ELEG/CISC 867: Advanced Machine Learning **Spring 2019**

Lecture 2: A Simplified Learning Model

Lecturer: Xiugang Wu *02/19/2019 & 02/21/2019*

Last time, we have introduced the basic idea of machine learning using the example of image classification. From this lecture, we will start our mathematical analysis of such learning problems. As a gentle start, the lecture today will introduce a simplified learning model, and demonstrate how successful learning can be achieved under this model.

1 A Simple Learning Model

Recall the following general statistical learning framework we introduced in the last lecture.

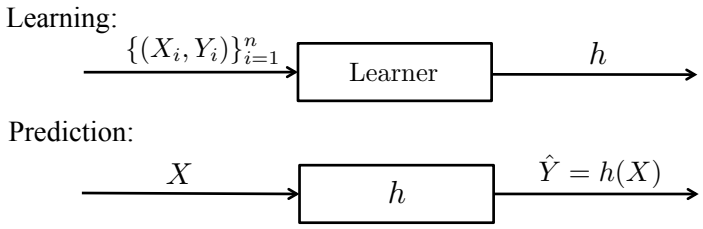

Figure 1: Supervised machine learning.

Later in this course, we will discuss this framework in its full generality. In today's lecture, however, we will make the following assumptions to simplify our discussion.

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. For example, it could be the set of images in the image classification task. 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 $\mathcal{Y} = \{0, 1\}$, where, e.g., 0 could represent label "cat" and 1 could represent label "dog" in image classification.
- 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 $Z^n = \{(X_i, Y_i)\}_{i=1}^n$.

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

Data-Generation Mechanism. The training data is generated in the following manner. First, instances $\{X_i\}_{i=1}^n$ are i.i.d. generated according some probability distribution P over $\mathcal{X}$. Then, each instance X_i is labeled according to some labelling function f so that $Y_i = f(X_i)$, for any $i \in [1 : n]$. The testing data point X is generated independently of the training data Z^n, according to the same distribution P.

Note that both the probability distribution P and labelling function 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.

Performance Measure of a Classifier. We measure the performance of the learned classifier h by looking at its resultant probability of error during the prediction stage, i.e.,

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

where we have assumed that the data generating distribution is P and the target function is f. 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. Note that here we use the letter "L" for "error" since we can view this error as the *loss* of the learner — next lecture we will see other formulations of such loss.

2 Empirical Risk Minimization

We now describe a simple learning paradigm for the above setup and analyze its performance.

As we have seen, a learning algorithm takes a training sequence Z^n as input, which is sampled from an unknown distribution P and labeled by some target function f, and outputs a predictor $h_{Z^n} : \mathcal{X} \to \mathcal{Y}$, where we use the subscript Z^n to emphasize the dependence of the learned predictor on Z^n. The goal of the learning algorithm is to find h_{Z^n} that achieves small generalization error $L(h_{Z^n}, P, f)$ even though the underlying distribution P and target function f are unknown to the learner.

Since the learner doesn't know what P and f are, it cannot directly calculate the true error $L(h, P, f)$ associated with a predictor h. Instead, what the learner can calculate is the training error $L(h, Z^n)$ — the error a predictor h incurs over the training sample 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 $L(h, P_n, f_n)$ of h evaluated under the empirical distribution P_n and the labelling function 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 terms empirical error and empirical risk are also interchangeably used for this error. Since the training sample is a snapshot of the world that is available to the learner, it makes sense to search for a solution that works well on the training data. This learning pradigm – coming up with a predictor h that minimizes the empirical risk $L(h, Z^n)$ – is called *Empirical Risk Minimization* or simply ERM.

2.1 ERM with Inductive Bias

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 testing 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 on the true world is very poor. This phenomenon is called *overfitting*.

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 $\mathcal{H}$. Each $h \in \mathcal{H}$ is a function mapping from $\mathcal{X}$ to $\mathcal{Y}$. For a given class $\mathcal{H}$ and a training sample Z^n, the $\text{ERM}_\mathcal{H}$ learner uses the ERM rule to choose a predictor $h \in \mathcal{H}$ with the smallest training error over Z^n, i.e.,

$$\text{ERM}_\mathcal{H}(Z^n) \in \operatorname*{argmin}_{h \in \mathcal{H}} L(h, Z^n).$$

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".)

A fundamental question in learning theory is, over which hypothesis classes $\text{ERM}_\mathcal{H}$ learning will not result in overfitting. We will study this question later in the course. 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 tradeoff later as well.

3 Finite Hypothesis Classes with Realizability Assumption

We now 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 $\text{ERM}_\mathcal{H}$ will not overfit if the training sample is sufficiently large.

Assume that $\mathcal{H}$ is a finite class which also satisifies 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 $\text{ERM}_\mathcal{H}$ algorithm always equals to 0. But how about the true error $L(h_{Z^n}, P, f)$?

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*. Why shall we have these two parameters?

- First, it is not realistic to hope to find "exactly" correct h_{Z^n} such that $L(h_{Z^n}, P, f) = 0$. As an example, for every $\epsilon \in (0, 1)$, let $\mathcal{X} = \{x_1, x_2\}$, and $P(X = x_1) = 1 - \epsilon$ while $P(X = x_2) = \epsilon$. Then the probability of not seeing x_2 at all in Z^n is $(1 - \epsilon)^n \leq e^{-n\epsilon}$. So if $\epsilon \ll 1/n$ we are likely not to see x_2 at all and thus cannot know its label.
- Second, 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.

3.1 The Probability of Failure of the 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 generalization 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.,

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

Note that the failure of the 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, we have

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

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

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

where the second equality is due to the independence among the n training examples, and the last equality holds because for any bad hypothesis $h \in \mathcal{H}_B$, $P(h(X) \neq f(X)) > \epsilon$. Combining the above we have

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

where we have used the fact that $x + 1 \leq e^x$.

This immediately leads to the following corollary:

Corollary 3.1 *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 $\text{ERM}_{\mathcal{H}}$ rule over a finite hypothesis class will be probably (with confidence $1 - \delta$) approximately (up to an error of ϵ) correct. In the next lecture we will formally define the model of Probably Approximately Correct (PAC) learning.