WSABI Summer Research Lab **Summer 2025**

Lecture 16: Bias-Variance Tradeoff

Instructor: Jonathan Pipping *Author: Ryan Brill*

16.1 Introduction

The **Bias-Variance Tradeoff** is arguably the most important concept in machine learning! Basically every model we fit will address this tradeoff in some way, and it's important to be aware of how they do so. We'll begin by reviewing where we've seen this concept in previous lectures.

16.1.1 Review from Past Lectures

We saw in Lecture 1 that the OLS estimator is **unbiased**, meaning that its expected value is equal to the true parameter (meaning on average, the OLS estimator is "correct"). However, we saw in Lecture 14 that this useful property comes at the cost of **high sensitivity** to noise, meaning it is prone to overfit to idiosyncracies in the training set, especially in the presence of **multicollinearity** or **small sample sizes**.

As an alternative to this, we considered constant estimators like 0 or the overall mean. These estimators are **extremely** stable (meaning they are resistant to noise), but they are also **extremely biased**, meaning that they are not correct on average.

To balance these two extremes, we introduced Ridge Regression , which shrinks OLS estimates towards 0. This **biases** our estimates, but it also makes them **more stable**, meaning they are less sensitive to noise and less likely to overfit.

We've seen comparatively how sensitive these estimators are to noise in Lecture 14, but now we wnat to **quantify** this sensitivity for **any given estimator**. We establish this in the next section.

16.2 Bias-Variance Decomposition

16.2.1 Model Setup and the Mean Squared Error

Consider the following general model formulation:

$$y_i = f(x_i) + \epsilon_i, \quad \mathbb{E}[\epsilon_i] = 0.$$

where y_i is the response variable, x_i are our predictors, f is the "true" underlying function, and ϵ_i is the noise. In general, the goal of Machine Learning is to estimate f as "best" as possible from a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$. Let's denote our estimate of f as $\hat{f} = \hat{f}(x \mid \mathcal{D})$.

We want our estimate $\hat{f}$ to be as "close" as possible to the true f, but since our estimate depends on a randomly-drawn training set $\mathcal{D}$, we are really interested in the **expected** (or average) performance of $\hat{f}$ across all possible training sets. The Mean Squared Error (MSE) is a natural way to quantify this, averaging over the randomness in the training set $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$:

$$\text{MSE}(x \mid \mathcal{D}) = \mathbb{E}_{\mathcal{D}}\left[(f(x) - \hat{f}(x \mid \mathcal{D}))^{(2)}\right].$$

But since we don't actually observe $f(x)$, we instead use out-of sample testing data instead. So our out-of-sample MSE is

$$\text{MSE}_{\text{out}}(x \mid \mathcal{D}) = \mathbb{E}_{\mathcal{D}}\left[(Y - \hat{f}(x \mid \mathcal{D}))^{(2)}\right]$$

where Y is the true response variable and $\hat{f}$ is our estimate.

Note that this requires our full dataset to be split into two parts: a training set $\mathcal{D}$ and a testing set. The former is used to calculate our estimate $\hat{f}$, while the latter is used to evaluate it. We proceed by decomposing the MSE into its subcomponents.

16.2.2 Deriving the Bias-Variance Decomposition

Recall our expression for the out-of-sample MSE:

$$\text{MSE}_{\text{out}}(x \mid \mathcal{D}) = \mathbb{E}_{\mathcal{D}}\left[(Y - \hat{f}(x \mid \mathcal{D}))^{(2)}\right]$$

Concisely, we can write this as

$$\begin{aligned}
\text{MSE} &= \mathbb{E}\left[Y - \hat{f}\right]^{(2)} \quad \text{where } \hat{f} = \hat{f}(x \mid \mathcal{D}). \\
&= \mathbb{E}\left[Y^{(2)} - 2Y\hat{f} + \hat{f}^{(2)}\right] \\
&= \mathbb{E}\left[Y^{(2)}\right] - 2\mathbb{E}\left[Y\hat{f}\right] + \mathbb{E}\left[\hat{f}^{(2)}\right]
\end{aligned}$$

Since $Y = f + \epsilon$, we have

$$\begin{aligned}
MSE &= \mathbb{E}\left[(f+\epsilon)^{(2)}\right] - 2\mathbb{E}\left[(f+\epsilon)\,\hat{f}\right] + \mathbb{E}\left[\hat{f}^{(2)}\right] \\
&= \mathbb{E}\left[f^{(2)} + 2f\epsilon + \epsilon^{(2)}\right] - 2\mathbb{E}\left[f\hat{f} + \epsilon\hat{f}\right] + \mathbb{E}\left[\hat{f}^{(2)}\right]
\end{aligned}$$

Since $f(x)$ is an unknown fixed constant and $\hat{f} \perp\!\!\!\perp \epsilon$, we have

$$MSE = f^{(2)} + 2f\mathbb{E}\left[\epsilon\right] + \mathbb{E}\left[\epsilon^{(2)}\right] - 2f\mathbb{E}\left[\hat{f}\right] - 2\mathbb{E}\left[\epsilon\right]\mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}^{(2)}\right]$$

Since $\mathbb{E}\left[\epsilon\right] = 0$, we have

$$\begin{aligned}
MSE &= f^{(2)} - 2f\mathbb{E}\left[\hat{f}\right] + \mathbb{E}\left[\hat{f}^{(2)}\right] + \mathbb{E}\left[\epsilon^{(2)}\right] \\
&= f^{(2)} - 2f\mathbb{E}\left[\hat{f}\right] + \left(\mathbb{E}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}\left[\hat{f}^{(2)}\right] - \left(\mathbb{E}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}\left[\epsilon^{(2)}\right] \\
&= \left(f - \mathbb{E}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}\left[\hat{f}^{(2)}\right] - \left(\mathbb{E}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}\left[\epsilon^{(2)}\right] \\
&= \left(f - \mathbb{E}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}\left[\left(\hat{f} - \mathbb{E}\left[\hat{f}\right]\right)^{(2)}\right] + \mathbb{E}\left[\epsilon^{(2)}\right]
\end{aligned}$$

So we have the following decomposition for our out-of-sample MSE:

$$\begin{aligned}\text{MSE}_{\text{out}}(x \mid \mathcal{D}) &= \mathbb{E}_{\mathcal{D}}\left[(Y - \hat{f}(x \mid \mathcal{D}))^{(2)}\right] \\ &= \left(f - \mathbb{E}_{\mathcal{D}}\left[\hat{f}\right]\right)^{(2)} + \mathbb{E}_{\mathcal{D}}\left[\left(\hat{f} - \mathbb{E}_{\mathcal{D}}\left[\hat{f}\right]\right)^{(2)}\right] + \mathbb{E}\left[\epsilon^{(2)}\right] \\ &= \text{Bias}(\hat{f})^{(2)} + \text{Var}(\hat{f}) + \text{Irreducible Error}\end{aligned}$$

Let's take a deeper look at each of these components.

$$\text{Bias}(\hat{f}) = \left(f - \mathbb{E}_{\mathcal{D}}\left[\hat{f}\right]\right)^{(2)}$$

This bias term tells us how close our estimate $\hat{f}$ (which is fit from $\mathcal{D}$) is to the true function f, averaged over $N \to \infty$ draws of the training set $\mathcal{D}$.

$$\text{Var}(\hat{f}) = \mathbb{E}_{\mathcal{D}}\left[\left(\hat{f} - \mathbb{E}_{\mathcal{D}}\left[\hat{f}\right]\right)^{(2)}\right]$$

This variance term tells us how close our estimate $\hat{f}$ is to the expected value of $\hat{f}$, averaged over $N \to \infty$ draws of the training set $\mathcal{D}$. In other words, it tells us how variable $\hat{f}$ is across different draws of the training set $\mathcal{D}$, or how sensitive $\hat{f}$ is to the random idiosyncracies of this training set.

$$\text{Irreducible Error} = \mathbb{E}\left[\epsilon^{(2)}\right] = \sigma_{\epsilon}^{(2)}$$

This irreducible error term tells us the noise that is inherent to the problem (e.g. for park effects, $\sigma_{\epsilon}^{(2)}$ is inherent noise of scoring a certain number of runs in a half inning). Overfitting occurs when a model memorizes the noise of the training set $\mathcal{D}$ instead of the true underlying trent f. High-variance estimates are more prone to overfitting.

16.2.3 Back to Our Park Effects Models

Consider the models we fit in Lecture 14. What do we expect the bias and variance to be for each of these models?

- **Overall Mean**: Very low variance and very high bias.
- **OLS**: Low bias, but high variance.
- **Ridge**: Introduces bias with the penalty term $\lambda \sum_{j=1}^{p} \beta_j^{(2)}$ to lower the variance. Has bias and variance that are between the overall mean and OLS.

Takeaway: To make better predictions, sometimes it helps to introduce some bias to lower the variance!

As we said before, the bias-variance tradeoff is a fundamental concept in machine learning, and we'll be revisiting it throughout this unit.