2.2 Assessing Model Accuracy

October 17, 2014

Measuring Quality of Fit

- How well do our predictions match what is truly observed?
  - Specifically does the outcome we see in the training data match what we see in unseen test data?

- Example: We want to predict how many influenza admissions to the emergency department for this flu season.
  - Prediction model will use training data from prior influenza seasons
  - But will it work for the current influenza season? And how well?
    - Could we test the model on the first month in flu season then adjust as needed?

Mean Squared Error (MSE)

- One method of determining the best model to use.
  - *Most often used in regression settings*

- Goal: Choose a model with the lowest test MSE

  Better than the model with the lowest training MSE

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2,$$

$\hat{f}(x_i)$ = Prediction that $\hat{f}$ gives for the i^{th} observation

In other words, does our prediction from training data work for new and unseen data?

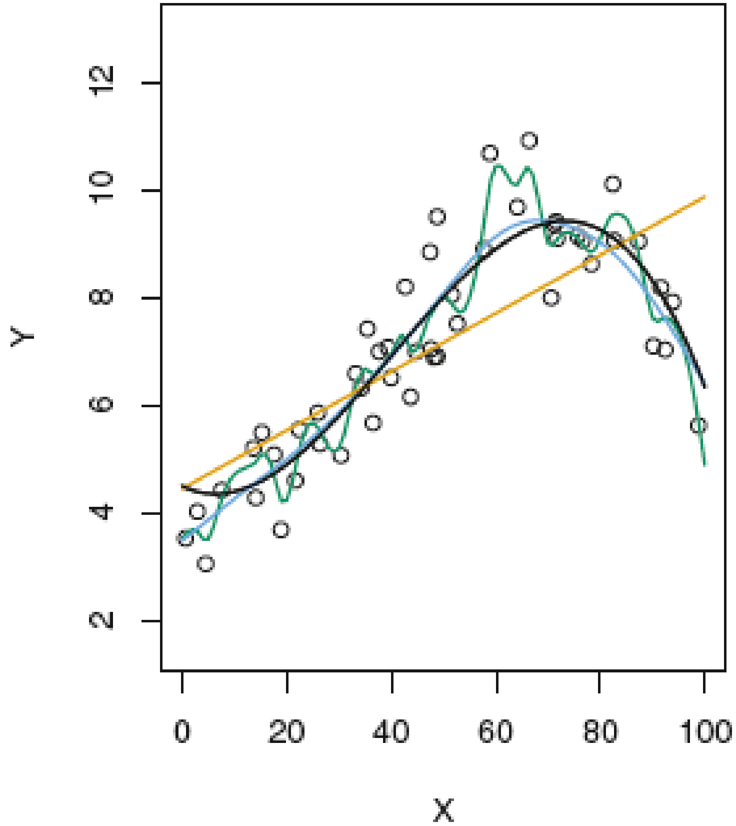
MSE, continued…

- Two different MSE: training and test
  - Training: Data we already have
  - Test: Future data (data we can use to check our training results)
- #1 Goal is to have a low MSE in our test data
  - Training MSE will decline as model flexibility increases

- So…how does this all work?

MSE Example: data first!

Black Curve = True F



O = Test data

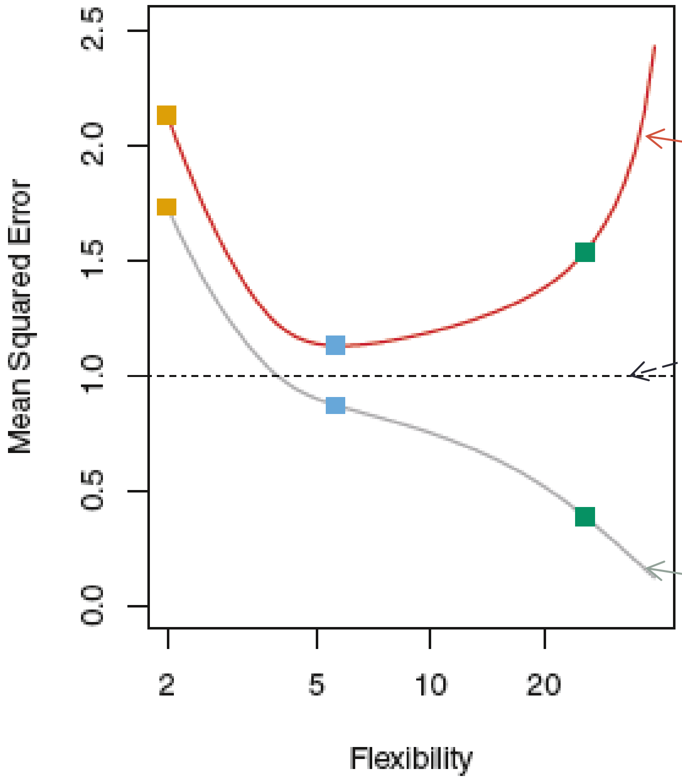
Model Flexibility

Green Curve = Smoothing Spline

Blue Curve = Smoothing Spline

Orange Curve = Linear Regression Fit

MSE Example

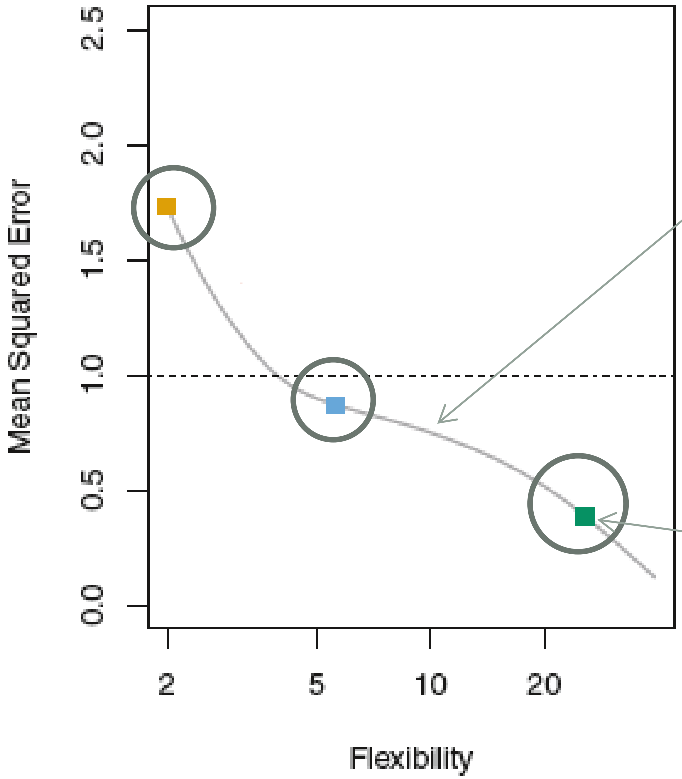


Red Curve = Test MSE

Dashed Line =
Minimum possible MSE
Var(ε) – irreducable error

Grey Curve = Training MSE

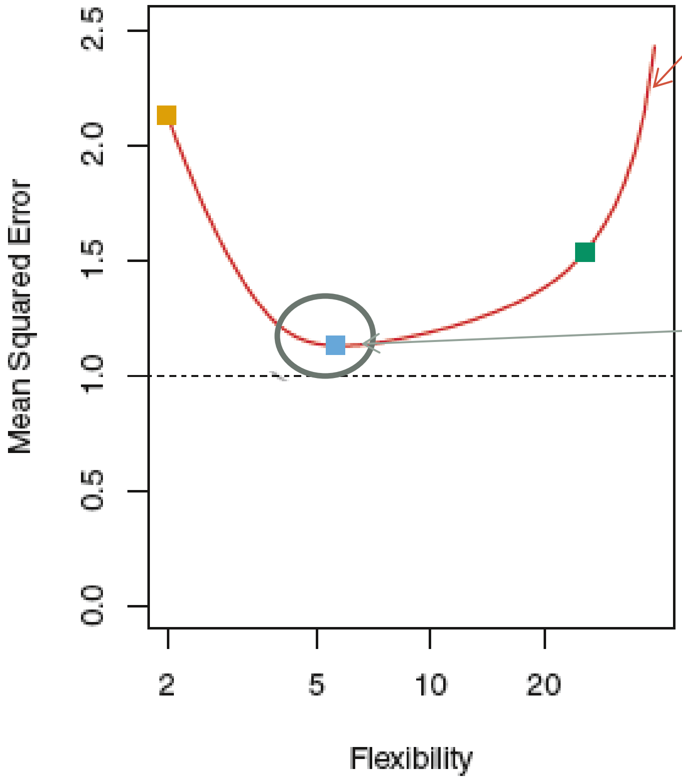
MSE Example: Training MSE



Grey Curve = Training MSE

Lowest MSE

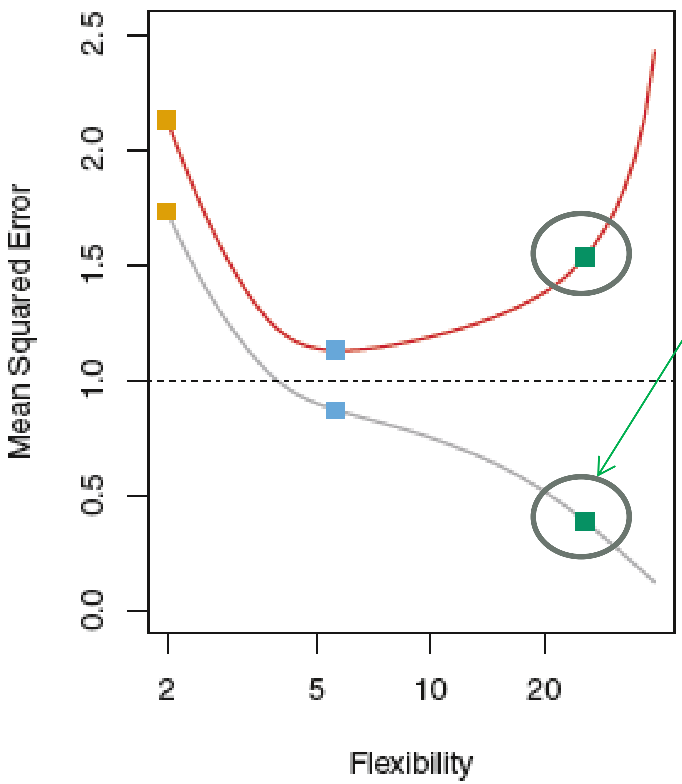
MSE Example: Test MSE



Red Curve = Test MSE

So now we see that when using our three prediction methodologies our BLUE (smoothing spline) model has the lowest MSE.

MSE Example: Overfitting the data



Using the training data, the green model had the lowest MSE but this was not true for the test data.

This happens because the model is working too hard to fit the training data.

We are picking up patterns due to random chance instead of the true pattern of *f*

MSE: Caveat

- We do not usually have “test” data.
  - We typically have training data and then have to use our model to predict outcomes
- In future chapters we will learn how to estimate training MSE

Bias-Variance Trade Off

$$E\left(y_0 - \hat{f}(x_0)\right)^2 = \mathrm{Var}(\hat{f}(x_0)) + [\mathrm{Bias}(\hat{f}(x_0))]^2 + \mathrm{Var}(\epsilon).$$

Expected test MSE = Variance of $\hat{f}$ + Variance of bias + Variance of error terms

The expected test MSE = average of repeated estimates of *f* using many large training datasets

Bias-Variance Trade Off

- Want a model with low variance and low bias
  - Variance = amount which $\hat{f}$ would change if we used a different training dataset

$\uparrow$ Model Flexibility = $\uparrow$ Variance

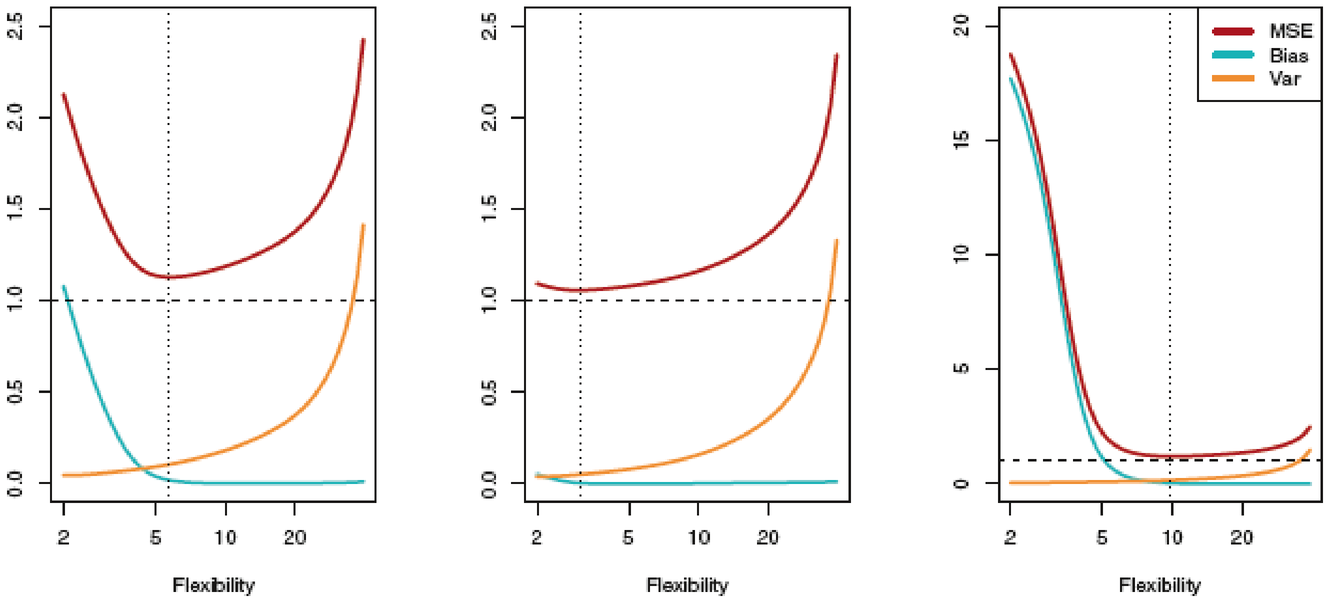
Bias-Variance Trade Off

- Want a model with low variance and low bias
  - Bias = error that is introduced by trying to look at a “real-life” problem

↑ Model Flexibility = ↓ Bias

Bias-Variance Trade Off

- When using training data it is important to find a method with **low variance and low squared bias**



Remember: just because a model eliminates bias, doesn't mean it will perform better than a simpler model

Bias-Variance Trade Off

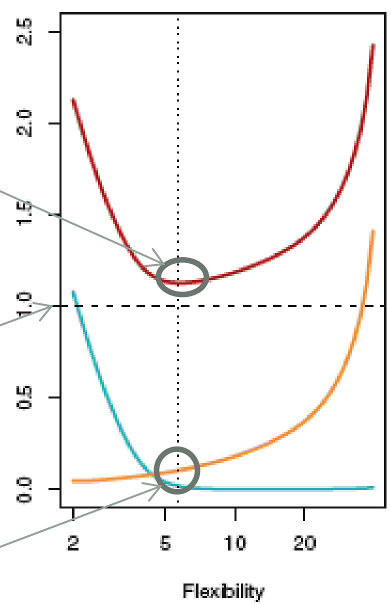
This is the bias and variance from our example from earlier.
For different levels of flexibility, we see a change in:

Squared bias (blue)
Variance (orange)
MSE (red)

Smallest Test MSE

Irreducable error

Ideal flexibility based on the MSE



As flexibility increases:

-Blue Curve = Bias initially decreases rapidly

-Red Curve = Test MSE sharp decline then increase

Orange Curve = Variance rises slow then steep incline

The Classification Setting

- Used when y_i is no longer numerical
  - What if the training observations are qualitative?
- We then need to find the training error rate (fraction of incorrect classifications)

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i).$$

The Bayes Classifier

- Error rates from the classification setting can be minimized by using a probability

$$\Pr(Y = j | X = x_0)$$

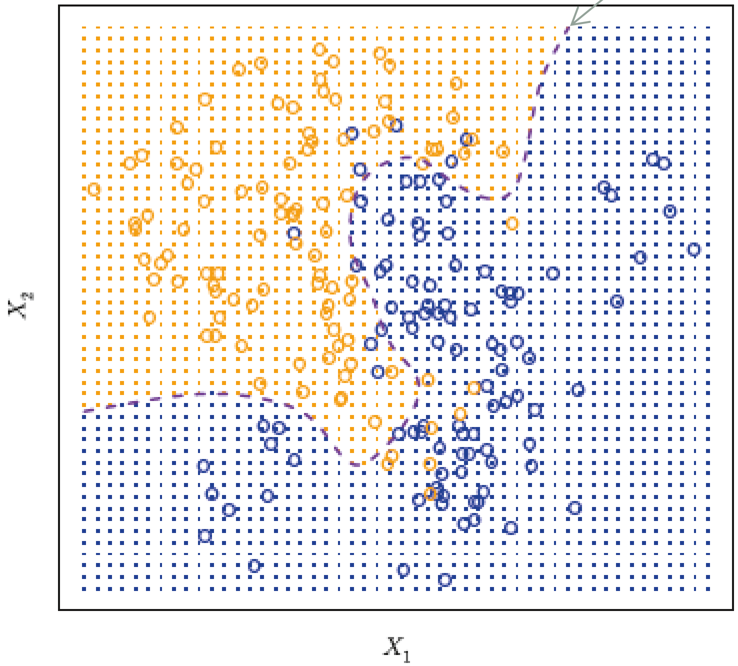
- We will assign each test observation to the MOST LIKELY class given its predictive values (from the training data)
- This is a conditional probability

<u>Class assignments will be made based on whether Pr>0.5 or Pr<0.5</u>

The Bayes Classifier

<u>Bayes Decision Boundary:</u>
Pr=0.5

Orange =
Pr(Y=orange|x)>0.5



Blue =
Pr(Y=blue|x)<0.5

The Bayes Classifier

Requires the conditional distribution of Y given X, which is something we do not know

It is therefore an:

Impossible Gold Standard

K-Nearest Neighbor

- Attempts to estimate the conditional distribution of Y given X with an estimated probability
  - Use this because we cannot use Bayes

K-Nearest Neighbor

- We want to make a prediction about x

K = 3 for all values of X_1 and X_2

Black Line = KNN decision boundary

Since K=3 the three closest points to X are chosen

Therefore, the test observation is predicted to the most commonly occurring class: BLUE