pattern recognition problem, we could set

$$f(x) = \begin{cases} y_i & \text{if } x = x_i \text{ for some } i = 1, \ldots, m \\ 1 & \text{otherwise.} \end{cases} \tag{5.4}$$

This does not amount to any form of learning, however: suppose we are now given a test point drawn from the same distribution, $(x, y) \sim P(x, y)$. If $\mathcal{X}$ is a continuous domain, and we are not in a degenerate situation, the new pattern x will almost never be exactly equal to any of the training inputs x_i. Therefore, the learning machine will almost always predict that $y = 1$. *If we allow* all *functions from $\mathcal{X}$ to $\mathcal{Y}$, then the values of the function at points $x_1, \ldots, x_m$ carry no information about the values at other points.* In this situation, a learning machine cannot do better than chance. This insight lies at the core of the so-called *No-Free-Lunch Theorem* popularized in [608]; see also [254, 48].

The message is clear: if we make no restrictions on the class of functions from which we choose our estimate f, we cannot hope to learn anything. Consequently, machine learning research has studied various ways to implement such restrictions. In statistical learning theory, these restrictions are enforced by taking into account the *complexity* or *capacity* (measured by VC dimension, covering numbers, entropy numbers, or other concepts) of the class of functions that the learning machine can implement.[1]

In the Bayesian approach, a similar effect is achieved by placing *prior distributions $P(f)$* over the class of functions (Chapter 16). This may sound fundamentally different, but it leads to algorithms which are closely related; and on the theoretical side, recent progress has highlighted intriguing connections [92, 91, 353, 238].

5.2 The Law of Large Numbers

Let us step back and try to look at the problem from a slightly different angle. Consider the case of pattern recognition using the misclassification loss function. Given a fixed function f, then for each example, the loss $\xi_i := \frac{1}{2}|f(x_i) - y_i|$ is either

1. As an aside, note that the same problem applies to *training on the test set* (sometimes called *data snooping*): sometimes, people optimize tuning parameters of a learning machine by looking at how they change the results on an independent test set. Unfortunately, once one has adjusted the parameter in this way, the test set is not independent anymore. This is identical to the corresponding problem in training on the *training* set: once we have chosen the function to minimize the training error, the latter no longer provides an unbiased estimate of the test error. Overfitting occurs much faster on the training set, however, than it does on the test set. This is usually due to the fact that the number of tuning parameters of a learning machine is much smaller than the total number of parameters, and thus the capacity tends to be smaller. For instance, an SVM for pattern recognition typically has two tuning parameters, and optimizes m weight parameters (for a training set size of m). See also Problem 5.3 and [461].

0 or 1 (provided we have a ± 1-valued function f), and all examples are drawn independently. In the language of probability theory, we are faced with *Bernoulli trials*. The $\xi_1, \ldots, \xi_m$ are independently sampled from a random variable

$$\xi := \frac{1}{2}|f(x) - y|. \tag{5.5}$$

Chernoff Bound

A famous inequality due to Chernoff [107] characterizes how the empirical mean $\frac{1}{m}\sum_{i=1}^m \xi_i$ converges to the expected value (or expectation) of ξ, denoted by $\mathbf{E}(\xi)$:

$$\mathrm{P}\left\{\left|\frac{1}{m}\sum_{i=1}^m \xi_i - \mathbf{E}(\xi)\right| \geq \epsilon\right\} \leq 2\exp(-2m\epsilon^2) \tag{5.6}$$

Note that the P refers to the probability of getting a sample $\xi_1, \ldots, \xi_m$ with the property $\left|\frac{1}{m}\sum_{i=1}^m \xi_i - \mathbf{E}(\xi)\right| \geq \epsilon$. Mathematically speaking, P strictly refers to a so-called *product* measure (cf. (B.11)). We will presently avoid further mathematical detail; more information can be found in Appendix B.

In some instances, we will use a more general bound, due to Hoeffding (Theorem 5.1). Presently, we formulate and prove a special case of the Hoeffding bound, which implies (5.6). Note that in the following statement, the ξ_i are no longer restricted to take values in $\{0, 1\}$.

Theorem 5.1 (Hoeffding [244]) *Let ξ_i, $i \in [m]$ be m independent instances of a bounded random variable ξ, with values in $[a, b]$. Denote their average by $Q_m = \frac{1}{m}\sum_i \xi_i$. Then for any $\epsilon > 0$,*

Hoeffding Bound

$$\left.\begin{array}{l} \mathrm{P}\{Q_m - \mathbf{E}(\xi) \geq \epsilon\} \\ \mathrm{P}\{\mathbf{E}(\xi) - Q_m \geq \epsilon\} \end{array}\right\} \leq \exp\left(-\frac{2m\epsilon^2}{(b-a)^2}\right). \tag{5.7}$$

The proof is carried out by using a technique commonly known as Chernoff's bounding method [107]. The proof technique is widely applicable, and generates bounds such as Bernstein's inequality [44] (exponential bounds based on the variance of random variables), as well as concentration-of-measure inequalities (see, e.g., [356, 66]). Readers not interested in the technical details underlying laws of large numbers may want to skip the following discussion.

We start with an auxiliary inequality.

Lemma 5.2 (Markov's Inequality (e.g., [136])) *Denote by ξ a nonnegative random variable with distribution* P. *Then for all $\lambda > 0$, the following inequality holds:*

$$\mathrm{P}\{\xi \geq \lambda \mathbf{E}(\xi)\} \leq \frac{1}{\lambda}. \tag{5.8}$$

Proof Using the definition of $\mathbf{E}(\xi)$, we have

$$\mathbf{E}(\xi) = \int_0^\infty \xi dP(\xi) \geq \int_{\lambda\mathbf{E}(\xi)}^\infty \xi dP(\xi) \geq \lambda\mathbf{E}(\xi)\int_{\lambda\mathbf{E}(\xi)}^\infty dP(\xi) = \lambda\mathbf{E}(\xi)\mathrm{P}\{\xi \geq \lambda\mathbf{E}(\xi)\}.$$

■

Proof of Theorem 5.1. Without loss of generality, we assume that $\mathbf{E}(\xi) = 0$ (otherwise simply define a random variable $\bar{\xi} := \xi - \mathbf{E}(\xi)$ and use the latter in the proof). Chernoff's bounding method consists in transforming a random variable ξ into $\exp(s\xi)$ ($s > 0$), and applying Markov's inequality to it. Depending on ξ, we can obtain different bounds. In our case, we use

$$\mathrm{P}\{\xi \geq \epsilon\} = \mathrm{P}\left\{\exp(s\xi) \geq \exp(s\epsilon)\right\} \leq e^{-s\epsilon}\mathbf{E}\left[\exp(s\xi)\right] \tag{5.9}$$

$$= e^{-s\epsilon}\mathbf{E}\left[\exp\left(\frac{s}{m}\sum_{i=1}^{m}\xi_i\right)\right] \leq e^{-s\epsilon}\prod_{i=1}^{m}\mathbf{E}\left[\exp\left(\frac{s}{m}\xi_i\right)\right]. \tag{5.10}$$

In (5.10), we exploited the fact that for positive random variables $\mathbf{E}\left[\prod_i \xi_i\right] \leq \prod_i \mathbf{E}\left[\xi_i\right]$. Since the inequality holds independent of the choice of s, we may minimize over s to obtain a bound that is as tight as possible. To this end, we transform the expectation over $\exp\left(\frac{s}{m}\xi_i\right)$ into something more amenable. The derivation is rather technical; thus we state without proof [244]: $\mathbf{E}\left[\exp(\frac{s}{m}\xi_i)\right] \leq \exp\left(\frac{s^2(b-a)^2}{8m^2}\right)$. From this, we conclude that the optimal value of s is given by $s = \frac{4m\epsilon}{(b-a)^2}$. Substituting this value into the right hand side of (5.10) proves the bound. ■

Let us now return to (5.6). Substituting (5.5) into (5.6), we have a bound which states how likely it is that for a given function f, the empirical risk is close to the actual risk,

$$\mathrm{P}\{|R_{\mathrm{emp}}[f] - R[f]| \geq \epsilon\} \leq 2\exp(-2m\epsilon^2). \tag{5.11}$$

Using Hoeffding's inequality, a similar bound can be given for the case of regression estimation, provided the loss $c(x, y, f(x))$ is bounded.

For any fixed function, the training error thus provides an unbiased estimate of the test error. Moreover, the convergence (in probability) $R_{\mathrm{emp}}[f] \to R[f]$ as $m \to \infty$ is exponentially fast in the number of training examples.[2] Although this sounds just about as good as we could possibly have hoped, there is one caveat: a crucial property of both the Chernoff and the Hoeffding bound is that they are probabilistic in nature. They state that the probability of a large deviation between test error and training error of f is small; the larger the sample size m, the smaller the probability. Granted, they do not rule out the presence of cases where the deviation is large, and our learning machine will have many functions that it can implement. Could there be a function for which things go wrong? It appears that

2. *Convergence in probability*, denoted as

$$|R_{\mathrm{emp}}[f] - R[f]| \xrightarrow{\mathrm{P}} 0 \text{ as } m \to \infty,$$

means that for all $\epsilon > 0$, we have

$$\lim_{m\to\infty} \mathrm{P}\{|R_{\mathrm{emp}}[f] - R[f]| > \epsilon\} = 0.$$

we would be very unlucky for this to occur *precisely* for the function f chosen by empirical risk minimization.

At first sight, it seems that empirical risk minimization should work — in contradiction to our lengthy explanation in the last section, arguing that we have to do more than that. What is the catch?

5.3 When Does Learning Work: the Question of Consistency

It turns out that in the last section, we were too sloppy. When we find a function f by choosing it to minimize the training error, we are no longer looking at independent Bernoulli trials. We are actually choosing f such that the mean of the ξ_i is as small as possible. In this sense, we are actively looking for the worst case, for a function which is very atypical, with respect to the average loss (i.e., the empirical risk) that it will produce.

We should thus state more clearly what it is that we actually need for empirical risk minimization to work. This is best expressed in terms of a notion that statisticians call *consistency*. It amounts to saying that as the number of examples m tends to infinity, we want the function f^m that minimizes $R_{\text{emp}}[f]$ (note that f^m need not be unique), to lead to a test error which converges to the lowest achievable value. In other words, f^m is asymptotically as good as whatever we could have done if we were able to directly minimize $R[f]$ (which we cannot, as we do not even know it). In addition, consistency requires that asymptotically, the training and the test error of f^m be identical.[3]

Consistency

It turns out that *without restricting the set of admissible functions*, empirical risk minimization is not consistent. The main insight of VC (Vapnik-Chervonenkis) theory is that actually, the *worst case* over all functions that the learning machine can implement determines the consistency of empirical risk minimization. In other words, we need a version of the law of large numbers which is *uniform* over all functions that the learning machine can implement.

5.4 Uniform Convergence and Consistency

The present section will explain how consistency can be characterized by a uniform convergence condition on the set of functions $\mathcal{F}$ that the learning machine can implement. Figure 5.2 gives a simplified depiction of the question of consistency. Both the empirical risk and the actual risk are drawn as functions of f. For

3. We refrain from giving a more formal definition of consistency, the reason being that there are some caveats to this classical definition of consistency; these would necessitate a discussion leading us away from the main thread of the argument. For the precise definition of the required notion of "nontrivial consistency," see [561].