Decomposing MSE as Noise + Squared Bias + Variance

Suppose we're trying to predict the response, Y, at for some test input, x. We measure how good our guess, $\hat{Y}$, is by *Mean Squared Error (MSE)*, defined as

$$\text{MSE} \;=\; E[(\hat{Y} - Y)^2]$$

$\hat{Y}$ varies randomly according to the random responses in the training data. Here, we treat the training inputs as fixed (though the book sometimes doesn't). Y is also subject to noise, assumed to be independent of the noise in the training data.

We can decompose $\text{MSE}(\hat{Y})$ as follows:

$$\begin{aligned}
\text{MSE} &= E[\hat{Y}^2] \;+\; E[Y^2] \;-\; E[2Y\hat{Y}] \\
&= \text{Var}(\hat{Y}) \;+\; E[\hat{Y}]^2 \;+\; \text{Var}(Y) \;+\; E[Y]^2 \;-\; 2E[Y]E[\hat{Y}] \\
&= \text{Var}(Y) \;+\; (E[\hat{Y}] - E[Y])^2 \;+\; \text{Var}(\hat{Y})
\end{aligned}$$

The three terms have these meanings:

$\text{Var}(Y)$ is the inherent uncertainty from noise

$(E[\hat{Y}] - E[Y])^2$ is the squared bias of $\hat{Y}$ as a predictor of Y

$\text{Var}(\hat{Y})$ is the variance of $\hat{Y}$ due to random training data

The Bias–Variance Tradeoff

We've decomposed MSE for predicting Y as $\text{Var}(Y) + \text{Bias}(\hat{Y})^2 + \text{Var}(\hat{Y})$.

We have no control over $\text{Var}(Y)$, so we might as well view the problem as estimating $E(Y)$, for which MSE $= \text{Bias}(\hat{Y})^2 + \text{Var}(\hat{Y})$. (The text calls MSE for Y with noise included the *Expected Prediction Error (EPE)*.)

Idea for finding a good predictor: Choose a method for which $\hat{Y}$ has small bias and small variance. If can't have both, make an optimal *tradeoff* between bias and variance to minimize MSE.

Problem with this idea: The bias and the variance depend on the true relationship of the response to the inputs, which we don't know. (That's what we're trying to learn.) So how can we find the optimal tradeoff?

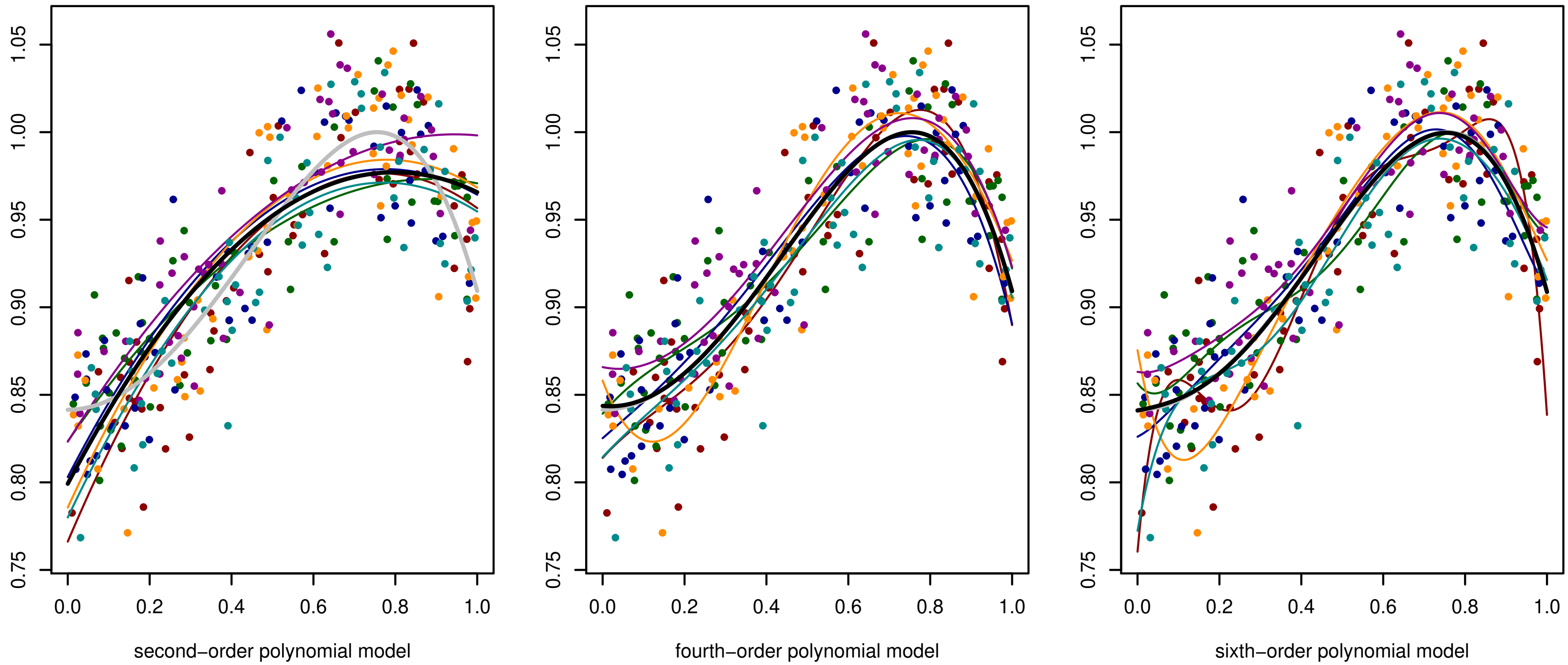
Solutions: Empirical methods like *cross-validation* try to estimate the tradeoff. The theory of *VC dimension* gives bounds on variance that are maybe helpful. Or maybe we should try some other idea.

Bias and Variance for Polynomial Models

Recall last week's example using polynomial models of the form

$$y \;=\; \beta_0 \;+\; \beta_1 x \;+\; \cdots \;+\; \beta_p x^p \;+\; \text{noise}$$

Here are six curves (different colours) fitted to six data sets of 50 points, for $p = 2, 4, 6$. Also shown is the average of 300 such curves (black), and the true noise-free curve (gray), with $y = \sin(1 + x^2)$.



Bias and Variance of Nearest Neighbor Methods

Suppose we use the k-NN method on data where $y = f(x) + \text{noise}$, for some function $f(x)$. Let's assume that the noise has variance σ^2 for all values of x.

What will be the variance of $\hat{Y}(x)$ for some test point, x? Recall that we computed this as

$$\hat{Y}(x) \;=\; \frac{1}{k} \sum_{i \in N_k(x)} y_i$$

Let's regard the test responses, y_i, as varying randomly, but consider the training inputs, x_i, to be fixed. So the k nearest neighbors of x are also fixed, and

$$\text{Var}(\hat{Y}(x)) \;=\; \frac{1}{k^2} \sum_{i \in N_k(x)} \text{Var}(y_i) \;=\; \sigma^2/k$$

So larger k gives smaller variance. What about bias?

$$E(\hat{Y}(x)) \;=\; \frac{1}{k} \sum_{i \in N_k(x)} E(y_i) \;=\; \frac{1}{k} \sum_{i \in N_k(x)} f(x_i)$$

With bigger k, we average $f(x_i)$ for x_i that are farther from x. Unless $f(x)$ is constant, or we are lucky, this average will be farther from $f(x)$ when k is big.

Adjusting Bias and Variance by Varying "Complexity"

For both polynomial models and k-NN methods, we see that we can *reduce bias* by increasing "complexity" (large p or small k), but this *increaes variance.*

See Figures 2.3, 2.4, and 2.5 in the text for an illustration for k-NN classification.

Some moderate level of complexity will give the optimal tradeoff, though this level may be hard to judge. See Figure 2.11 in the text.

We can also try to decrease *both* bias and variance by finding a better method — eg, by combining neighbor-based methods with linear models.

This is a *learning machine* approach — we look at how the machine is working, and try to make a better machine.

The alternative is the *probabilistic modeling* approach — we try to build a model that expresses our knowledge of the problem. If we build a good model, it should perform well, even if we never think about "bias" and "variance".

The Effect of Dimensionality on Nearest Neighbor Methods

What happens to bias and variance of k-NN as the number of inputs increases?

The variance isn't affected, if we regard the inputs as fixed (assuming the noise level stays the same). If the inputs are seen as random, then the variance does increase with dimension.

With inputs fixed, the bias is different for different test inputs — smaller for test inputs that happen to be close to training inputs. We might take an average over the input space, or regard the training inputs as random.

Either way, the bias will tend to get bigger when there are more inputs, since the k nearest neighbors will be farther away (on average) — see Figure 2.7 in the text.

Actually, the bias will depend on the true function, so it's hard to say anything precise in general. But if we assume that Y depends only on the first input, X_1, then introducing extra inputs, $X_2, \ldots, X_p$ will certainly make things worse.

The Effect of Dimensionality on Linear Regression

What happens to MSE for a linear regression model as the number of inputs increases? Assume that only the untransformed inputs are used, so the model is

$$Y \;=\; \beta_0 \,+\, \beta_1 X_1 \,+\, \cdots \,+\, \beta_p X_p \,+\, \text{noise}$$

What happens in general depends on the true values of the β_i and on the correlation of the X_i. But suppose that $\beta_i = 0$ for $i > 1$ (so y depends only on X_1), and that the X_i are uncorrelated.

Result: As we'll discuss later, the estimates $\hat{\beta}_i$ are independent when the X_i are uncorrelated. Our predictions will have the form

$$\hat{Y}(x) \;=\; \hat{\beta}_0 \,+\, \hat{\beta}_1 x_1 \,+\, \cdots \,+\, \hat{\beta}_p X_p$$

With a reasonable-size training set, we may get good estimates of β_0 and β_1, but our predictions will also contain the terms

$$\hat{\beta}_2 x_2 \,+\, \cdots \,+\, \hat{\beta}_p X_p$$

These terms will be independent, and all will be "noise" that just degrade our predictions. The more such terms there are, the worse our predictions will be.

What Can We Do?

So what can we do to avoid this "curse of dimensionality"?

One approach: Treat extra inputs as extra "complexity", and try to trade off bias and variance. Various "variable selection" methods are of this sort.

Another approach: Maybe there are better learning methods that aren't so badly affected by dimensionality.

The modelling approach: Rather than being bad, extra inputs should be good! (They give us more information.) But we need to define a suitable model that incorporates our information about which inputs are most likely to be relevant.