Introduction: What Is Statistical Learning Theory?

Maxim Raginsky

August 26, 2014

1 A simple example: coin tossing

Let us start things off with a simple illustrative example. Suppose someone hands you a coin that has an unknown probability θ of coming up heads. You wish to determine this probability (coin bias) as accurately as possible by means of experimentation. Experimentation in this case amounts to repeatedly tossing the coin (this assumes, of course, that the bias of the coin on subsequent tosses does not change, but let's say you have no reason to believe otherwise). Let us denote the two possible outcomes of a single toss by 1 (for HEADS) and 0 (for TAILS). Thus, if you toss the coin n times, then you can record the outcomes as $X_1, \ldots, X_n$, where each $X_i \in \{0,1\}$ and $\mathbb{P}(X_i = 1) = \theta$ independently of all other X_j's. More succinctly, we can write our sequence of outcomes as $X^n \in \{0,1\}^n$, which is a *random* binary n-tuple. This is our *sample*.

What would be a reasonable estimate of θ? Well, by the Law of Large Numbers we know that, in a long sequence of independent coin tosses, the relative frequency of heads will eventually approach the true coin bias with high probability. So, without further ado you go ahead and estimate θ by

$$\widehat{\theta}_n(X^n) = \frac{1}{n}\sum_{i=1}^{n} X_i$$

(recall that each $X_i \in \{0,1\}$, so the sum in the above expression simply counts the number of times the coin came up HEADS). The notation $\widehat{\theta}_n(X^n)$ indicates the fact that the above estimate depends on the *sample size* n and on the entire sample X^n.

How accurate can this estimator be? To answer this question, let us fix an *accuracy parameter* $\varepsilon \in [0,1]$. Given θ and n, we can partition the entire set $\{0,1\}^n$ into two disjoint sets:

$$\mathsf{G}_{n,\varepsilon} \triangleq \left\{x^n \in \{0,1\}^n : |\widehat{\theta}_n(x^n) - \theta| \le \varepsilon\right\}$$
$$\mathsf{B}_{n,\varepsilon} \triangleq \left\{x^n \in \{0,1\}^n : |\widehat{\theta}_n(x^n) - \theta| > \varepsilon\right\}.$$

As the notation suggests, the n-tuples in $\mathsf{G}_{n,\varepsilon}$ are the "good ones:" if our random sequence of tosses X^n happens to land in $\mathsf{G}_{n,\varepsilon}$, then our estimate $\widehat{\theta}_n$ will differ from the true bias θ by at most ε in either direction. On the other hand, if X^n lands in $\mathsf{B}_{n,\varepsilon}$, then we will have no such luck. Of course, since we do not know θ, we have no way of telling whether X^n is in $\mathsf{G}_{n,\varepsilon}$ or in $\mathsf{B}_{n,\varepsilon}$. The best we can do is to compute the probability of a "bad sample" for each possible value of θ. This can be done using the so-called *Chernoff bound* [HR90]

$$\mathbb{P}_\theta^n(\mathsf{B}_{n,\varepsilon}) \equiv \mathbb{P}_\theta^n\left(|\widehat{\theta}_n(X^n) - \theta| > \varepsilon\right) \le 2e^{-2n\varepsilon^2} \tag{1}$$

(soon you will see where this comes from). Here, $\mathbb{P}_\theta^n$ denotes the distribution of the random sample X^n when the probability of heads on each toss is θ. Now, Eq. (1) says two things: (1) For any desired accuracy ε, probability of getting a bad sample decreases *exponentially* with sample size n. (2) In order to guarantee that the probability of a bad sample is at most δ, you will need

$$n \geq \frac{1}{2\varepsilon^2} \log\left(\frac{2}{\delta}\right)$$

coin tosses[1]. Thus, if you toss the coin at least this many times, then, no matter what θ is, you can assert with *confidence* at least $1-\delta$ that θ is somewhere between $\widehat{\theta}_n - \varepsilon$ and $\widehat{\theta}_n + \varepsilon$. This leads to the following

Observation 1. *For any true value θ of the coin bias,*

$$n(\varepsilon,\delta) \triangleq \left\lceil \frac{1}{2\varepsilon^2} \log\left(\frac{2}{\delta}\right) \right\rceil$$

tosses suffice to guarantee with confidence $1-\delta$ that the estimate $\widehat{\theta}_n$ has accuracy ε.

In view of this observation, we can call the function $(\varepsilon,\delta) \mapsto n(\varepsilon,\delta)$ the *sample complexity* of coin tossing.

This simple example illustrates the essence of statistical learning theory: We wish to learn something about a phenomenon of interest, and we do so by observing random samples of some quantity pertaining to the phenomenon. There are two basic questions we can ask:

1. How large of a sample do we need to achieve a given accuracy with a given confidence?
2. How efficient can our learning algorithm be?

Statistical learning theory [Vap98, Vid03] primarily concerns itself with the first of these questions, while the second question is within the purview of *computational learning theory* [Val84, KV94]. However, there are some overlaps between these two fields. In particular, we can immediately classify learning problems into "easy" and "hard" ones by looking at how their sample complexity grows as a function of $1/\varepsilon$ and $1/\delta$. In general, an easy problem is one whose sample complexity is *polynomial* in $1/\varepsilon$ and *polylogarithmic* in $1/\delta$ ("polylogarithmic" means polynomial in $\log(1/\delta)$). Of course, there are other factors that affect the sample complexity, and we will pay close attention to those as well.

2 From estimation to prediction

The coin tossing example of Section 1 was concerned with *estimation*. In fact, estimation was the focus of classical statistics, with early works dating back to Gauss, Laplace and the numerous members of the Bernoulli clan (the book by Stigler [Sti86] is an excellent survey of the history of statistics, full of amusing anecdotes and trivia, and much historical and scientific detail besides). By contrast, much of statistical learning theory (and much of modern statistics too) focuses on *prediction* (see the book by Clarke, Fokoué and Zhang [CFZ09] for a comprehensive exposition of the predictive view of statistical machine learning and data mining). In a typical prediction problem, we have two jointly distributed random variables[2] X and Y, where only X is available for observation, and we wish to devise a means of

[1] Unless stated otherwise, log will always denote natural logarithms (base e).

[2] Please consult Appendix A for basic definitions and notation pertaining to probability distributions and random variables.

predicting Y on the basis of this observation. Thus, a *predictor* is any well-behaved[3] function from X (the domain of X) into Y (the domain of Y). For example, in medical diagnosis, X might record the outcomes of a series of medical tests and other data for a single patient, while $Y \in \{0,1\}$ would correspond to the patient either having or not having a particular health issue.

The basic premise of statistical learning theory is that the details of the joint distribution P of X and Y are vague (or even completely unknown), and the only information we have to go on is a sequence of n independent observations $(X_1, Y_1), \ldots, (X_n, Y_n)$ drawn from P. Assuming we have a quantitative criterion by which to judge a predictor's accuracy, the same basic question presents itself: How large does the sample $\{(X_i, Y_i)\}_{i=1}^n$ have to be in order for us to be able to construct a predictor achieving a given level of accuracy and confidence?

Of course, not all learning problems involve prediction. For example, problems like clustering, density estimation, feature (or representation) learning do not. We will see later that the mathematical formalism of statistical learning theory is flexible enough to cover such problems as well. For now, though, let us focus on prediction to keep things concrete. To get a handle on the learning problem, let us first examine the ideal situation, in which the distribution P is known.

2.1 Binary classification

The simplest prediction problem is that of *binary classification* (also known as *pattern classification* or *pattern recognition*) [DGL96]. In a typical scenario, X is a subset of $\mathbb{R}^p$, the p-dimensional Euclidean space, and $\mathsf{Y} = \{0,1\}$. A predictor (or a *classifier*) is any mapping $f : \mathsf{X} \to \{0,1\}$. A standard way of evaluating the quality of binary classifiers is by looking at their probability of classification error. Thus, for a classifier f we define the *classification loss* (or *risk*)

$$L_P(f) \triangleq \mathbb{P}(f(X) \neq Y) \equiv \int_{\mathsf{X}\times\{0,1\}} \mathbf{1}_{\{f(x)\neq y\}} P(\mathrm{d}x, dy),$$

where $\mathbf{1}_{\{\cdot\}}$ is the *indicator function* taking the value 1 if the statement in the braces is true, and 0 otherwise. What is the best classifier for a given P? The answer is given by the following

Proposition 1. *Given the joint distribution P on $\mathsf{X} \times \{0,1\}$, let $\eta(x) \triangleq \mathbb{E}[Y|X = x] \equiv \mathbb{P}(Y = 1|X = x)$. Then the classifier*

$$f_P^*(x) \triangleq \begin{cases} 1, & \text{if } \eta(x) \geq 1/2 \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

minimizes the probability of classification error over all $f : \mathsf{X} \to \{0,1\}$, i.e.,

$$L_P(f_P^*) = \min_{f:\mathsf{X}\to\{0,1\}} L_P(f).$$

Remark 1. Some terminology: The function η defined above is called the *regression function*, the classifier in (2) is called the *Bayes classifier*, and its risk

$$L_P^* \triangleq L_P(f_P^*)$$

is called the *Bayes rate*.

[3] In fancy language, "well-behaved" will typically mean "measurable" with respect to appropriate σ-fields defined on X and Y. We will ignore measurability issues in this course.

Proof. Consider an arbitrary classifier $f : \mathsf{X} \to \{0,1\}$. Then

$$\begin{aligned} L_P(f) &= \int_{\mathsf{X}\times\{0,1\}} \mathbf{1}_{\{f(x)\neq y\}} P(\mathrm{d}x, \mathrm{d}y) \\ &= \int_{\mathsf{X}} P_X(\mathrm{d}x) \left\{ P_{Y|X}(1|x)\mathbf{1}_{\{f(x)\neq 1\}} + P_{Y|X}(0|x)\mathbf{1}_{\{f(x)\neq 0\}} \right\} \\ &= \int_{\mathsf{X}} P_X(\mathrm{d}x) \underbrace{\left\{ \eta(x)\mathbf{1}_{\{f(x)\neq 1\}} + (1-\eta(x))\mathbf{1}_{\{f(x)\neq 0\}} \right\}}_{\triangleq \ell(f,x)}, \end{aligned} \tag{3}$$

where we have used Eq. (2.1), the factorization $P = P_X \times P_{Y|X}$, and the definition of η. From the above, it is easy to see that, in order to minimize $L_P(f)$, it suffices to minimize the term $\ell(f,x)$ in (3) separately for each value of $x \in \mathsf{X}$. If we let $f(x) = 1$, then $\ell(f,x) = 1-\eta(x)$, while for $f(x) = 0$ we will have $\ell(f,x) = \eta(x)$. Clearly, we should set $f(x)$ to 1 or 0, depending on whether $1-\eta(x) \leq \eta(x)$ or not. This yields the rule in (2). □

2.2 Minimum mean squared error prediction

Another prototypical example of a prediction problem is *minimum mean squared error (MMSE) prediction* [CZ07], where $\mathsf{X} \subseteq \mathbb{R}^p$, $\mathsf{Y} \subseteq \mathbb{R}$, and the admissible predictors are functions $f : \mathsf{X} \to \mathbb{R}$. The quality of such a predictor f is measured by the *MSE*

$$L_P(f) \triangleq \mathbb{E}(f(X)-Y)^2 \equiv \int_{\mathsf{X}\times\mathsf{Y}} (f(x)-y)^2 P(\mathrm{d}x, \mathrm{d}y).$$

The MMSE predictor is characterized by the following

Proposition 2. *Given the joint distribution P on $\mathsf{X}\times\mathsf{Y}$ with $\mathsf{X} \subseteq \mathbb{R}^p$ and $\mathsf{Y} \subseteq \mathbb{R}$, the regression function $f_P^*(x) \triangleq \mathbb{E}[Y|X=x]$ is the MMSE predictor. Moreover, for any other predictor f we have*

$$L_P(f) = \|f - f_P^*\|^2_{L^2(P_X)} + L_P^*,$$

where for any function $g : \mathsf{X} \to \mathbb{R}$

$$\|g\|^2_{L^2(P_X)} \triangleq \int_{\mathsf{X}} |g(x)|^2 P_X(\mathrm{d}x) \equiv \mathbb{E}|g(X)|^2$$

is the squared L^2 norm with respect to the marginal distribution P_X, and $L_P^* \triangleq L_P(f_P^*)$.

Proof. Consider an arbitrary predictor $f : \mathsf{X} \to \mathbb{R}$. Then

$$\begin{aligned} L_P(f) &= \mathbb{E}(f(X)-Y)^2 \\ &= \mathbb{E}(f(X) - f_P^*(X) + f_P^*(X) - Y)^2 \\ &= \mathbb{E}(f(X) - f_P^*(X))^2 + 2\mathbb{E}[(f(X)-f_P^*(X))(f_P^*(X)-Y)] + \mathbb{E}(f_P^*(X)-Y)^2 \\ &= \|f - f_P^*\|^2_{L^2(P_X)} + 2\mathbb{E}[(f(X)-f_P^*(X))(f_P^*(X)-Y)] + L_P^*. \end{aligned}$$

Let us analyze the second (cross) term. Using the law of iterated expectation, we have

$$\begin{aligned} \mathbb{E}[(f(X)-f_P^*(X))(f_P^*(X)-Y)] &= \mathbb{E}\big[\mathbb{E}[(f(X)-f_P^*(X))(f_P^*(X)-Y)|X]\big] \\ &= \mathbb{E}\big[(f(X)-f_P^*(X))\mathbb{E}[(f_P^*(X)-Y)|X]\big] \\ &= \mathbb{E}\big[(f(X)-f_P^*(X))(f_P^*(X)-\mathbb{E}[Y|X])\big] \\ &= 0, \end{aligned}$$

where in the last step we used the definition $f_P^*(x) \triangleq \mathbb{E}[Y|X=x]$. Thus,

$$L_P(f) = \|f - f_P^*\|^2_{L^2(P_X)} + L_P^* \geq L_P^*,$$

where equality holds if and only if $f = f_X^*$ (with P_X-probability one). □

2.3 A general prediction problem

In the general case, X and Y are arbitrary sets, admissible predictors are functions $f : \mathsf{X} \to \mathsf{Y}$ (or, more generally, $f : \mathsf{X} \to \mathsf{U}$ for some suitable *prediction space* U), and the quality of a predictor f on a pair $(x, y) \in \mathsf{X} \times \mathsf{Y}$ is judged in terms of some fixed *loss function* $\ell : \mathsf{U} \times \mathsf{Y} \to \mathbb{R}$ by $\ell(f(x), y)$, the loss incurred in predicting the true y by $\widehat{u} = f(x)$. The expected risk of f is then

$$L_P(f) \triangleq \mathbb{E}[\ell(f(X), Y)] \equiv \int_{\mathsf{X}\times\mathsf{Y}} \ell(f(x), y) P(\mathrm{d}x, \mathrm{d}y).$$

This set-up covers the two previous examples:

1. If $\mathsf{X} \subseteq \mathbb{R}^p$, $\mathsf{Y} = \mathsf{U} = \{0, 1\}$, and $\ell(u, y) \triangleq \mathbf{1}_{\{u \neq y\}}$, then we recover the binary classification problem.
2. If $\mathsf{X} \subseteq \mathbb{R}^p$, $\mathsf{Y} \subseteq \mathbb{R} = \mathsf{U}$, and $\ell(u, y) \triangleq (u - y)^2$, then we recover the MMSE prediction problem.

Given P and ℓ, we define the minimum risk

$$L_P^* \triangleq \inf_{f:\mathsf{X}\to\mathsf{U}} \mathbb{E}[\ell(f(X), Y)], \tag{4}$$

where we use inf instead of min since there may not be a minimizing f (when that happens, one typically picks some small $\varepsilon > 0$ and seeks *ε-minimizers*, i.e., any $f_\varepsilon^* : \mathsf{X} \to \mathsf{U}$, such that

$$L_P(f_\varepsilon^*) \leq L_P(f) + \varepsilon \tag{5}$$

for all $f : \mathsf{X} \to \mathsf{U}$). We will just assume that a minimizer exists, but continue to use inf to keep things general.

Thus, an abstract prediction problem is characterized by three objects: a probability distribution P of $(X, Y) \in \mathsf{X} \times \mathsf{Y}$, a class of admissible predictors $f : \mathsf{X} \to \mathsf{U}$, and a loss function $\ell : \mathsf{U} \times \mathsf{Y} \to \mathbb{R}$. The solution to the prediction problem is any f_P^* that attains the infimum in (4) (or comes ε-close as in (5)). Once such a f_P^* is computed, we can use it to predict the *output* $Y \in \mathsf{Y}$ for any given *input* $X \in \mathsf{X}$ by $\widehat{Y} = f_P^*(X)$, where the interpretation is that the random couple $(X, Y) \sim P$ pertains to the phenomenon of interest, X corresponds to its observable aspects, and Y corresponds to some unobservable characteristic that we may want to ascertain.

3 Goals of learning

We will close our introduction to statistical learning theory by a rough sketch of the "goals of learning" in a random environment. Please keep in mind that this is not meant to be a definitive treatment, which will come later in the course.

So far we have discussed the "ideal" case when the distribution P of (X, Y) is known. Statistical learning theory deals with the setting where our knowledge of P is only partial (or nonexistent), but we have access to a *training sample* $(X_1, Y_1), \ldots, (X_n, Y_n)$ of independent draws from P. Formally, we say that

the pairs $(X_i, Y_i), 1 \le i \le n$, are *independent and identically distributed (i.i.d.)* according to P, and we often write this as

$$(X_i, Y_i) \overset{\text{i.i.d.}}{\sim} P, \qquad i = 1, \ldots, n.$$

To keep the notation simple, let us denote by Z the product space $\mathsf{X} \times \mathsf{Y}$ and let $Z_i = (X_i, Y_i)$ for each i. Our training sample is then $Z^n = (Z_1, \ldots, Z_n) \in \mathsf{Z}^n$. Roughly speaking, the goal of learning is to take Z^n as an input and to produce a *candidate predictor* $\widehat{f}_n : \mathsf{X} \to \mathsf{U}$ as an output. Note that since Z^n is a random variable, so is $\widehat{f}_n$. A *learning algorithm* (or a *learner*) is a procedure that can do this for any sample size n. Thus, a learning algorithm is a box for converting training samples into predictors.

Let's suppose that we have some learning algorithm to play with. Given a sample Z^n of size n, it outputs a candidate predictor $\widehat{f}_n$. How good is this predictor? Well, let's suppose that someone (say, Nature) hands us a fresh independent sample $Z = (X, Y)$ from the same distribution P that has generated the training sample Z^n. Then we can test $\widehat{f}_n$ by applying it to X and seeing how close $\widehat{U} = \widehat{f}_n(X)$ is to Y by computing the *instantaneous loss* $\ell(\widehat{U}, Y) \equiv \ell(\widehat{f}_n(X).Y)$. The *expectation* of the instantaneous loss w.r.t. the (unknown) distribution P,

$$L_P(\widehat{f}_n) \equiv \int_{\mathsf{X}\times\mathsf{Y}} \ell(\widehat{f}_n(x), y) P(\mathrm{d}x, \mathrm{d}y), \tag{6}$$

is called the *generalization error* of the learner at sample size n. It is crucial to note that $L_P(\widehat{f}_n)$ is a *random variable*, since $\widehat{f}_n$ is a function of the random sample Z^n. In fact, to be more precise, we should write the generalization error as the conditional expectation $\mathbb{E}[\ell(\widehat{f}_n(X), Y)|Z^n]$, but since $Z = (X, Y)$ is assumed to be independent from Z^n, we get (6).

Now, we will say that our learner has done a good job when its generalization error is suitably small. But how small can it be? To answer this question (or at least to point towards a possible answer), we must first agree that learning without any initial assumptions is a futile task. For example, consider fitting a curve to a training sample $(X_1, Y_1), \ldots, (X_n, Y_n)$, where both the X_i's and the Y_i's are real numbers. A simple-minded approach would be to pick any curve that precisely agreed with the entire sample – in other words, to select some $\widehat{f}_n$, such that $\widehat{f}_n(X_i) = Y_i$ for all $i = 1, \ldots, n$. But there is an uncountable infinity of such functions! Which one should we choose? The answer is, of course, there is no way to know, if only because we have no clue about P! We could pick a very smooth function, but it could very well happen that the optimal f_P^* tends to be smooth for some values of the input and rough for some others. Alternatively, we could choose a very wiggly and complicated curve, but then it might just be the case that f_P^* is really simple.

A way out of this dilemma is to introduce what is known in the artificial intelligence community as an *inductive bias*. We go about it by restricting the space of candidate predictors our learner is allowed to search over to some suitable family $\mathscr{H}$, which is typically called the *hypothesis space*. Thus, we stipulate that $\widehat{f}_n \in \mathscr{H}$ for any sample Z^n. Given P, let us define the minimum risk over $\mathscr{H}$:

$$L_P^*(\mathscr{H}) \triangleq \inf_{f \in \mathscr{H}} L_P(f). \tag{7}$$

Clearly, $L^* P(\mathscr{H}) \ge L_P^*$, since the latter involves minimization over a larger set. However, now, provided the hypothesis space $\mathscr{H}$ is "manageable," we may actually hope to construct a learner that would guarantee that

$$L_P(\widehat{f}_n) \approx L_P^*(\mathscr{H}) \qquad \text{with high probability.} \tag{8}$$

Then, if we happen to be so lucky that f_P^* is actually in $\mathscr{H}$, we will have attained the Holy Grail, but even if we are not so lucky, we may still be doing pretty well. To get a rough idea of what is involved, let us look at the *excess risk* of $\widehat{f}_n$ relative to the best predictor f_P^*:

$$E_P(\widehat{f}_n) \triangleq L_P(\widehat{f}_n) - L_P^* = \underbrace{L_P(\widehat{f}_n) - L_P^*(\mathscr{H})}_{E_{\text{est}}} + \underbrace{L_P^*(\mathscr{H}) - L_P^*}_{E_{\text{approx}}}. \tag{9}$$

If the learner is good in the sense of (8), then we will have

$$E_P(\widehat{f}_n) \approx L_P^*(\mathscr{H}) - L_P^* \qquad \text{with high probability,}$$

which, in some sense, is the next best thing to the Holy Grail, especially if we can choose $\mathscr{H}$ so well that we can guarantee that the difference $L_P^*(\mathscr{H}) - L_P^*$ is small for any possible choice of P.

Note the decomposition of the excess risk into two terms, denoted in (9) by E_{est} and E_{approx}. The first term, E_{est}, depends on the learned predictor $\widehat{f}_n$, as well as on the hypothesis class $\mathscr{H}$, and is referred to as the *estimation error* of the learner. The second term, E_{approx}, depends only on $\mathscr{H}$ and on P, and is referred to as the *approximation error* of the hypothesis space. Most of the effort in statistical learning theory goes into analyzing and bounding the estimation error for various choices of $\mathscr{H}$. Analysis of E_{approx} is the natural domain of approximation theory. The overall performance of a given learning algorithm depends on the interplay between these two sources of error. The text by Cucker and Zhou [CZ07] does a wonderful job of treating both the estimation and the approximation aspects of learning algorithms.

3.1 Beyond prediction

As we had briefly pointed out earlier, not all learning problems involve prediction. Luckily, the mathematical formalism we have just introduced can be easily adapted to a more general view of learning. Consider a random object Z taking values in some space Z according to an unknown distribution P. Suppose that there is a very large class $\mathscr{F}$ of functions $f : \mathsf{Z} \to \mathbb{R}$, and for each $f \in \mathscr{F}$ we can define its expected loss (or risk)

$$L_P(f) \triangleq \mathbb{E}[f(Z)] = \int_{\mathsf{Z}} f(z) P(\mathrm{d}z). \tag{10}$$

Suppose also that $\mathscr{F}$ has the property that there exists at least one $f_P^* \in \mathscr{F}$ that achieves

$$L_P(f_P^*) = \inf_{f \in \mathscr{F}} L_P(f). \tag{11}$$

The class $\mathscr{F}$ may even depend on P. Let's see how we can describe some unsupervised learning problems in this way:

- **Density estimation.** Suppose that $\mathsf{Z} \subseteq \mathbb{R}^d$ for some d, and that P has a probability density function (pdf) p. We can construct a suitable class $\mathscr{F}$ as follows: pick a nonnegative function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}^+$, and let $\mathscr{F} = \mathscr{F}_{P,\ell}$ consist of all functions of the form

  $$f_q(z) = \ell\big(p(z), q(z)\big), \tag{12}$$

  as q ranges over a suitable class of pdf's q on $\mathbb{R}^d$. Then

  $$L_P(f_q) = \mathbb{E}_P\left[\ell\big(p(Z), q(Z)\big)\right] = \int_{\mathbb{R}^d} p(z)\ell\big(p(z), q(z)\big)\mathrm{d}z. \tag{13}$$

This is fairly general. For example, if we assume that $p > 0$ on Z, then we can let $\ell(u,u') = \left|\frac{u'}{u} - 1\right|^2$, in which case we recover the well-known L^2 criterion:

$$L_P(f_q) = \int_{\mathsf{Z}} p(z) \left| \frac{q(z)}{p(z)} - 1 \right|^2 \mathrm{d}z = \|p - q\|_{L^2}^2.$$

Or we can let $\ell(u,u') = \log(u/u')$, which gives us the *relative entropy* (also known as the *Kullback–Leibler divergence*):

$$L_P(f_q) = \int_{\mathsf{Z}} p(z) \log \frac{p(z)}{q(z)} \mathrm{d}z = D(p\|q).$$

- **Clustering.** In a basic form of the clustering problem, we seek a partition of the domain Z of interest into a fixed number, say k, of disjoint clusters $\mathsf{C}_1, \ldots, \mathsf{C}_k$, such that all points z that belong to the same cluster are somehow "similar." For example, we may define a distance function $d : \mathsf{Z} \times \mathsf{Z} \to \mathbb{R}^+$ and represent each cluster C_j, $1 \le j \le k$, by a single "representative" $v_j \in \mathsf{Z}$. A clustering $\mathbf{C}$ is then described by k pairs $\{(\mathsf{C}_j, v_j)\}_{j=1}^k$, where $\mathsf{Z} = \bigcup_{j=1}^k \mathsf{C}_j$. Consider the class $\mathscr{F} = \mathscr{F}_k$ of all functions of the form

$$f_{\mathbf{C}}(z) = \sum_{j=1}^k \mathbf{1}_{\{z \in \mathsf{C}_j\}} d(z, v_j)$$

  as $\mathbf{C}$ runs over all clusterings $\{(\mathsf{C}_j, v_j)\}_{j=1}^k$. We can then evaluate the quality of our clustering $\mathbf{C}$ by looking at the expectation

$$L_P(f_{\mathbf{C}}) = \mathbb{E}_P \left[\sum_{j=1}^k \mathbf{1}_{\{Z \in \mathsf{C}_j\}} d(Z, v_j) \right]$$

- **Feature learning.** Broadly speaking, feature learning refers to constructing a representation of the original input Z that could be fed to a supervised learning algorithm further down the line. There could be multiple reasons for wanting to do this, ranging from computational considerations to a desire to capture "salient" characteristics of the data that could be relevant for prediction, while "factoring out" the irrelevant parts. Mathematically, a feature is a mapping $\varphi : \mathsf{Z} \to \tilde{\mathsf{Z}}$ into some other representation space $\tilde{\mathsf{Z}}$, so that each point $z \in \mathsf{Z}$ is represented $\tilde{z} = \varphi(z)$, and it is this representation that will be used by another learning algorithm down the line. (Ideally, good feature representations should be agnostic with respect to the nature of the learning problem where they will be used.) One way to score the quality of a feature is to consider a loss function of the form $\ell : \mathsf{Z} \times \tilde{\mathsf{Z}} \to \mathbb{R}^+$, so that $\ell(z, \tilde{z})$ is small if z is well-represented by $\tilde{z}$. Then, for a fixed collection Φ of candidate feature maps, we could consider a class $\mathscr{F} = \mathscr{F}_{\Phi,\ell}$ of functions of the form

$$f_\varphi(z) = \ell\big(z, \varphi(z)\big), \qquad \varphi \in \Phi.$$

  This is a very wide umbrella that can cover a wide variety of unsupervised learning tasks (e.g., clustering).

With this framework in place, we can proceed to talk about learning: We fix an appropriate hypothesis space $\mathscr{H}$ (which, unlike $\mathscr{F}$, cannot depend on P, since we don't know it); in that case, $L_P^*(\mathscr{H})$ is defined exactly as before. And, just as before, a learning algorithm is a rule for mapping an i.i.d. sample $Z^n = (Z_1, \ldots, Z_n)$ from P to an element $\widehat{f}_n \in \mathscr{H}$. The objective is also the same as before: ensure that

$$L_P(\widehat{f}_n) \approx L_P^*(\mathscr{H}) \qquad \text{with high probability.}$$

Thus, we can treat supervised learning and unsupervised learning on the same footing.

A Probability and random variables

Probability theory is the foundation of statistical learning theory. In this appendix, I will give a quick overview of the main concepts and set up the notation that will be used consistently throughout the course. This is by no means intended as a substitute for a serious course in probability; as a good introductory reference, I recommend the text by Gray and Davisson [GD04], which is geared towards beginning graduate students in electrical engineering.

Let Ω be a set. A collection $\mathscr{F}$ of subsets of Ω is called a *σ-algebra* if it has the following two properties:

1. If $A \in \Omega$, then $A^c \equiv \Omega \setminus A \in \mathscr{F}$
2. For any sets $A_1, A_2 \ldots \in \mathscr{F}$, their union belongs to $\mathscr{F}$: $\bigcup_{i=1}^{\infty} A_i \in \mathscr{F}$.

In other words, any σ-algebra is closed under complements and countable unions. This implies, in particular, that the empty set $\emptyset$ and the entire set Ω are contained in any σ-algebra $\mathscr{F}$, and that such an $\mathscr{F}$ is closed under countable intersections. A pair $(\Omega, \mathscr{F})$ consisting of a set and a σ-algebra is called a *measurable space*. A *probability measure* on $(\Omega, \mathscr{F})$ is a function $\mathbb{P} : \mathscr{F} \to [0,1]$, such that

1. $\mathbb{P}(\Omega) = 1$
2. Given any countably infinite sequence $A_1, A_2, \ldots \mathscr{F}$ of pairwise disjoint sets, i.e., $A_i \cap A_j = \emptyset$ for every pair $i \neq j$,
$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} A_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(A_i).$$

The triple $(\Omega, \mathscr{F}, \mathbb{P})$ is called a *probability space*.

Let $(\mathsf{X}, \mathscr{B})$ be some other measurable space. A *random variable* on Ω with values in X is any function $X : \Omega \to \mathsf{X}$ with the property that, for any $B \in \mathscr{B}$, the set

$$X^{-1}(B) \triangleq \{\omega \in \Omega : X(\omega) \in B\}$$

lies in $\mathscr{F}$ (in fancy language, X is a *measurable* mapping from $(\Omega, \mathscr{F})$ into $(\mathsf{X}, \mathscr{B})$). Together, X and $\mathbb{P}$ induce a probability measure P_X on $(\mathsf{X}, \mathscr{B})$ by setting

$$P_X(B) \triangleq \mathbb{P}\left(X^{-1}(B)\right) \equiv \mathbb{P}\left(\{\omega \in \Omega : X(\omega) \in B\}\right),$$

which is called the *distribution* of X. (A good character-building exercise is to try and prove that P_X is indeed a valid probability measure.) Once P_X is defined, we can forget all about $(\Omega, \mathscr{F}, \mathbb{P})$ and just work with P_X. Here are two standard examples to keep in mind. One is when X is a finite set, in which case we can take the σ-algebra consisting of all subsets of X, let P_X be the probability mass function (pmf) of X, and then for any $B \subseteq \mathsf{X}$ we will have

$$P_X(B) = \sum_{x \in B} P_X(x). \tag{14}$$

The other is when X is the real line $\mathbb{R}$ (with the so-called Borel σ-algebra, which at this point you don't have to worry about), and P_X has a probability density function (pdf) p_X, giving

$$P_X(B) = \int_B p_X(x)\mathrm{d}x. \tag{15}$$

for any (measurable) set $B \subseteq \mathbb{R}$. We will use a more abstract notation that covers these two cases (and much more besides):

$$P_X(B) = \int_B P_X(\mathrm{d}x), \qquad \forall B \in \mathscr{B}. \tag{16}$$

When seeing something like (16), just think of (14) or (15).

If $f : \mathsf{X} \to \mathbb{R}$ is a real-valued function on X, the *expected value* of $f(X)$ is

$$\mathbb{E}[f(X)] = \int_{\mathsf{X}} f(x) P_X(\mathrm{d}x);$$

again, think of either

$$\mathbb{E}[f(X)] = \sum_{x \in \mathsf{X}} f(x) P_X(x)$$

in the case of discrete X, or

$$\mathbb{E}[f(X)] = \int_{\mathbb{R}} f(x) p_X(x) \mathrm{d}x$$

in the case of $\mathsf{X} = \mathbb{R}$ and a random variable with a pdf p_X.

For any two jointly distributed random variables $X \in \mathsf{X}$ and $Y \in \mathsf{X}$, we have their joint distribution P_{XY}, the marginals

$$P_X(A) \triangleq P_{XY}(A \times \mathsf{Y}) \equiv \int_{A \times \mathsf{Y}} P_{XY}(\mathrm{d}x, \mathrm{d}y)$$
$$P_Y(B) \triangleq P_{XY}(\mathsf{X} \times B) \equiv \int_{\mathsf{X} \times B} P_{XY}(\mathrm{d}x, \mathrm{d}y)$$

for all measurable sets $A \subseteq \mathsf{X}, B \subseteq \mathsf{Y}$, and the conditional distribution

$$P_{Y|X}(B|A) \triangleq \frac{P_{XY}(A \times B)}{P_X(A)}$$

of Y given that $X \in A$. Neglecting technicalities and considerations of rigor, we can define the conditional distribution of Y given $X = x$, denoted by $P_Y|X(\cdot|x)$, implicitly through

$$P_{XY}(A \times B) = \int_A P_X(\mathrm{d}x) \left(\int_B P_{Y|X}(\mathrm{d}y|x) \right).$$

Here, it is helpful to think of the conditional pmf

$$P_{Y|X}(y|x) = \frac{P_{XY}(x, y)}{P_X(x)}$$

in the discrete case, and of the conditional pdf

$$p_{Y|X}(y|x) = \frac{p_{XY}(x, y)}{p_X(x)}$$

in the continuous case. The *conditional expectation* of any function $f : \mathsf{X} \times \mathsf{Y} \to \mathbb{R}$ given X, denoted by $\mathbb{E}[f(X, Y)|X]$, is a random variable $g(X)$ that takes values in $\mathbb{R}$, such that[4]

$$\mathbb{E}[f(X, Y)] = \mathbb{E}[g(X)] \equiv \mathbb{E}[\mathbb{E}[f(X, Y)|X]].$$

[4] As usual, I am being rather cavalier with the definitions here, since the choice of g is not unique; one typically speaks of different *versions* of the conditional expectation, which, properly speaking, should be defined w.r.t. the σ-algebra generated by X. Again, this will not be an issue in this course.

This is known as the *law of iterated expectations*. Once again, think of

$$\mathbb{E}[f(X,Y)|X=x] = \sum_{y\in\mathsf{Y}} f(x,y)P_{Y|X}(y|x)$$

if both X and Y are discrete sets, and of

$$\mathbb{E}[f(X,Y)|X=x] = \int_{\mathsf{Y}} f(x,y)p_{Y|X}(y|x)\mathrm{d}y$$

if both X and Y are subsets of $\mathbb{R}$.

References

[CFZ09] B. Clarke, E. Fokoué, and H. H. Zhang. *Principles and Theory for Data Mining and Machine Learning*. Springer, 2009.

[CZ07] F. Cucker and D. X. Zhou. *Learning Theory: An Approximation Theory Viewpoint*. Cambridge University Press, 2007.

[DGL96] L. Devroye, L. Györfi, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[GD04] R. M. Gray and L. D. Davisson. *An Introduction to Statistical Signal Processing*. Cambridge University Press, 2004.

[HR90] T. Hagerup and C. Rüb. A guided tour of Chernoff bounds. *Information Processing Letters*, 33(6):305–308, 1990.

[KV94] M. J. Kearns and U. V. Vazirani. *An Introduction to Computational Learning Theory*. MIT Press, 1994.

[Sti86] S. M. Stigler. *The History of Statistics: The Measurement of Uncertainty Before 1900*. Harvard University Press, 1986.

[Val84] L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[Vid03] M. Vidyasagar. *Learning and Generalization*. Springer, 2 edition, 2003.