IEOR 290A – Lecture 4
Cross-Validation

1 Bias and Variance of OLS

We begin by computing the expectation of the OLS estimate

$$\begin{aligned}
\mathbb{E}(\hat{\beta}) &= \mathbb{E}((X'X)^{-1}X'Y) \\
&= \mathbb{E}((X'X)^{-1}X'(X\beta + \underline{\epsilon})) \\
&= \mathbb{E}(\beta + (X'X)^{-1}X'\underline{\epsilon}) \\
&= \beta + \mathbb{E}(\mathbb{E}[(X'X)^{-1}X'\underline{\epsilon}|X]) \\
&= \beta.
\end{aligned}$$

As a result, we conclude that bias$(\hat{\beta}) = 0$.

Now assume that the x_i are independent and identically distributed (i.i.d.) random variables, with mean zero $\mathbb{E}(x_i) = 0$ and have covariance matrix Σ that is **assumed to be invertible**. By the weak law of large numbers (wlln), we have that $\frac{1}{n}X'X \xrightarrow{p} \Sigma$ and that $\frac{1}{n}X'Y \xrightarrow{p} \Sigma\beta$. Using the Continuous Mapping Theorem, we have that $\hat{\beta} \xrightarrow{p} \beta$. When an estimate converges in probability to its true value, we say that the estimate is *consistent*. Thus, the OLS estimate is consistent under the model we have described; however, consistency does not hold if Σ is not invertible.

We next determine the asymptotic variance of the OLS estimate. By the Central Limit Theorem (CLT), we have that $\frac{1}{n}X'X = \Sigma + O_p(\frac{1}{\sqrt{n}})$ and $\frac{1}{n}X'Y = \Sigma\beta + O_p(\frac{1}{\sqrt{n}})$. Applying the Continuous Mapping Theorem gives that $\hat{\beta} = \beta + O_p(\frac{1}{\sqrt{n}})$. Since we know that

$$\mathbb{E}((\hat{\beta} - \beta)^2) = (\text{bias}(\hat{\beta}))^2 + \text{var}(\hat{\beta}),$$

we must have that var$(\hat{\beta}) = O_p(1/n)$, since we showed above that bias$(\hat{\beta}) = 0$.

2 Bias and Variance of LLR

Next, we describe the bias and variance of LLR. The heuristic derivations are more complicated, and so we do not include them. It can be shown that the bias and variance of an LLR estimate is

$$\begin{aligned}
\text{bias}(\hat{\beta}_0[x_0]) &= O(h^2) \\
\text{var}(\hat{\beta}_0[x_0]) &= O_p\left(\frac{1}{nh^d}\right),
\end{aligned}$$

where h is the bandwidth and d is the dimensionality of x_i (i.e., $x_i \in \mathbb{R}^d$). Note that we have described the bias and variance for the estimate of β_0 and not that of β. The bias and variance for the estimate of β are slightly different. The interesting aspect is that these expressions show how the choice of the bandwidth h affects bias and variance. A large bandwidth means that we have high bias and low variance, whereas a small bandwidth means that we have a low bias but high variance.

When implementing LLR, we need to pick the bandwidth h in some optimal sense that trades off these competing effects. We can, at least analytically, pick the asymptotic rate of h:

$$\begin{aligned}
h &= \arg\min_h \left(c_1 h^4 + c_2 \frac{1}{nh^d} \right) \\
&\Rightarrow \tilde{c}_1 h^3 + \tilde{c}_2 \frac{1}{nh^{d+1}} = 0 \\
&\Rightarrow \bar{c}_1 h^{d+4} = \bar{c}_2 \frac{1}{n} \\
&\Rightarrow h = O(n^{-1/(d+4)}).
\end{aligned}$$

The resulting optimal rate is that

$$\beta_0 = \beta + O_p(n^{-2/(d+4)}).$$

Note that this is slower than the rate of OLS (i.e, $O_p(1/\sqrt{n})$) for any $d \geq 1$, and is an example of the statistical penalty that occurs when using nonparametric methods. The situation is in fact even worse: There is a "curse of dimensionality" that occurs, because the convergence rate gets exponentially worse as the ambient dimension d increases. It turns out that if the model has additional structure, then one can improve upon this "curse of dimensionality" and get better rates of convergence when using LLR; this will be discussed later in the course.

3 Cross-Validation

Cross-validation is a data-driven approach that is used to choose tuning parameters for regression. The choice of bandwidth h is an example of a tuning parameter that needs to be chosen in order to use LLR. The basic idea of cross-validation is to split the data into two parts. The first part of data is used to compute different estimates (where the difference is due to different tuning parameter values), and the second part of data is used to compute a measure of the quality of the estimate. The tuning parameter that has the best computing measure of quality is selected, and then that particular value for the tuning parameter is used to compute an estimate using all of the data. Cross-validation is closely related to the jackknife and bootstrap methods, which are more general. We will not discuss those methods here.

3.1 Leave k-Out Cross-Validation

We can describe this method as an algorithm.

```
data  : $(x_i, y_i)$ for $i = 1, \dots, n$ (measurements)
input : $\lambda_j$ for $j = 1, \dots, z$ (tuning parameters)
input : $R$ (repetition count)
input : $k$ (leave-out size)
output: $\lambda^*$ (cross-validation selected tuning parameter)
for $j \leftarrow 1$ to $z$ do
    set $e_j \leftarrow 0$;
end
for $r \leftarrow 1$ to $R$ do
    set $\mathcal{V}$ to be $k$ randomly picked indices from $\mathcal{I} = \{1, \dots, n\}$;
    for $j \leftarrow 1$ to $z$ do
        fit model using $\lambda_j$ and $(x_i, y_i)$ for $i \in \mathcal{I} \setminus \mathcal{V}$;
        compute cross-validation error $e_j \leftarrow e_j + \sum_{i \in \mathcal{V}} (y_i - \hat{y}_i)^2$;
    end
end
set $\lambda^* \leftarrow \lambda_j$ for $j := \arg\min e_j$;
```

3.2 k-Fold Cross-Validation

We can describe this method as an algorithm.

```
data  : $(x_i, y_i)$ for $i = 1, \dots, n$ (measurements)
input : $\lambda_j$ for $j = 1, \dots, z$ (tuning parameters)
input : $k$ (block sizes)
output: $\lambda^*$ (cross-validation selected tuning parameter)
for $j \leftarrow 1$ to $k$ do
    set $e_j \leftarrow 0$;
end
partition $\mathcal{I} = \{1, \dots, n\}$ into $k$ randomly chosen subsets $\mathcal{V}_r$ for $r = 1, \dots, k$;
for $r \leftarrow 1$ to $k$ do
    for $j \leftarrow 1$ to $z$ do
        fit model using $\lambda_j$ and $(x_i, y_i)$ for $i \in \mathcal{I} \setminus \mathcal{V}_r$;
        compute cross-validation error $e_j \leftarrow e_j + \sum_{i \in \mathcal{V}_r} (y_i - \hat{y}_i)^2$;
    end
end
set $\lambda^* = \lambda_j$ for $j := \arg\min e_j$;
```

3.3 Notes on Cross-Validation

There are a few important points to mention. The first point is that the cross-validation error is an estimate of prediction error, which is defined as

$$\mathbb{E}((\hat{y} - y)^2).$$

One issue with cross-validation error (and this issue is shared by jackknife and bootstrap as well), is that these estimates of prediction error must necessarily be biased lower. The intuition is that we are trying to estimate prediction error using data we have seen, but the true prediction error involves data we have not seen. The second point is related, and it is that the typical use of cross-validation is heuristic in nature. In particular, the consistency of an estimate is affected by the use of cross-validation. There are different cross-validation algorithms, and some algorithms can "destroy" the consistency of an estimator. Because these issues are usually ignored in practice, it is important to remember that cross-validation is usually used in a heuristic manner. The last point is that we can never eliminate the need for tuning parameters. Even though cross-validation allows us to pick a λ^* in a data-driven manner, we have introduced new tuning parameters such as k. The reason that cross-validation is considered to be a data-driven approach to choosing tuning parameters is that estimates are usually less sensitive to the choice of cross-validation tuning parameters, though this is not always true.