Lecture 11: Introduction to Statistical Modeling

UNIVERSITY OF
SAN FRANCISCO

James D. Wilson

BSDS 100 - Intro to Data Science with R

Outline

- Model Building Basics
- Prediction versus Inference
- Model Choice
- Assessing Model Accuracy
- Training vs. Test Set Evaluation
- Bias-Variance Trade-off

Reference: Chapters 22 - 25 in *R for Data Science* book; Chapter 2 in *Intro to Statistical Learning*

Setting

Given:

- Response $y = (y_1, \ldots, y_n)^T$ – *continuous-valued*
- Design matrix / data $X \in \mathbb{R}^{n \times p}$

Aim: Estimate a function f that best represents the relationship between X and y:

$$y = f(X) + \epsilon$$

Important Questions:

- How do we *choose* and *estimate* f?
- How do we *assess* our model choice?
- Are we concerned with *inference* or *prediction*?

No Free Lunch Principle

There are *many* models and methods to choose from in regression (and classification / clustering for that matter).

No Free Lunch Principle: There is no *one* method that dominates all others over all possible data sets.

Focus of this course: introduce a wide array of methods for a variety of problems.

Motto: "All models are wrong but some are useful" - George Box

Example: Multiple Linear Regression

A parametric model that supposes a linear relationship between y and data observations X:

$$y = \underbrace{X\beta}_{f(X)} + \epsilon$$

where $\epsilon = (\epsilon_1, \ldots, \epsilon_n)$ is assumed to satisfy either

1. $\mathbb{E}[\epsilon_i] = 0$, $\text{Var}(\epsilon_i) = \sigma^2$, $\mathbb{E}[\epsilon_i \epsilon_j] = 0$ for all $i \neq j$, or
2. $\epsilon_i \overset{i.i.d}{\sim} N(0, \sigma^2)$

Prediction vs. Inference

When choosing a model f, we are usually concerned with one of two primary goals: prediction or inference. It *is* possible to choose a model that is reasonably well-calibrated for both prediction and inference.

Prediction

Main Objective: Predicting new Y using $\hat{Y} = \hat{f}(X)$

Model Choice: models that have the highest prediction performance. Often black box methods, which have no concern with the exact form of $\hat{f}$.

Prediction vs. Inference

Inference

Main Objective: Understand the relationship between Y and X:

- Variable Selection: what predictors are most associated with the response
- Focus is functional relationship of Y and X
- Strive for parsimony! *KISS: Keep It Simple Stupid!*

Model Choice: models that have high interpretability. Often parametric methods, which explicitly dictate the form of $\hat{f}$ via parameters.

Model Choice: Parametric vs. Non-Parametric

Parametric Methods

- Makes an assumption about the functional form of f
- Identifyng f reduces to the estimation of a set of parameters
- Often simpler than estimating the entire function (that's the aim anyway)
- Caution: Can result in overly-simplistic models
- **Examples**: linear regression, logistic regression

Model Choice: Parametric vs. Non-Parametric

Non-parametric Methods

- Not restricted to assumptions about the functional form of f
- Can accurately fit a wider range of possible shapes / forms for f
- Often requires a very large number of observations
- Caution: Can quickly over-fit data!
- **Examples**: polynomial splines, smoothing splines

Assessing Model Accuracy

Question: How well does your method perform on *new* data, i.e., data you have *not* seen during learning?

Example: You get a new data instance:

$$X_{new} = (\text{No history of cancer, smoker, male})$$

Can you assess the goodness of your throat cancer predictor?

Training vs. Test Set Evaluation

1. Divide the data (X, y) into two sets:
   - Training set: (X_{train}, y_{train})
   - Test set: (X_{test}, y_{test})
2. Use training set to produce a predictor $\widehat{f}()$ via

$$y_{train} = f(X_{train}) + \epsilon$$

3. Use test set to evaluate performance of predictor:

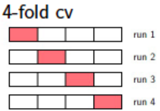
$$\widehat{y}_{test} = \widehat{f}(X_{test})$$

Assess difference between $\widehat{y}_{test}$ and y_{test}

Choosing Training and Test Sets

1. **Random sampling**: choose a test set at random from data and test *all* models on the same set

2. **k-fold cross validation**: split data into k subsets. In turn treat each subset as held-out and train on the remaining. Performance is evaluated as average performance of each of k test sets.



3. **Leave-one-out cross validation**: special case of k-fold cross validation where $k = n$, and test sets are of size 1.

Training and Test Sets: Important considerations

1. Never let information from the test set make its way into the training data! This is the #1 most common mistake in model assessment.

2. Difference between *prediction* and *estimation* error:
   - Prediction Error: error associated with predictions on the *test* set

$$y_{test} \quad \text{vs.} \quad \hat{y}_{test}$$

   - Estimation Error: error associated with estimates in the *training* set

$$y_{train} \quad \text{vs.} \quad \hat{y}_{train}$$

Training and Test Sets: Important considerations

1. By splitting data in cross validation, the variance of the estimated regression coefficients can increase if the data set is not large.

2. Once a model has been validated and compared against other potential models, we typically use the entire data set for estimating the final regression model.

3. Sometimes the training and test sets are chosen in a systematic way (e.g., up-sampling and down-sampling) so as to avoid bias in the analysis. We'll come back to this in classification.

Assessing Model Accuracy

Mean squared error

The mean squared error (MSE) of a model f is given by:

$$MSE(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2$$

The MSE measures acts as a yardstick for model assessment for continuous data.

Note: In prediction, the mean squared difference between y_{new} and $\hat{f}(\mathbf{x}_{new})$ is known as the mean square prediction error (MSPE).

Note: There are many choices for measuring accuracy. The choice depends on the possible values of y.

Assessing Model Accuracy

Let Ω = index (which rows of X) that represent the training set

Let $\Theta = \Omega^c$ = index that represent the test set

General Approach:

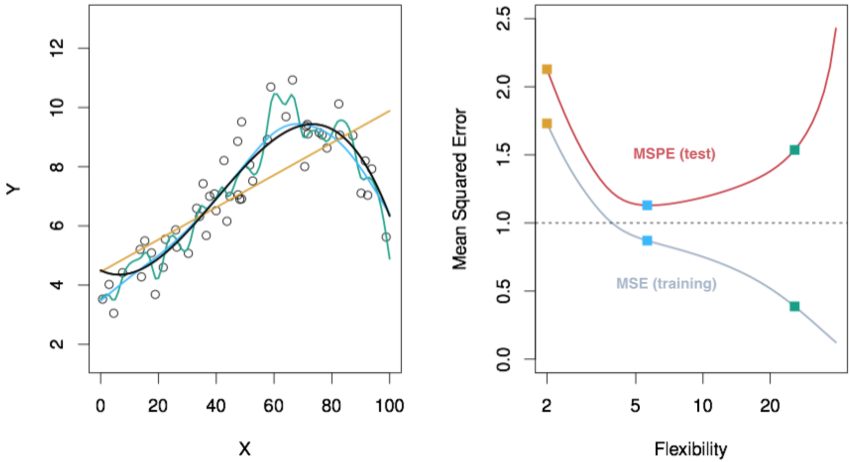
1. Estimate model $\widehat{f}$:

$$\hat{f} = \text{argmin}_f \left(\frac{1}{|\Omega|} \sum_{j \in \Omega} (y_j - f(\mathbf{x}_j))^2 \right) = \text{argmin}_f \big(MSE(f) \big)$$

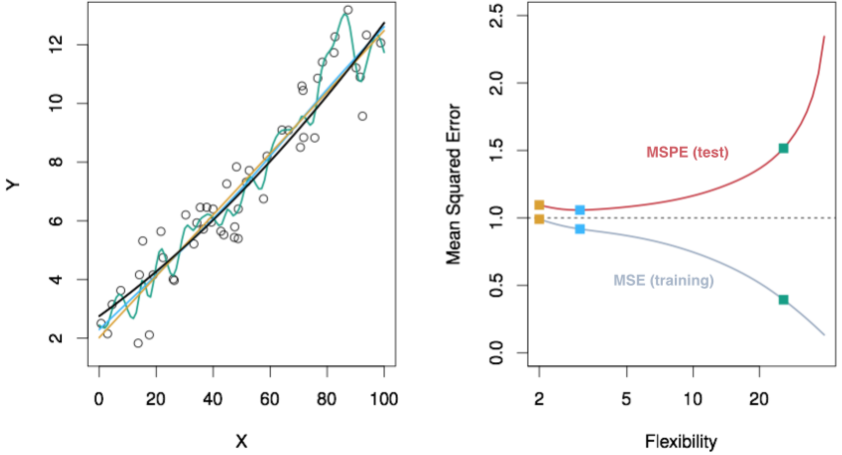
2. Evaluate model on test set:

$$MSPE(\hat{f}) = \frac{1}{|\Theta|} \sum_{j \in \Theta} (y_j - \hat{f}(\mathbf{x}_j))^2$$

MSE vs. MSPE

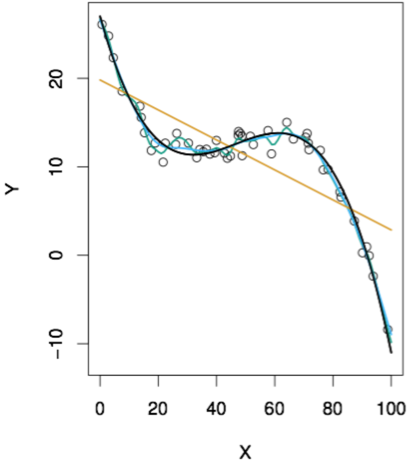


MSE vs. MSPE

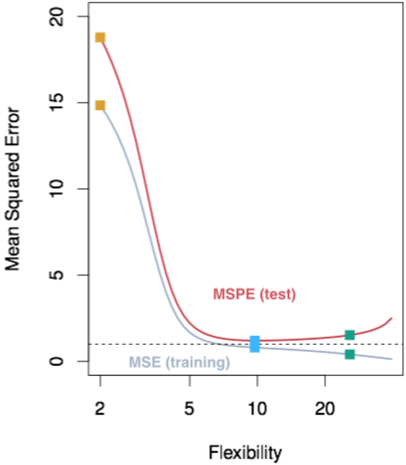


Y
12
10
8
6
4
2
0
20
40
60
80
100
X
Mean Squared Error
2.5
2.0
1.5
1.0
0.5
0.0
MSPE (test)
MSE (training)
2
5
10
20
Flexibility

MSE vs. MSPE



Y
20
10
0
−10
0
20
40
60
80
100
X



Mean Squared Error
20
15
10
5
0
MSPE (test)
MSE (training)
2
5
10
20
Flexibility

MSE vs. MSPE: Noticeable Trends

- **Fact**: $\mathbb{E}[\text{MSPE}(f(X_{test}))] \geq \mathbb{E}[\text{MSE}(f(X_{train}))]$
- **Trend 1:** After a certain point in complexity (the blue boxes in the previous plots), there is an inverse relationship between MSE and MPSE. We say that a model is overfitting the data when we are in this range of complexity!
- **Trend 2:** The MSE tends to decrease as complexity increases.

Bias-Variance Trade-off

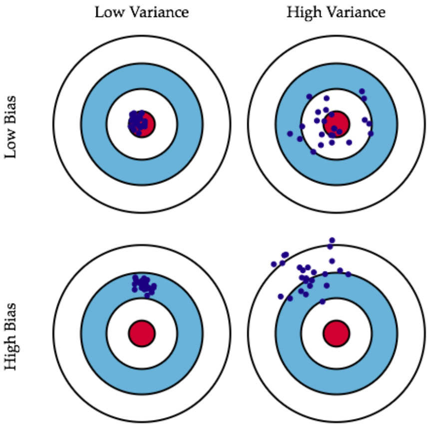
An important means of understanding $\text{MPSE}(\hat{f}) = \frac{1}{|\Theta|}\sum_{j\in\Theta}(y_j - \hat{f}(\mathbf{x}_j))^2$ comes from the following decomposition for new data (X_0, y_o).

$$\begin{aligned}
\mathbb{E}[MPSE(\hat{f})] &= \mathbb{E}[(y_o - \hat{f}(X_o)]^2 + \text{Var}(\hat{f}(X_o)) + \text{Var}(\epsilon) \\
&= Bias(\hat{f}(X_o))^2 + \text{Var}(\hat{f}(X_o)) + \text{Var}(\epsilon)
\end{aligned}$$

Result: the expected MSPE of a model is a function of the bias and variance of $\hat{f}$, as well as the variance of the error term ϵ.

Bias-Variance Trade-off

Example: Point estimation. What is bias and variance of an estimate?

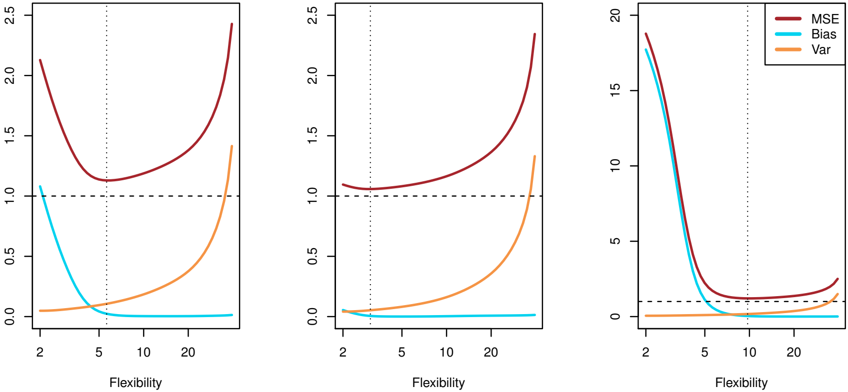


Bias-Variance Trade-off

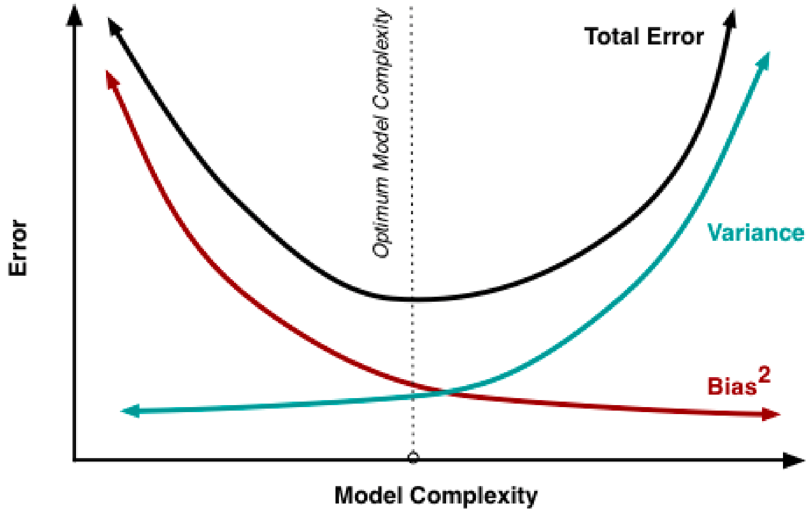
Components of MSPE

- $\text{Bias}(\hat{f}(X_o))^2$: quantifies distance between model and truth
  - Non-negative
  - Generally decreases as the model becomes more complex
- $\text{Var}(\hat{f}(X_o))$: quantifies variance of the model
  - Non-negative
  - High variance implies that the model is highly sensitive to small changes in training data
  - Generally increases as the model becomes more complex
- $\text{Var}(\epsilon)$: variance of the error terms
  - Non-negative
  - Is not affected by complexity of the model; constant value

Bias-Variance Trade-off Example



Bias-Variance Trade-off Example



Bias-Variance Trade-off

Resulting Trade-off: Since Var(ϵ) is constant, we'd like to choose a model with minimum bias and minimum variance.

Primary Issues:

1. Both the bias and variance terms are non-negative
2. The bias and variance terms are often inversely related
3. The bias and variance change at different rates

Solution: Decide what is important in application (prediction vs. interpretation) and choose model accordingly. Seek optimal model complexity if possible.

Model Building in `R`

We now go through several examples in `R` to show how to build regression models using pre-specified functions like `lm()` and `glm()`.

Go to the Statistical Modeling.Rmd file on the Github site to see these examples.

For more information on Linear Regression modeling as well as statistical modeling in general, be sure to take **MATH 372** and **MATH 373**, which are offered in the Fall and Spring, respectively.