Linear Regression - Model Comparsions

Ryan Miller

Introduction

- Most applications require the data analyst to choose between competing models
  - Is `Tip ~ TotBill` a better model than `Tip ~ Size`?
  - Is `Tip ~ TotBill + Size` better than both? How would you know?

Quantifying Model Fit

- A good model will have fitted values (predictions) that are *close* to the observed y-values
  - Thus, we might measure the *distance* between $E(y)$ and the observed values of y to objectively compare the fit of two competing models

Quantifying Model Fit

- A good model will have fitted values (predictions) that are *close* to the observed y-values
  - Thus, we might measure the *distance* between $E(y)$ and the observed values of y to objectively compare the fit of two competing models

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

- **Root mean squared error**, often abbreviated *RMSE*, is perhaps the most common measure of fit
  - Models with more accurate predictions (lower average values of $y_i - \hat{y}_i$) will have smaller *RMSE*
  - It's logical to favor the model with the smallest *RMSE* among our candidate models

Overfitting

- Unfortunately, *in-sample RMSE* will *always* favor larger, more complex models
  - This tendency is known as **overfitting**
  - Conceptually, the issue is that complex models can become so tailored to what was observed in the *sample data* that they are no longer a good representation of the population

Overfitting

- Unfortunately, *in-sample RMSE* will *always* favor larger, more complex models
  - This tendency is known as **overfitting**
  - Conceptually, the issue is that complex models can become so tailored to what was observed in the *sample data* that they are no longer a good representation of the population
- The example on the next slide is from our lab this week, it adds complexity to the model `Tip ~ TotBill + ...` by adding variables that are just random values from a Normal distribution
  - How do you think adding purely random values as extra predictors will impact the *in-sample RMSE*?

Overfitting

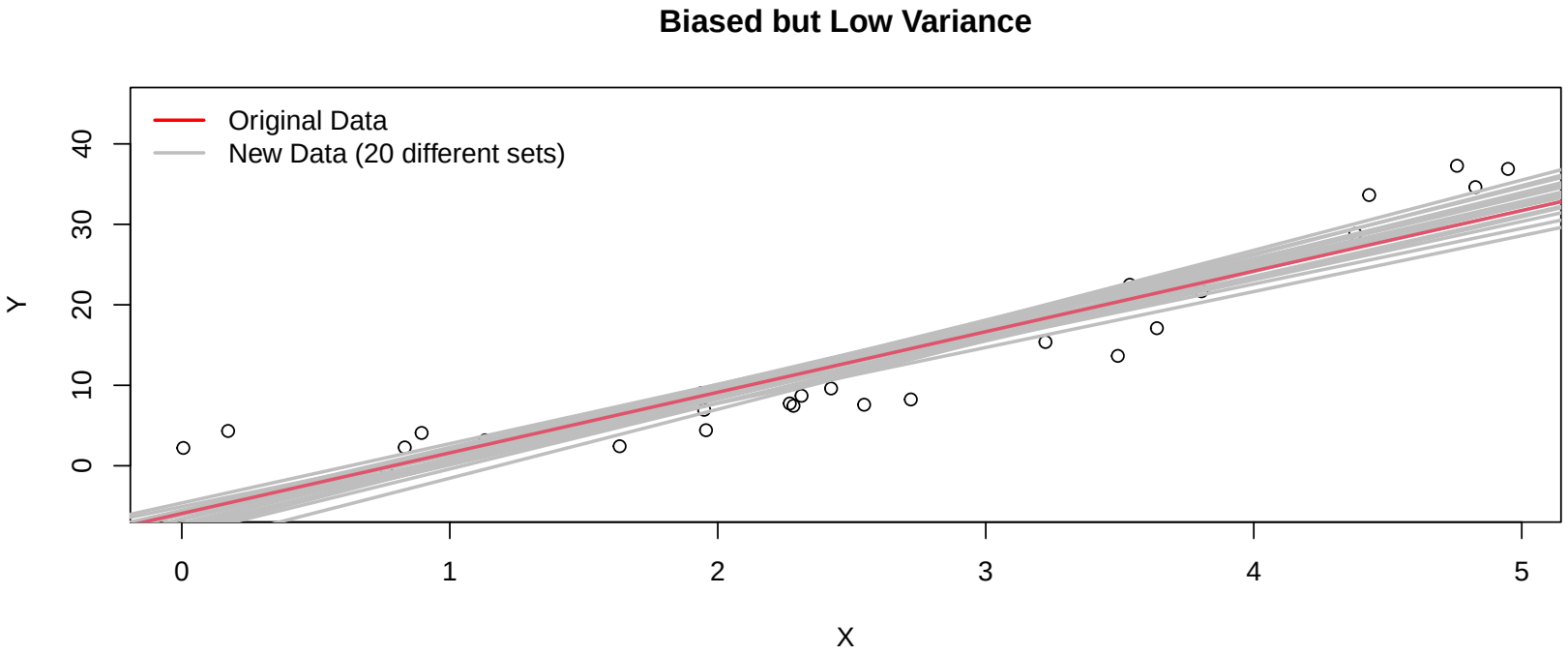
```
tips <- read.csv("https://remiller1450.github.io/data/Tips.csv")

## Add random values as three extra variables to Tips
tips$R1 <- rnorm(nrow(tips))
tips$R2 <- rnorm(nrow(tips))
tips$R3 <- rnorm(nrow(tips))

## Build bigger and bigger models using these non-sensical variables
m1 <- lm(Tip ~ TotBill, data = tips)
m2 <- lm(Tip ~ TotBill + R1, data = tips)
m3 <- lm(Tip ~ TotBill + R1 + R2, data = tips)
m4 <- lm(Tip ~ TotBill + R1 + R2 + R3, data = tips)

## Calculate RMSE for each model
rmse1 <- sqrt(1/nrow(tips)*sum((tips$Tip - m1$fitted.values)^2))
rmse2 <- sqrt(1/nrow(tips)*sum((tips$Tip - m2$fitted.values)^2))
rmse3 <- sqrt(1/nrow(tips)*sum((tips$Tip - m3$fitted.values)^2))
rmse4 <- sqrt(1/nrow(tips)*sum((tips$Tip - m4$fitted.values)^2))

## Notice how prediction errors get smaller! (try re-running multiple times)
print(c(rmse1,rmse2,rmse3,rmse4))
```

```
## [1] 1.017850 1.016369 1.016304 1.016134
```

The Bias vs. Variance Tradeoff

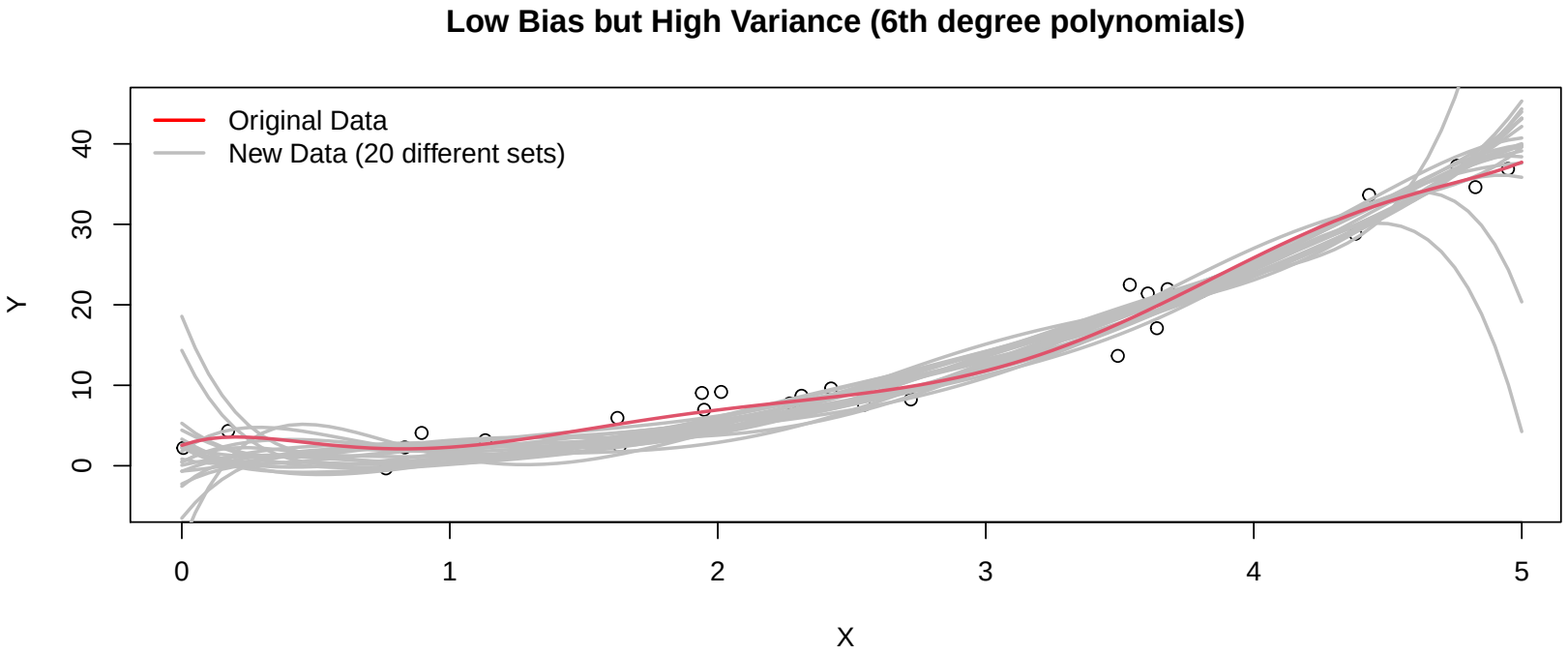
- As a model includes more complexity, it becomes less biased (think about what happens if you omit a quadratic term for a truly quadratic relationship)
  - However, additional complexity will also increase a model's variance
- If a model is too complex, it might fit the sample data very well (low bias) but it's coefficients could change dramatically if data is added or removed (high variance)

The Bias vs. Variance Tradeoff



- Simple linear regression is biased because it doesn't account for the curvature in the true relationship between X and Y
- However, it is shows low variance, fitting it to a different sample doesn't change much

The Bias vs. Variance Tradeoff



- This model is very capable of capturing the curvature in the true relationship between X and Y
- However, it contains too many parameters, it changes dramatically depending on the specific sample that it is fit to

Out-of-Sample Prediction

- Overfitting typically occurs when the same data to *train* and *test* the model
  - For linear regression, "training" refers to estimating the model's coefficients (β_0, β_1, etc.)
  - "Testing" refers to evaluating a trained model, for example by calculating $RMSE = \sqrt{\frac{1}{n}\sum_{i=1} n(y_i - \hat{y}_i)^2}$

Out-of-Sample Prediction

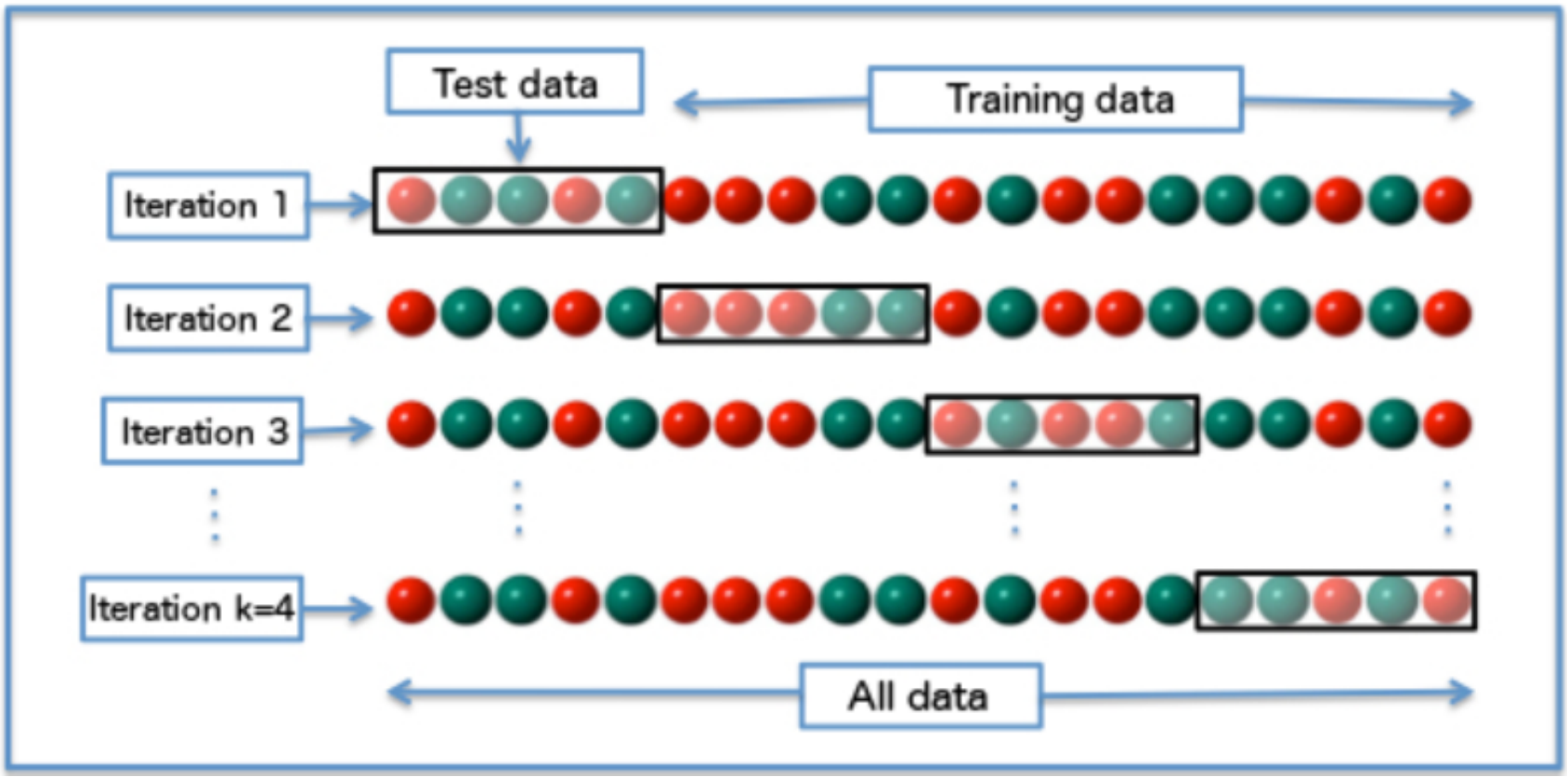
- Overfitting typically occurs when the same data to *train* and *test* the model
  - For linear regression, "training" refers to estimating the model's coefficients (β_0, β_1, etc.)
  - "Testing" refers to evaluating a trained model, for example by calculating $RMSE = \sqrt{\frac{1}{n}\sum_{i=1} n(y_i - \hat{y}_i)^2}$
- Because $\hat{\beta}_0$ and $\hat{\beta}_1$ are found my minimizing the squared residuals in the sample data, you'd expect the *RMSE* to be lower on the the training data than if the model were applied to a completely different sample from the same population
  - An unbiased assessment of a model's *RMSE* would use different data for *training* and *testing*

Cross Validation

- Cross-validation works by using different subsets of data for model *training* and *testing*
  - We will focus on **k-fold cross-validation**, which uses the following algorithm:

1. Randomly divide the original dataset into k equally sized, non-overlapping subsets
2. Fit the candidate model using data from $k-1$ folds, then find predicted values ($\hat{y}_i$'s) for k^{th} fold (the "left out" fold)
3. Repeat step two until each fold has been left out exactly once, resulting in an *out-of-sample* prediction for each observation in the dataset

Cross Validation



Cross Validation

Let's revisit the earlier example of adding three predictors that were just random values to the Tips dataset:

	Tip = TotBill	Tip = TotBill + R1 + R2 + R3
out-of-sample	1.027	1.038
in-sample	1.018	1.013

- Notice how using random values as predictors lowers the *in-sample RMSE*, but raises the *cross-validated RMSE*

Other Model Comparison Tools

- Cross-validation is extremely general, so it's fast becoming the most widely used method of comparing competing models
- In the context of linear regression, here are a few other popular tools (we'll cover these in greater detail later on):
  - **Adjusted** R^2 - A modified version of R^2 that adjusts for the number of parameters in the model. Higher values of Adjusted R^2 indicate better models.
  - **AIC** (Akaike Information Criteria) - a numeric "score" that uses a model's goodness of fit (log-likelihood) and a penalty for the number of parameters. Lower values of *AIC* indicate superior models.
  - **BIC** (Bayesian Information Criteria) - a numeric "score" that uses a model's goodness of fit (log-likelihood) and a penalty for the number of parameters that also *incorporates sample size*. Lower values of *BIC* indicate superior models.

Closing Remarks

- Models should be chosen on the basis of *out-of-sample* performance, not *in-sample* performance
  - Cross-validation provides a general method of estimating *out-of-sample* performance that can be applied to nearly any situation

Closing Remarks

- Models should be chosen on the basis of *out-of-sample* performance, not *in-sample* performance
  - Cross-validation provides a general method of estimating *out-of-sample* performance that can be applied to nearly any situation
- Other methods like Adjusted R^2, AIC, and BIC will approximately track *out-of-sample* performance
  - AIC and BIC are popular among statisticians, they can be used to compare any *likelihood-based models*
  - Adjusted R^2 is popular in applied fields, it's thought to be more easily interpreted