LECTURE 3, PART II: ESTIMATING THE PREDICTION RISK IN PRACTICE

Review: Recall from last time that we defined the prediction risk as

$$R = \mathbb{E}(Y - \widehat{r}(X))^2 = \sigma^2 + \mathbb{E}(r(X) - \widehat{r}(X))^2 = \sigma^2 + \text{MSE},$$

where MSE means mean-squared-error. Also,

$$\text{MSE} = \int \text{bias}^2(x)p(x)dx + \int \text{var}(x)p(x)dx$$

where

$$\text{bias}(x) = \mathbb{E}(\widehat{r}(x)) - r(x))$$

is the bias of $\widehat{r}(x)$ and

$$\text{var}(x) = \text{Variance}(\widehat{r}(x))$$

is the variance.

When the data are oversmoothed, the bias term is large and the variance is small. When the data are undersmoothed the opposite is true; see Figure 1. This is called the **bias–variance tradeoff**. Minimizing risk corresponds to balancing bias and variance.

Ideally, we would like to choose the model with the smallest risk R but R depends on the unknown function $r(x)$. Instead, we will minimize an estimate $\widehat{R}(h)$ of $R(h)$. As we showed last time, the average residual sums

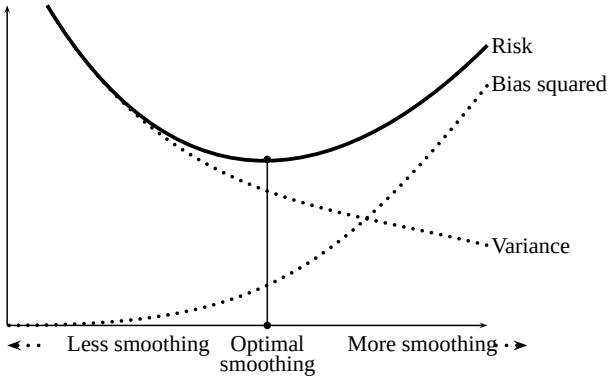

Figure 1: The bias–variance tradeoff. The bias increases and the variance decreases with the amount of smoothing. The optimal amount of smoothing, indicated by the vertical line, minimizes the risk = bias^2 + variance.

of squares, also called the **training error**,

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \widehat{r}_n(X_i))^2, \tag{1}$$

is a poor estimate of R: it is biased downwards and typically leads to undersmoothing (overfitting). In this lecture, we will discuss other approaches for estimating the prediction error from data, as well as their respective performances.

[1]

A Single Hold-Out Test Point ("toy example")

In practice, if we fit $\widehat{r}$ on the data (X_i, Y_i), $i = 1, \ldots n$, then we don't typically have a way of computing the expected test error.

Now looking back at how we defined the prediction error, we can see that

we need a test point (X, Y) *independent* of the training set; so without access to any more data, what can we do?

One idea is to fit our prediction function on the first $n-1$ data points (X_i, Y_i), $i = 1, \ldots n-1$, call it $\widehat{r}_{(-n)}$, and then treat the last point (X_n, Y_n) as a test point.

We then have the estimated error

$$\left(Y_n - \widehat{r}_{(-n)}(X_n)\right)^2.$$

Let us consider the following two questions:

1. Does this estimate have the right expectation?

2. Is this estimate variable?

Leave-One-Out Cross-Validation

Here is a natural extension of the above: Hold out each point (X_i, Y_i) from our training set in turn, fit $\widehat{r}_{(-i)}$ on all points except this one (i.e., (X_j, Y_j), $j \neq i$), record the squared error on (X_i, Y_i), and average the results. This yields the estimated risk, $\widehat{R}_{\mathrm{CV}}$:

This method is called *leave-one-out cross-validation* (sometimes acronymized LOOCV), and the error estimate is sometimes referred to as *leave-one-out cross-validation score*.

Compared to using a single hold-out test point, we have increased the computational burden (by a factor of n), but with the advantage of greatly reducing the variance of our error estimate.

We haven't really changed the expectation of our test error and we still have that

$$\mathbb{E}(\widehat{R}) \approx \text{predictive risk},$$

that is, the cross-validation score is a *nearly unbiased estimate of the risk*.

As we are going to see later, there is a shortcut formula for computing the leave-one-out CV score for linear smoothers.

K-Fold Cross-Validation

There is yet another way to carve things up. We split our data into K roughly equal-sized parts or *folds*, for some number K. Usually this division is done randomly. Write these as $F_1, \ldots F_K$ (so $F_1 \cup \ldots \cup F_K = \{1, \ldots n\}$). For example, when $K = 5$, the scenario looks like this:

Now for each $j = 1, \ldots K$, we fit our prediction model on all points *except for* those in the jth fold (third above), call it $\widehat{r}_j$, and evaluate the error on the points in the jth fold, CV_j:

Here n_j denotes the number of points in the jth fold, $n_j = |F_j|$. Finally we *average* these fold-based errors to yield the cross-validation estimate of the prediction error, $\widehat{R}_{\mathrm{CV}}$:

This is called *K-fold cross validation*, and note that leave-one-out cross-validation is a special case of this corresponding to $K = n$.

What value should we choose for K? Another highly common choice (other than $K = n$) is to choose $K = 5$ or $K = 10$. What does a smaller value of K do? Explain in terms of bias and variance of the estimated error.

Cross-Validation Standard Errors

What is the variability of the cross-validation error estimate? Can we assign a quantitative notion of variability? We argue the following for the variance of $\widehat{R}_{\mathrm{CV}}$:

Why is this an approximation? This would hold exactly if $\mathrm{CV}_1(\widehat{r}_1), \ldots \mathrm{CV}_K(\widehat{r}_K)$ were i.i.d., but they're not. This approximation is valid for *small* K (e.g., $K = 5$ or 10) but not really for big K (e.g., $K = n$), because then the quantities $\mathrm{CV}_1(\widehat{r}_1), \ldots \mathrm{CV}_K(\widehat{r}_K)$ are highly correlated.

For small K (e.g., $K = 5$ or 10), we can use the above approximation to get an estimate of the variance of the cross-validation error estimate. We compute the sample variance and the sample standard deviation or *standard error* of the CV error estimate:

Model Selection In Practice

In summary:

- To choose between models in practice, we simply compute cross-validated errors for each, and then take the model with the minimum cross-validated error
- For tuning parameter selection (as in k in k-nearest neighbors), write $\widehat{r}_\theta$ and $R(\theta)$ to denote that our regression estimator and its prediction error depend on some parameter θ. Then for a range of parameter values $\theta_1, \ldots \theta_m$ of interest, we compute the cross-validation scores

  $$\widehat{R}(\theta_1), \ldots \widehat{R}(\theta_m),$$

  and choose the value of θ minimizing the cross-validation error curve (a curve over θ),

  $$\widehat{\theta} = \underset{\theta \in \{\theta_1, \ldots \theta_m\}}{\operatorname{argmin}} \widehat{R}(\theta)$$

- For close calls, pay attention to standard errors! More on this later ...

R Demo 3.1: In-Sample, Out-of-Sample and Cross-Validation Errors

[Ref: Shalizi Sections 3.3 and 3.4]

(a) Sample 50 X values from a Gaussian distribution, and let $Y = 7X^2 - 0.5X + \epsilon$, $\epsilon \sim \mathcal{N}(0, 1)$. That is, the true regression curve is a parabola, with additive and independent Gaussian noise. Plot the "training data" and the true regression curve.

(b) Fit polynomials of degree 0 to 7 and add them to the plot. Calculate and plot *in-sample errors*.

(c) Draw 10,000 *new* data points from the same distribution as the training data. Plot the old curves with the "testing data". Calculate and plot the *out-of-sample errors*. Compare with the in-sample errors.

(d) Create folds for 5-fold cross-validation. Perform cross-validation. Plot the individual cross-validation error curves for each fold.

(e) Average across folds and compute the standard errors. Plot the (averaged) *cross-validation error curve with standard errors*.