Notes for *Elements of Statistical Learning*

Sebastian Claici

November 29, 2024

Chapter 2: Overview of Supervised Learning

Statistical Decision Theory

The goal is to minimize the expected prediction error:

$$\begin{aligned}\mathrm{EPE}(f) &= \mathrm{E}\left(Y - f(X)\right)^2 \\ &= \int [y - f(x)]^2 p(x, y)\,\mathrm{d}x\,\mathrm{d}y \qquad (1)\end{aligned}$$

If we break down the expectation as $E_{X,Y} = E_X E_{Y|X=x}$ we can rewrite this as

$$\begin{aligned}\mathrm{EPE}(f) &= \mathrm{E}_X \mathrm{E}_{Y|X=x}(Y - f(X))^2 \\ &= \int_X \int_Y [y - f(x)]^2 p(y|X = x) p(x)\,\mathrm{d}y\,\mathrm{d}x \\ &= \int_X p(x) \left(\int_Y [y - f(x)]^2 p(y|X = x)\,\mathrm{d}y \right) \mathrm{d}x\end{aligned}$$

We have moved the dependence on $p(x)$ outside the inner expectation. Since f is unconstrained, we can solve for the optimal f pointwise. That is:

$$\arg\min_f \mathrm{EPE}(f) = \arg\min_c \int_Y [y - c]^2 p(y|X = x)\,\mathrm{d}y$$

Differentiating wrt c and using the fact that

$$\int_Y y\ p(y|X = x)\,\mathrm{d}y = \mathrm{E}(Y|X = x)$$

gives us (2.13) in the book.

Nearest-neighbor methods try to model the regression function directly by averaging predictions around the query point x. To drive this point home, we can show that $\mathrm{NN}(x) \to x$ as the number of training points $N \to \infty$.

To sketch this proof out, assume $x_1, \ldots, x_N$ are drawn i.i.d from X. We want to bound $\min_i \|x - x_i\|$, but since this is a bit complicated, let's instead compute

$$\mathrm{P}(\|x - x_i\| \geq \varepsilon, \forall i).$$

for some $\varepsilon > 0$.

Since the x_i are sampled independently, we can expand the probability as

$$\mathrm{P}(\|x - x_i\| \geq \varepsilon, \forall i) = \prod_{i=1}^{N} \mathrm{P}(\|x - x_i\| \geq \varepsilon).$$

As the x_i are also identically distributed, the product can be written as

$$[\mathrm{P}(\|x - x_i\| \geq \varepsilon)]^N$$

which goes to 0 as $N \to \infty$ as long as the probability is not exactly 1. This shows that with infinite samples the Nearest-neighbor of x is x and so nearest neighbors yields the Bayes optimal decision boundary even with a single neighbor.

However, we often do not have enough samples to use a model-free approach to regression. The second proposal is to assume the regression function is linear in its arguments:

$$f(x) \approx x^T \beta$$

If we plug this for f into (1), we get

$$\int [y - x^T \beta]^2 p(x, y) \,\mathrm{d}x \,\mathrm{d}y.$$

We can differentiate this wrt β[1]

$$\begin{aligned}
\frac{\partial \mathrm{EPE}}{\partial \beta} &= 2 \int x[y - x^T \beta] p(x, y) \,\mathrm{d}x \,\mathrm{d}y \\
&= 2 \left(\int xy \; p(x, y) \,\mathrm{d}x \,\mathrm{d}y - \int x x^T \beta \; p(x, y) \,\mathrm{d}x \,\mathrm{d}y \right)
\end{aligned}$$

Since β is not a random variable, we can set this to 0 to arrive at the minimizer in (2.16) in the book:

$$\beta = [\mathrm{E}(XX^T)]^{-1} \mathrm{E}(XY)$$

[1]See this link for a review of matrix calculus.

Bias-Variance Decomposition

We can express the mean-squared error in terms of a squared bias term and a variance term. In equation (2.25) in the book, these vary w.r.t. the training set T. To clarify the notation a bit, x_0 is the point 0, $\hat{y}_0$ is the model estimate (in this case the nearest neighbor estimate), and $f(x_0)$ is the true value at 0, but the following derivation holds generally for any model approximation $\hat{y}$ of a function $f(x)$[2]:

$$
\begin{aligned}
\mathrm{MSE}(x_0) &= \mathrm{E}_T\left[f(x_0) - \hat{y}_0\right]^2 \\
&= \mathrm{E}_T\left[\hat{y}_0 - \mathrm{E}_T[\hat{y}_0] + \mathrm{E}_T[\hat{y}_0] - f(x_0)\right]^2 \\
&= \mathrm{E}_T\left[\hat{y}_0 - \mathrm{E}_T[\hat{y}_0]\right]^2 + \left(f(x_0) - \mathrm{E}_T[\hat{y}_0]\right)^2 \\
&= \mathrm{Var}_T(\hat{y}_0) + \mathrm{Bias}^2(\hat{y}_0)
\end{aligned}
$$

It is a somewhat instructive exercise to figure out how to go from the second line to the third. Easiest if you recall that

$$
\begin{gathered}
\mathrm{E}_T[\mathrm{E}_T[y]] = \mathrm{E}_T[y] \\
\mathrm{E}_T[f(x)] = f(x)
\end{gathered}
$$

In the example in the book, the variance is consistently low, but the bias increases with dimension as the nearest point to 0 becomes increasingly distant.

We can discuss equations (2.27) and (2.28) in the book briefly. We have

$$\hat{\beta} = (X^T X)^{-1} X^T y = (X^T X)^{-1} X^T (X\beta + \varepsilon) = \beta + (X^T X)^{-1} X^T \varepsilon$$

and thus

$$\hat{y}_0 = x_0^T \hat{\beta} = x_0^T(\beta + (X^T X)^{-1} X^T \varepsilon)$$

which gives

$$\hat{y}_0 = x_0^T \beta + \sum_{i=1}^{N} l_i(x_0)\varepsilon_i \tag{2}$$

since $x_0^T(X^T X)^{-1} X^T \varepsilon$ is a scalar and

$$(x_0^T (X^T X)^{-1} X^T)^T = X(X^T X)^{-1} x_0$$

to give the expression in the book.

Let's write out $\mathrm{EPE}(x_0)$. Note that because the true data was generated from a noisy process, we have to integrate out for y_0 given a fixed x_0:

$$\mathrm{EPE}(x_0) = \mathrm{E}_{y_0|x_0}\mathrm{E}_T[y_0 - \hat{y}_0]^2$$

[2]See, for example, the wikipedia page.

In this particular case, because Y depends on X stochastically, $\mathrm{E}_T = \mathrm{E}_X \mathrm{E}_{Y|X}$

We will write out $y_0 - \hat{y}_0$ as:

$$y_0 - \hat{y}_0 = (y_0 - x_0^T \beta) + (x_0^T \beta - \mathrm{E}_T[\hat{y}_0]) + (\mathrm{E}_T[\hat{y}_0] - \hat{y}_0)$$

Let's square this, and keep in mind that $\mathrm{E}[\varepsilon] = 0$, and $\mathrm{Var}[\varepsilon] = \sigma^2$:

$$\begin{aligned}\mathrm{E}_{y_0|x_0} \mathrm{E}_T [y_0 - \hat{y}_0]^2 =& \mathrm{E}_{y_0|x_0}[y_0 - x_0^T \beta]^2 + (x_0^T \beta - \mathrm{E}_T[\hat{y}_0])^2 + \mathrm{E}_T[\hat{y}_0 - \mathrm{E}_T[\hat{y}_0]]^2 + \\ & \text{cross terms}\end{aligned}$$

For the cross terms, we notice the following:

$$\begin{aligned}\mathrm{E}_{y_0|x_0}(y_0 - x_0^T \beta) = 0 \\ \mathrm{E}_T(\mathrm{E}_T[\hat{y}_0] - \hat{y}_0) = 0\end{aligned}$$

and

$$\begin{aligned}(x_0^T \beta - \mathrm{E}_T[\hat{y}_0]) &= \mathrm{E}_T \left[\sum_{i=1}^{N} l_i(x_0) \varepsilon_i \right] \\ &= \mathrm{E}_X \left[\sum_{i=1}^{N} l_i(x) \mathrm{E}_{Y|X}(\varepsilon_i) \right] \\ &= 0 \end{aligned} \tag{3}$$

where we have used (2).

This gives

$$\begin{aligned}\mathrm{E}_{y_0|x_0} \mathrm{E}_T [y_0 - \hat{y}_0]^2 &= \mathrm{E}_{y_0|x_0}[y_0 - x_0^T \beta]^2 + (x_0^T \beta - \mathrm{E}_T[\hat{y}_0])^2 + \mathrm{E}_T[\hat{y}_0 - \mathrm{E}_T[\hat{y}_0]]^2 + \\ &= \mathrm{Var}[y_0|x_0] + \mathrm{Bias}^2(\hat{y}_0) + \mathrm{Var}_T(\hat{y}_0)\end{aligned}$$

but the bias is 0 by (3), and $\mathrm{Var}[y_0|x_0] = \sigma^2$, we have:

$$\mathrm{E}_{y_0|x_0} \mathrm{E}_T [y_0 - \hat{y}_0]^2 = \sigma^2 + \mathrm{Var}_T(\hat{y}_0).$$

To finish the derivation, let's write out $\mathrm{Var}_T(\hat{y}_0)$. We have just proved that $\mathrm{E}_T(\hat{y}_0) = x_0^T \beta$, and so

$$\begin{aligned}\mathrm{Var}_T(\hat{y}_0) &= \mathrm{E}_T \left[x_0^T (X^T X)^{-1} X^T \varepsilon \right]^2 \\ &= \mathrm{E}_T \left[x_0^T (X^T X)^{-1} X^T \varepsilon \varepsilon^T X (X^T X)^{-1} x_0 \right]\end{aligned}$$

Since $\varepsilon \sim N(0, \sigma^2)$, $\varepsilon \varepsilon^T = \sigma^2 I_N$, and we can replace above:

$$\begin{aligned}\mathrm{Var}_T(\hat{y}_0) &= \mathrm{E}_T \left[x_0^T (X^T X)^{-1} X^T X (X^T X)^{-1} x_0 \right] \sigma^2 \\ &= \mathrm{E}_T \left[x_0^T (X^T X)^{-1} x_0 \right] \sigma^2\end{aligned}$$

which is the value in the book.

To derive (2.28), we assume large N and that $X^T X \to N\text{Cov})(X)$, hence:

$$\begin{aligned} \mathrm{E}_{x_0}\mathrm{EPE}(x_0) &= \sigma^2 + \mathrm{E}_{x_0}\left[x_0^T(X^TX)^{-1}x_0\right]\sigma^2 \\ &\sim \sigma^2 + \mathrm{E}_{x_0}\left[x_0^T\mathrm{Cov}(X)^{-1}x_0\right]\sigma^2/N \end{aligned} \tag{4}$$

Now $x_0^T\mathrm{Cov}(X)^{-1}x_0$ is a scalar, and can be written as $\mathrm{trace}[x_0^T\mathrm{Cov}(X)^{-1}x_0]$. Exploiting the cyclic properties of the trace operator:

$$\begin{aligned} \mathrm{E}_{x_0}\mathrm{trace}[x_0^T\mathrm{Cov}(X)^{-1}x_0] &= \mathrm{E}_{x_0}\mathrm{trace}[\mathrm{Cov}(X)^{-1}x_0x_0^T] \\ &= \mathrm{trace}[\mathrm{Cov}(X)^{-1}\mathrm{E}_{x_0}(x_0x_0^T)] \\ &= \mathrm{trace}[\mathrm{Cov}(X)^{-1}\mathrm{Cov}(x_0)] \end{aligned}$$

Since $\mathrm{Cov}(X) = \mathrm{Cov}(x_0)$, and since each training set point is p-dimensional:

$$\mathrm{trace}[\mathrm{Cov}(X)^{-1}\mathrm{Cov}(x_0)] = \mathrm{trace}[I_p] = p$$

which when replaced in (4) gives equation (2.28) in the book.

Exercises