in \mathcal{Y}} p(y) \log \frac{1}{q(y)}.$$

The risk of h under distribution P and loss function $\ell_{\log}$ is given by

$$\mathbb{E}_P[\ell_{\log}(Y, h(X))] = \mathbb{E}_P[-\log [h(X)](Y)] = \mathbb{E}_{P_X}[H(P_{Y|X}, h(X))].$$

Bayes Predictor

Suppose that one knows the underlying distribution P. The predictor

$$f = \arg\min_{h} L(h, P)$$

that minimizes the true risk is called the *Bayes predictor*, or *Bayes estimator*, or *Bayes decision rule*, and its resultant risk

$$\min_{h} L(h, P)$$

is called the *Bayes risk*.

- 0-1 loss: Under the 0-1 loss, the Bayes predictor f is given by the well-known maximum a posteriori (MAP) rule, i.e.,

  $$f(x) = \arg\max_{y \in \mathcal{Y}} p_{Y|X}(y|x),$$

  with the Bayes risk

  $$L(f, P) = 1 - \sum_{x \in \mathcal{X}} \max_{y \in \mathcal{Y}} p_{X,Y}(x, y).$$

Bayes Predictor

Suppose that one knows the underlying distribution P. The predictor

$$f = \arg\min_h L(h, P)$$

that minimizes the true risk is called the *Bayes predictor*, or *Bayes estimator*, or *Bayes decision rule*, and its resultant risk

$$\min_h L(h, P)$$

is called the *Bayes risk*.

- Square loss: Under the square loss, the Bayes predictor f is given by the conditional expectation of Y given $X = x$, i.e.,

  $$f(x) = \mathbb{E}_P[Y|X = x],$$

  with the Bayes risk

  $$L(f, P) = \mathbb{E}_P[\mathrm{Var}(Y|X)].$$

Bayes Predictor

Suppose that one knows the underlying distribution P. The predictor

$$f = \arg\min_h L(h, P)$$

that minimizes the true risk is called the *Bayes predictor*, or *Bayes estimator*, or *Bayes decision rule*, and its resultant risk

$$\min_h L(h, P)$$

is called the *Bayes risk*.

- Log loss: Under the log loss, the Bayes predictor f is given by the conditional distribution of Y given $X = x$, i.e.,

  $$[f(x)](y) = p_{Y|X}(y|x),$$

  with the Bayes risk being the conditional entropy of Y given X:

  $$L(f, P) = \mathbb{E}_{(X,Y)\sim P}[-\log p_{Y|X}(Y|X)] = H_P(Y|X).$$

Bayes Predictor

Suppose that one knows the underlying distribution P. The predictor

$$f = \arg\min_h L(h, P)$$

that minimizes the true risk is called the *Bayes predictor*, or *Bayes estimator*, or *Bayes decision rule*, and its resultant risk

$$\min_h L(h, P)$$

is called the *Bayes risk*.

Unfortunately, since we do not know P, we cannot utilize the above Bayes predictors to achieve the minimal possible error. Instead, what the learner does have access to is the training sample. So we will choose some hypothesis class, and require that the learner will, based on the training sample, find a predictor whose error is not much larger than the best possible error achievable by any hypothesis within the class.

PAC Learnability

Definition (PAC Learnability): A hypothesis class $\mathcal{H}$ is PAC (Probably Approximately Correct) learnable if there exist a function $n_{\mathcal{H}} : (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$ and every distribution P, if $n \geq n_{\mathcal{H}}$, then

$$\mathbb{P}_{Z^n \sim P^n} \left(L(h_{Z^n}, P) \leq \min_{h \in \mathcal{H}} L(h, P) + \epsilon \right) \geq 1 - \delta.$$

- Accuracy and confidence parameters: The definition of PAC learnability contains two approximation parameters mentioned before. The accuracy parameter ϵ determines how far the output predictor can be from the optimal one within the class (this corresponds to the "approximately correct"), and the confidence parameter δ indicates how likely the output predictor is to meet that accuracy requirement (corresponds to the "probably" part of "PAC").

PAC Learnability

Definition (PAC Learnability): A hypothesis class $\mathcal{H}$ is PAC (Probably Approximately Correct) learnable if there exist a function $n_{\mathcal{H}} : (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$ and every distribution P, if $n \geq n_{\mathcal{H}}$, then

$$\mathbb{P}_{Z^n \sim P^n} \left(L(h_{Z^n}, P) \leq \min_{h \in \mathcal{H}} L(h, P) + \epsilon \right) \geq 1 - \delta.$$

- Sample complexity: The function $n_{\mathcal{H}}$ determines the sample complexity of learning $\mathcal{H}$, that is, how many examples at least are required to guarantee a probably approximately correct solution. The sample complexity is a function of the accuracy and confidence parameters. It also depends on properties of the hypothesis class $\mathcal{H}$ — for example, we showed that for a finite class satisfying the realizability assumption the sample complexity depends on log of the size of $\mathcal{H}$. In fact, using the above definition of PAC learnability, one can rephrase that result as the following: Every finite hypothesis class $\mathcal{H}$ satisfying the realizability assumption is PAC learnable with sample complexity $n_{\mathcal{H}}(\epsilon, \delta) \leq \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}$.

PAC Learnability

Definition (PAC Learnability): A hypothesis class $\mathcal{H}$ is PAC (Probably Approximately Correct) learnable if there exist a function $n_{\mathcal{H}} : (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$ and every distribution P, if $n \geq n_{\mathcal{H}}$, then

$$\mathbb{P}_{Z^n \sim P^n} \left(L(h_{Z^n}, P) \leq \min_{h \in \mathcal{H}} L(h, P) + \epsilon \right) \geq 1 - \delta.$$

- Agnostic PAC learning: This framework is also generally known as agnostic PAC learning as it doesn't assume realizability. We will often omit the prefix "agnostic" in this course, in which case PAC learning refers to this general agnostic case.

Outline

- A Simple Learning Model
- Empirical Risk Minimization
- Finite Hypothesis Class: Realizable Case
- PAC Learning Model
- Finite Hypothesis Class: Agnostic Case

Uniform Convergence Is Sufficient For Learnability

For ERM to work, it suffices to ensure that the empirical risk of all hypothesis in $\mathcal{H}$ are good approximations of their true risk.

Definition: A training sequence Z^n is called ϵ-representative if

$$\forall h \in \mathcal{H}, |L(h, P_n) - L(h, P)| \leq \epsilon.$$

Lemma: If Z^n is $\epsilon/2$-representative, then the output h_{Z^n} of $\mathrm{ERM}_{\mathcal{H}}(Z^n)$ satisfies

$$L(h_{Z^n}, P) \leq L(h^*, P) + \epsilon,$$

where we assume that h^* achieves the minimum risk within the class $\mathcal{H}$.

Proof: $L(h_{Z^n}, P) \leq L(h_{Z^n}, P_n) + \epsilon/2 \leq L(h^*, P_n) + \epsilon/2 \leq L(h^*, P) + \epsilon/2 + \epsilon/2.$

Uniform Convergence Is Sufficient For Learnability

The last lemma implies that to ensure that ERM is a PAC learner, it suffices to show that with probability of at least $1-\delta$, Z^n is ϵ-representative. The uniform convergence condition formalizes this requirement.

Definition (Uniform convergence): We say $\mathcal{H}$ has the uniform convergence property if there exists a function $n_{\mathcal{H}}^{\mathrm{UC}} : (0,1)^2 \to \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and P, if $Z^n \sim P$ with $n \geq n_{\mathcal{H}}^{\mathrm{UC}}$ then with probability of at least $1 - \delta$, Z^n is ϵ-representative.

Corollary: If $\mathcal{H}$ has the uniform convergence property with a function $n_{\mathcal{H}}^{\mathrm{UC}}$, then the class is PAC learnable with sample complexity $n_{\mathcal{H}}(\epsilon, \delta) \leq n_{\mathcal{H}}^{\mathrm{UC}}(\epsilon/2, \delta)$, and in this case ERM is a successful PAC learner for $\mathcal{H}$.

Finite Classes Are Agnostic PAC Learnable

We now show that finite classes are agnostic PAC learnable by showing that uniform convergence holds for any finite hypothesis class. For this, we first introduce a measure concentration inequality due to Hoeffding, which quantifies the gap between empirical averages and their expected value.

Lemma (Hoeffding's Inequality): Let $X_1, X_2, \ldots, X_n$ be a sequence of independent random variables and assume that for all i, $\mathbb{E}[X_i] = \mu$ and $\mathbb{P}(X_i \in [a,b]) = 1$. Then, for any $\epsilon > 0$,

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n} X_i - \mu\right| \geq \epsilon\right) \leq 2e^{-\frac{2n\epsilon^2}{(b-a)^2}}.$$

Finite Classes Are Agnostic PAC Learnable

Now consider the empirical risk of any $h \in \mathcal{H}$ under training sample Z^n,

$$L(h, P_n) = \frac{1}{n} \sum_{i=1}^{n} \ell(Y_i, h(X_i)),$$

where $\ell(Y_i, h(X_i)), i \in [1:n]$ are i.i.d. with mean $L(h, P)$ for any $i \in [1:n]$. Let us further assume that the range of ℓ is $[0, 1]$. Then applying Hoeffding's inequality to the sequence of $\ell(Y_i, h(X_i))$, we obtain

$$P^n \left(|L(h, P_n) - L(h, P)| \geq \epsilon \right) \leq 2e^{-2n\epsilon^2},$$

and therefore

$$P^n \left(\exists h \in \mathcal{H}, \text{ s.t. } |L(h, P_n) - L(h, P)| \geq \epsilon \right) \leq 2|\mathcal{H}|e^{-2n\epsilon^2}.$$

This shows that if

$$n \geq \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2},$$

then

$$P^n \left(|L(h, P_n) - L(h, P)| \leq \epsilon, \forall h \in \mathcal{H} \right) \geq 1 - \delta,$$

Summary

Corollary: Let $\mathcal{H}$ be a finite hypothesis class and ℓ be a loss function with range $[0, 1]$. Then $\mathcal{H}$ enjoys the uniform convergence property with sample complexity

$$n_{\mathcal{H}}^{\mathrm{UC}}(\epsilon, \delta) \leq \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}.$$

Furthermore, the class is agnostically PAC learnable using the ERM algorithm with sample complexity

$$n_{\mathcal{H}}(\epsilon, \delta) \leq n_{\mathcal{H}}^{\mathrm{UC}}(\epsilon/2, \delta) = \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2}.$$