The Bias-Variance-Noise Decomposition

Following Hastie, Tibshirani, and Friedman (ESL)

FECD-B

ESRI and DCC-UFMG

June 11, 2026

The Setup: Supervised Learning Framework

Following the framework in *The Elements of Statistical Learning* (Chapter 2/7):

- Let Y be a target random variable generated by a true underlying function $\mu(X)$ corrupted by additive noise ε:

$$Y = \mu(X) + \varepsilon$$

- The noise term ε satisfies:
  - $\mathbb{E}(\varepsilon) = 0$
  - $\text{Var}(\varepsilon) = \sigma^2$ (Irreducible error)
  - ε is independent of the input vector X.
- We use a training dataset $\mathcal{T}$ to fit a model/hypothesis $\hat{\mu}(X)$.
- Note that $\hat{\mu}(X)$ is itself a random variable because it depends explicitly on the random draw of $\mathcal{T}$.

Expected Prediction Error (EPE)

Goal

We want to evaluate the performance of our estimate $\hat{\mu}(X)$ at a specific, fixed test point $X = x_0$ over all possible realizations of the training set $\mathcal{T}$.

Using the squared-error loss function, the Expected Prediction Error (EPE) at x_0 is defined as:

$$\text{Err}(x_0) = \mathbb{E}_{\mathcal{T}, Y|X=x_0}\left[(Y - \hat{\mu}(x_0))^2\right]$$

The expectation is taken with respect to two independent sources of randomness:

1. The target Y at the test point x_0 (due to the noise ε).
2. The variations in $\hat{\mu}$ due to drawing different training sets $\mathcal{T}$.

Deriving the Decomposition: Step 1

Let's expand the error term by adding and subtracting $\mu(x_0)$ inside the square:

$$\begin{aligned}
\text{Err}(x_0) &= \mathbb{E}\left[(Y - \hat{\mu}(x_0))^2\right] \\
&= \mathbb{E}\left[((\mu(x_0) + \varepsilon) - \hat{\mu}(x_0))^2\right] \\
&= \mathbb{E}\left[((\mu(x_0) - \hat{\mu}(x_0)) + \varepsilon)^2\right] \\
&= \mathbb{E}\left[(\mu(x_0) - \hat{\mu}(x_0))^2\right] + 2\,\mathbb{E}\left[(\mu(x_0) - \hat{\mu}(x_0))\,\varepsilon\right] + \mathbb{E}[\varepsilon^2]
\end{aligned}$$

Since ε has mean 0 and is independent of the training data $\mathcal{T}$ (and thus independent of $\hat{\mu}$), the cross-product term vanishes:

$$2\,\mathbb{E}\left[\mu(x_0) - \hat{\mu}(x_0)\right] \cdot \mathbb{E}[\varepsilon] = 0$$

Because $\mathbb{E}[\varepsilon^2] = \text{Var}(\varepsilon) = \sigma^2$, we arrive at:

$$\text{Err}(x_0) = \mathbb{E}_{\mathcal{T}}\left[(\mu(x_0) - \hat{\mu}(x_0))^2\right] + \sigma^2$$

Deriving the Decomposition: Step 2

Now we look closer at the first term: $\mathbb{E}_{\mathcal{T}}[(\mu(x_0) - \hat{\mu}(x_0))^2]$. We add and subtract the average model prediction $\mathbb{E}_{\mathcal{T}}[\hat{\mu}(x_0)]$ inside the expression:

$$\mathbb{E}_{\mathcal{T}}\left[(\mu(x_0) - \hat{\mu}(x_0))^2\right] = \mathbb{E}_{\mathcal{T}}\left[((\mu(x_0) - \mathbb{E}[\hat{\mu}(x_0)]) + (\mathbb{E}[\hat{\mu}(x_0)] - \hat{\mu}(x_0)))^2\right]$$

Expanding this quadratic form yields:

$$\begin{aligned} &= \mathbb{E}_{\mathcal{T}}\left[(\mu(x_0) - \mathbb{E}[\hat{\mu}(x_0)])^2\right] \quad \leftarrow \text{Deterministic value, drops out of expectation} \\ &\quad + 2\,\mathbb{E}_{\mathcal{T}}\left[(\mu(x_0) - \mathbb{E}[\hat{\mu}(x_0)])\,(\mathbb{E}[\hat{\mu}(x_0)] - \hat{\mu}(x_0))\right] \\ &\quad + \mathbb{E}_{\mathcal{T}}\left[(\mathbb{E}[\hat{\mu}(x_0)] - \hat{\mu}(x_0))^2\right] \end{aligned}$$

The middle cross-product term is zero because $\mathbb{E}_{\mathcal{T}}\left[\mathbb{E}[\hat{\mu}(x_0)] - \hat{\mu}(x_0)\right] = \mathbb{E}[\hat{\mu}(x_0)] - \mathbb{E}[\hat{\mu}(x_0)] = 0$.

The Famous Trio

By combining both steps, we arrive at the classic ESL decomposition:

The Bias-Variance-Noise Decomposition

$$\text{Err}(x_0) = (\mu(x_0) - \mathbb{E}[\hat{\mu}(x_0)])^2 + \mathbb{E}\left[(\hat{\mu}(x_0) - \mathbb{E}[\hat{\mu}(x_0)])^2\right] + \sigma^2$$

$$\text{Err}(x_0) = \textbf{Bias}^2\left(\hat{\mu}(x_0)\right) + \textbf{Variance}\left(\hat{\mu}(x_0)\right) + \textbf{Irreducible Error}$$

- **Bias**^{2}: Measures how far the average prediction of our model over all datasets is from the true functional value.
- **Variance**: Measures how much the model's predictions fluctuate around its own mean when trained on different datasets.
- **Irreducible Error (σ^2)**: The inherent noise in the system. No matter how perfect our model is, we cannot predict below this limit.

The Trade-Off Model Complexity

As described by Hastie et al., managing error is a balancing act controlled by model complexity:

Low Complexity / High Bias

- Simple models (e.g., linear regression).
- Cannot capture complex true functions $\mu(X)$.
- Low variance across training sets, but high systematic error (underfitting).

High Complexity / High Variance

- Flexible models (e.g., high-degree polynomials, deep trees).
- Easily fit the idiosyncrasies of a specific training dataset $\mathcal{T}$.
- Low bias, but highly unstable predictions across datasets (overfitting).

Goal: Find the sweet spot that minimizes the sum of Bias^2 + Variance.

An Instructive Example: k-Nearest Neighbors (k-NN)

To see the trade-off analytically, consider a k-NN classifier where $\hat{\mu}(x_0) = \frac{1}{k}\sum_{x_i \in N_k(x_0)} y_i$. Assuming for simplicity that the points in the neighborhood are close enough so $f(x_i) \approx \mu(x_0)$:

- **Expected Prediction Error:**

$$\text{Err}(x_0) \approx \left(\mu(x_0) - \frac{1}{k}\sum_{i=1}^{k} f(x_i)\right)^2 + \frac{\sigma^2}{k} + \sigma^2$$

- **When k is small (High Complexity):**
  - Neighborhood is tiny; $\frac{1}{k}\sum f(x_i) \approx \mu(x_0) \implies$ **Bias**$^2 \to 0$.
  - Variance term $\frac{\sigma^2}{k}$ is large because we average over very few points.
- **When k is large (Low Complexity):**
  - Variance $\frac{\sigma^2}{k} \to 0$ due to extensive averaging.
  - The neighborhood expands to distant points, so $\frac{1}{k}\sum f(x_i)$ diverges from $\mu(x_0) \implies$ **Bias**^{2} grows.

Summary Takeaways

- **Prediction Error** is explicitly broken down into structural assumptions (Bias^2), estimator stability (Variance), and nature's random variation (Noise).
- Minimizing training error alone does not yield a good model; it typically blows up the Variance component on unseen testing data.
- Regularization techniques (like Ridge, Lasso, or tree-pruning) deliberately introduce a small amount of bias in exchange for a massive reduction in variance.

This is a preliminary version of slides