Week 7: Learning Theory

Instructor: Sergey Levine

1 A Generalized View of Learning

In this unit, we'll take a deeper look at issues such as bias and variance, and gain a deeper understanding of when and why our machine learning algorithms might perform better or worse. In contrast to previous units, we will not introduce a new machine learning method, but rather construct a generalized view of various machine learning methods that allows us to analyze all of them together.

As I've mentioned previously, a machine learning problem consists of data, a hypothesis space, an objective, and an algorithm. We'll let $\mathcal{D}$ denote our data, which as usual will consist of attributes $\mathbf{x}$ and labels or response variables y, which may be real-valued or categorical. Our hypothesis space will consist of functions $f(\mathbf{x})$ that make predictions (either labels or probabilities), and we'll assume that we have a loss function $L(y, f(\mathbf{x}))$ that evaluates the quality of a prediction. For consistency, we'll assume that lower losses are better. For example, in linear regression, we might have

$$L(y, f(\mathbf{x})) = \frac{1}{2}(f(\mathbf{x}) - y)^2,$$

while in maximum likelihood classification, we might have

$$L(y, f(\mathbf{x})) = -\log f(\mathbf{x})_y,$$

where we assume that $f(\mathbf{x})$ outputs a vector with L_y probabilities, such that $f(\mathbf{x})_y$ is the predicted value of $p(y)$. Note that we negate the likelihood, since the loss will be minimized. We could also write a "0-1" loss

$$L(y, f(\mathbf{x})) = \delta(f(\mathbf{x}) \neq y),$$

or the exponentiated loss for boosting, or the hinge loss for SVMs.

The last part we have to figure out is the algorithm. In order to analyze machine learning algorithms at this high level of generality, we'll simply assume that the algorithm minimizes the expected loss function on the training set. That is, our algorithm will do something that looks like

$$\hat{f} \leftarrow \arg\min_f \frac{1}{N} \sum_{i=1}^{N} L(y^i, f(\mathbf{x}^i)).$$

In practice, many of the more complex hypothesis classes, such as decision trees or neural networks, only have approximate optimization algorithms: in the case of decision trees, we use a greedy heuristic, while in the case of neural networks, we optimize the loss, but cannot guarantee that we will find a global optimum.

At a high level, machine learning is a funny game where we minimize the loss of a training set, but we actually want to minimize the loss on all the data, including data that is not in the training set. Specifically, we care about the generalization error

$$\mathcal{E}_{\text{gen},\mathcal{D}} = \int p(\mathbf{x}, y) L(y, \hat{f}(\mathbf{x})) d\mathbf{x} dy = E[L(y, \hat{f}(\mathbf{x}))].$$

Since $\hat{f}$ is a deterministic consequence of the training set $\mathcal{D}$, we can also write this expectation as a conditional expectation, to express the fact that, once we chosen a hypothesis space and algorithm, only the choice of the training set (which is typically not up to us) determines our generalization error:

$$\mathcal{E}_{\text{gen},\mathcal{D}} = E[L(y, \hat{f}(\mathbf{x}))|\mathcal{D}].$$

An interesting quantity to analyze in this case is the expected generalization error:

$$\mathcal{E}_{\text{gen},N} = \int p(\mathcal{D}|N) \mathcal{E}_{\text{gen},\mathcal{D}} d\mathcal{D} = E[L(y, \hat{f}(\mathbf{x}))|N].$$

This is the generalization error we expect to see on all the data in the world, averaged over the possible training sets of size N that we can expect to get. Remember that we make the i.i.d. assumption, which means that we expect the dataset of size N to be sampled (independently for each datapoint) from the same distribution as the one we have at test time: $p(\mathbf{x}, y)$. This expected generalization error is a good metric to analyze if we want to understand the behavior of our hypothesis class, independently of the particular choice of training set. Another useful value to analyze is the expected error at a particular point:

$$\mathcal{E}_{\text{gen},N}(\mathbf{x}) = \int p(\mathcal{D}|N) p(y|\mathbf{x}) L(y, \hat{f}(\mathbf{x})) dy d\mathcal{D} = E[L(y, \hat{f}(\mathbf{x}))|\mathbf{x}, N].$$

This is the error we expect to see for a particular point $\mathbf{x}$ in expectation when we train $\hat{f}(\mathbf{x})$ on a dataset of size N.

2 Revisiting Bias & Variance

First, let's think about a regression model (like linear regression), where $f(\mathbf{x})$ predicts $y \in \mathbb{R}$ and we use the loss $L(y, \hat{f}(\mathbf{x})) = (y - \hat{f}(\mathbf{x}))^2$ (where we drop the factor $\frac{1}{2}$ for now). Let's assume that the real data is produced according to some (unknown) function $g(\mathbf{x})$ such that

$$y = g(\mathbf{x}) + \epsilon,$$

where $\epsilon \sim \mathcal{N}(0, \sigma_\epsilon^2)$. This means that the real (physical) process that gives rise to our data is the deterministic function $g(\mathbf{x})$ and some Gaussian noise with variance σ_ϵ^2. This is a very simple model of the randomness in our data, but it is useful in practice and allows us to analyze the error of our regressor. Let's consider $\mathcal{E}_{\text{gen},N}(\mathbf{x})$ for this model:

$$\begin{aligned}
\mathcal{E}_{\text{gen},N}(\mathbf{x}) &= E[L(y, \hat{f}(\mathbf{x}))|\mathbf{x}, N] \\
&= E[(y - \hat{f}(\mathbf{x}))^2|\mathbf{x}, N] \\
&= E[(g(\mathbf{x}) + \epsilon - \hat{f}(\mathbf{x}))^2|\mathbf{x}, N] \\
&= E[(\epsilon + (g(\mathbf{x}) - \hat{f}(\mathbf{x})))^2|\mathbf{x}, N] \\
&= E[\epsilon^2 + 2\epsilon(g(\mathbf{x}) - \hat{f}(\mathbf{x})) + (g(\mathbf{x}) - \hat{f}(\mathbf{x}))^2|\mathbf{x}, N] \\
&= E[\epsilon^2] + 2E[\epsilon(g(\mathbf{x}) - \hat{f}(\mathbf{x}))|\mathbf{x}, N] + E[(g(\mathbf{x}) - \hat{f}(\mathbf{x}))^2|\mathbf{x}, N]
\end{aligned}$$

The first term $E[\epsilon^2]$ is the expectation of the square of a zero-mean normally distributed variable, which is exactly equal to the variance σ_ϵ^2. The second term $2E[\epsilon(g(\mathbf{x}) - \hat{f}(\mathbf{x}))|\mathbf{x}, N]$ is zero, because ϵ has an expectation of zero and is completely independent of $\mathcal{D}$ (which is the only other random variable). Now let's analyze the last term (I'll drop the conditioning, it's always $\mathbf{x}$ and N):

$$\begin{aligned}
E[(g(\mathbf{x}) - \hat{f}(\mathbf{x}))^2] &= E[(g(\mathbf{x}) - E[\hat{f}(\mathbf{x})] + E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2] \\
&= E[((g(\mathbf{x}) - E[\hat{f}(\mathbf{x})]) + (E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})))^2] \\
&= E[(g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + 2(g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})) + (E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2] \\
&= (g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + 2(g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])E[E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})] + E[(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2]
\end{aligned}$$

Let's look closer at the middle term: $2(g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])E[E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})]$. The second part of the product $E[E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})]$ simplifies, because the first part is deterministic, so we can write

$$E[E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x})] = E[\hat{f}(\mathbf{x})] - E[\hat{f}(\mathbf{x})] = 0,$$

and therefore the entire middle term is zero. So we are left with

$$E[(g(\mathbf{x}) - \hat{f}(\mathbf{x}))^2] = (g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + E[(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2].$$

Note that the second term is exactly the equation for the variance of $\hat{f}(\mathbf{x})$ (which depends on the random variable $\mathcal{D}$): variance of a random variable a is $E[(E[a] - a)^2]$! And the first term is the squared distance between the expected prediction $E[\hat{f}(\mathbf{x})]$ and the true underlying signal $g(\mathbf{x})$. We call this distance the bias, so this term is just the bias squared. Therefore, the entire error becomes:

$$\begin{aligned}
\mathcal{E}_{\text{gen},N}(\mathbf{x}) &= E[\epsilon^2] + (g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + E[(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2] \\
&= \sigma_\epsilon^2 + \text{Bias} + \text{Variance}.
\end{aligned}$$

Here we can see where the terms bias and variance really come from: the generalization error when averaged over all possible choices of the dataset $\mathcal{D}$ consists of an irreducible error σ_ϵ^2 (which is due to the inherently noise in the data), a bias term Bias that is caused by the expected hypothesis $E[\hat{f}(\mathbf{x})]$ being unable to accurately match the true underlying function $g(\mathbf{x})$, and a variance term that is caused by the variability in the hypothesis $\hat{f}(\mathbf{x})$ with respect to the choice of training set $\mathcal{D}$. Note that if some hypothesis $\hat{f}(\mathbf{x})$ can accurately match $g(\mathbf{x})$, then $E[\hat{f}(\mathbf{x})]$ will match $g(\mathbf{x})$, because averaging the optimal hypotheses over all possible datasets produces the optimal hypothesis![1] So the bias is a term that *never* goes away if our hypothesis class is not expressive enough. However, as the size of our dataset $\mathcal{D}$ increases, we would expect variance to go down: in the limit, if the dataset $\mathcal{D}$ contains all possible points, then all datasets of that size look "the same." Hence, our interpretation from earlier that bias is the gap between the minimium attainable error (which we see is σ_ϵ^2) and the actual error we observe as $N \to \infty$. Note that since our loss function is the squared loss, we can also rewrite this error in terms of the loss L:

$$\begin{aligned}\mathcal{E}_{\text{gen},N}(\mathbf{x}) &= E[\epsilon^2] + (g(\mathbf{x}) - E[\hat{f}(\mathbf{x})])^2 + E[(E[\hat{f}(\mathbf{x})] - \hat{f}(\mathbf{x}))^2] \\ &= \sigma_\epsilon^2 + L(g(\mathbf{x}), E[\hat{f}(\mathbf{x})]) + E[L(\hat{f}(\mathbf{x}), E[\hat{f}(\mathbf{x})])].\end{aligned}$$

Indeed, we can extend this notion of bias and variance to any loss function, as well as to the classification setting. The extension proceeds as following: first, we define the mean prediction

$$\bar{f}(\mathbf{x}) = E[\hat{f}(\mathbf{x})]$$

This is the prediction that will be made by the optimal hypothesis with respect to the training error, averaged over all possible training sets $\mathcal{D}$. The bias, just like above, is simply the loss incurred by the mean prediction against the true underlying signal:

$$\text{Bias} = L(g(\mathbf{x}), \bar{f}(\mathbf{x})).$$

The variance is the average loss incurred by a hypothesis $\hat{f}(\mathbf{x})$ relative to the mean hypothesis $\bar{f}(\mathbf{x})$, averaged over all possible training sets:

$$\text{Variance} = L(\bar{f}(\mathbf{x}), \hat{f}(\mathbf{x})).$$

This definition can be applied to likelihoods, "0-1" losses, hinge losses, etc.

[1] As an exercise, it is worthwhile to think about how this idea connects to bagging.