Linear Models for Statistical Learning, Regression

David Dalpiaz

STAT 430, Fall 2017

Announcements

- Homework 01 due today.
- Homework 02 released later today. (Hopefully.)

Statistical Learning

- Supervised Learning
  - **Regression**
  - Classification
- Unsupervised Learning

Regression Setup

$$Y = f(x_1, x_2, x_3, \ldots x_p) + \epsilon$$

$$\text{numeric response} = \text{signal} + \text{noise}$$

- Want to learn the signal
- Want to be very careful not to "learn noise"

Using a Linear Model

Setup:

$$Y = f(x_1, x_2, x_3, \ldots x_p) + \epsilon$$

Assume:

$$f(x_1, x_2, x_3, \ldots x_p) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p$$

The Linear Model

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon, \quad \epsilon \sim N(0, \sigma^2)$$

$$Y \mid X \sim N(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p, \sigma^2)$$

There are a total of $p + 2$ **parameters** in this model

- The $p + 1$ β parameters, or coefficients, control the *signal*
- The σ^2 controls the *noise*

Fitting a Linear Model

This is a **parametric** model, meaning to fit the model, we need to estimate the parameters.

For the sake of making predictions, we only need to estimate the β parameters since

$$\hat{f}(x_1, x_2, x_3, \ldots x_p) = \hat{y}(x_1, x_2, x_3, \ldots x_p) = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \ldots + \hat{\beta}_p x_p$$

Using either **least squares** or **maximum likelihood**, this becomes the same optimization problem

$$\underset{\beta_0, \beta_1, \ldots \beta_p}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - (\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}))^2$$

Estimating σ^2

While it is not needed to make predictions, to fully estimate the model, we would also need to estimate σ^2.

$$s_e^2 = \frac{1}{n-(p+1)} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{Least Squares}$$

$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad \text{MLE}$$

Both are estimates of σ^2. What is the difference?

Model "Size"

Consider two models:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$$

Which is bigger?

Model Complexity

In general, we are interested in the **complexity** or **flexibility** of a model.

For nested linear models, the more parameters, the bigger, thus, more complex.

Models that are more complex will be more **wiggly**.

Pictures of Complexity

Go to ISL Slides

Test-Train Split

We've already discussed the *Test-Train Split* and RMSE

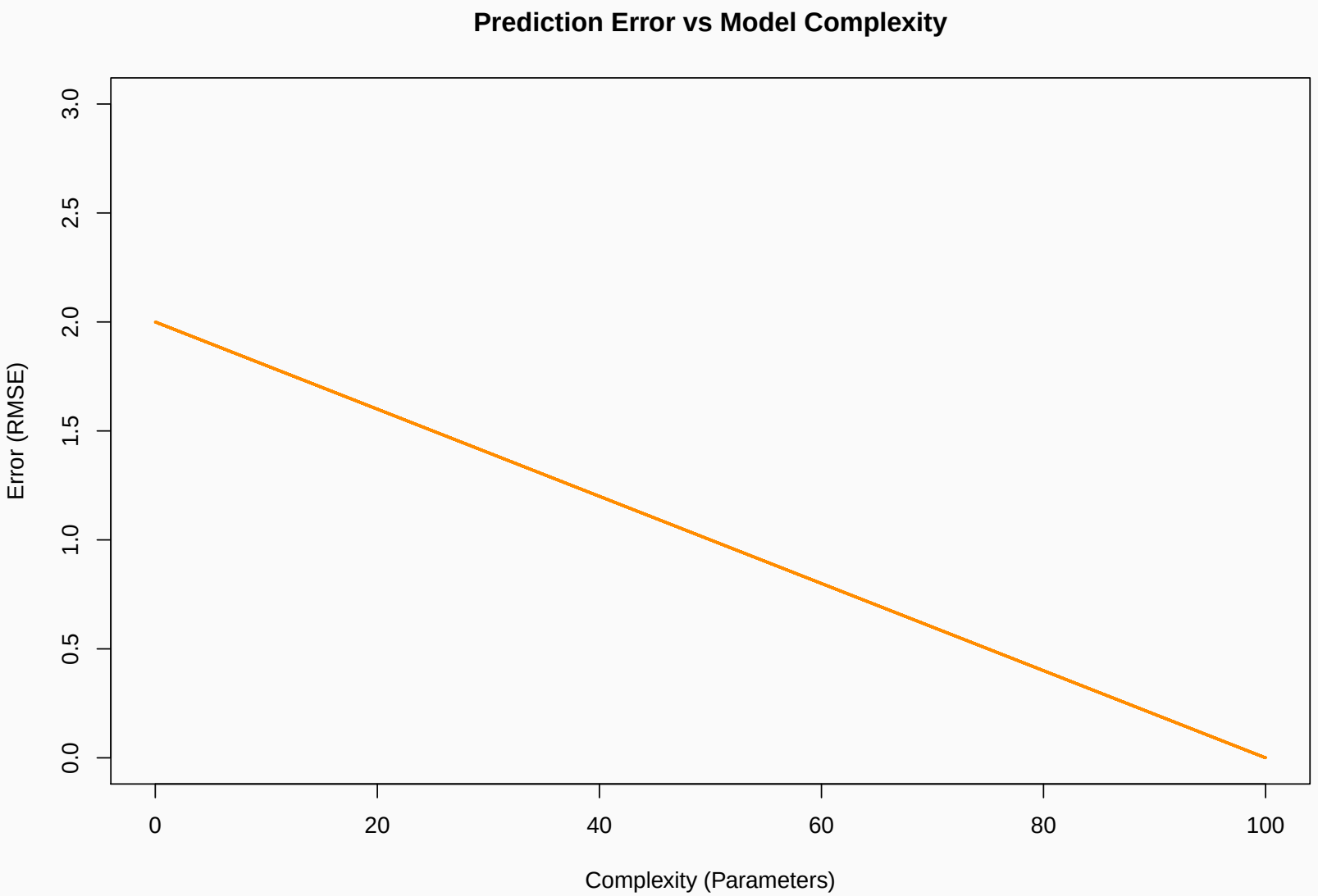
$$\text{RMSE}_{\text{Train}} = \text{RMSE}(\hat{f}, \text{Train Data}) = \sqrt{\frac{1}{n_{\text{Tr}}} \sum_{i \in \text{Train}} \left(y_i - \hat{f}(\mathbf{x_i})\right)^2}$$

$$\text{RMSE}_{\text{Test}} = \text{RMSE}(\hat{f}, \text{Test Data}) = \sqrt{\frac{1}{n_{\text{Te}}} \sum_{i \in \text{Test}} \left(y_i - \hat{f}(\mathbf{x_i})\right)^2}$$

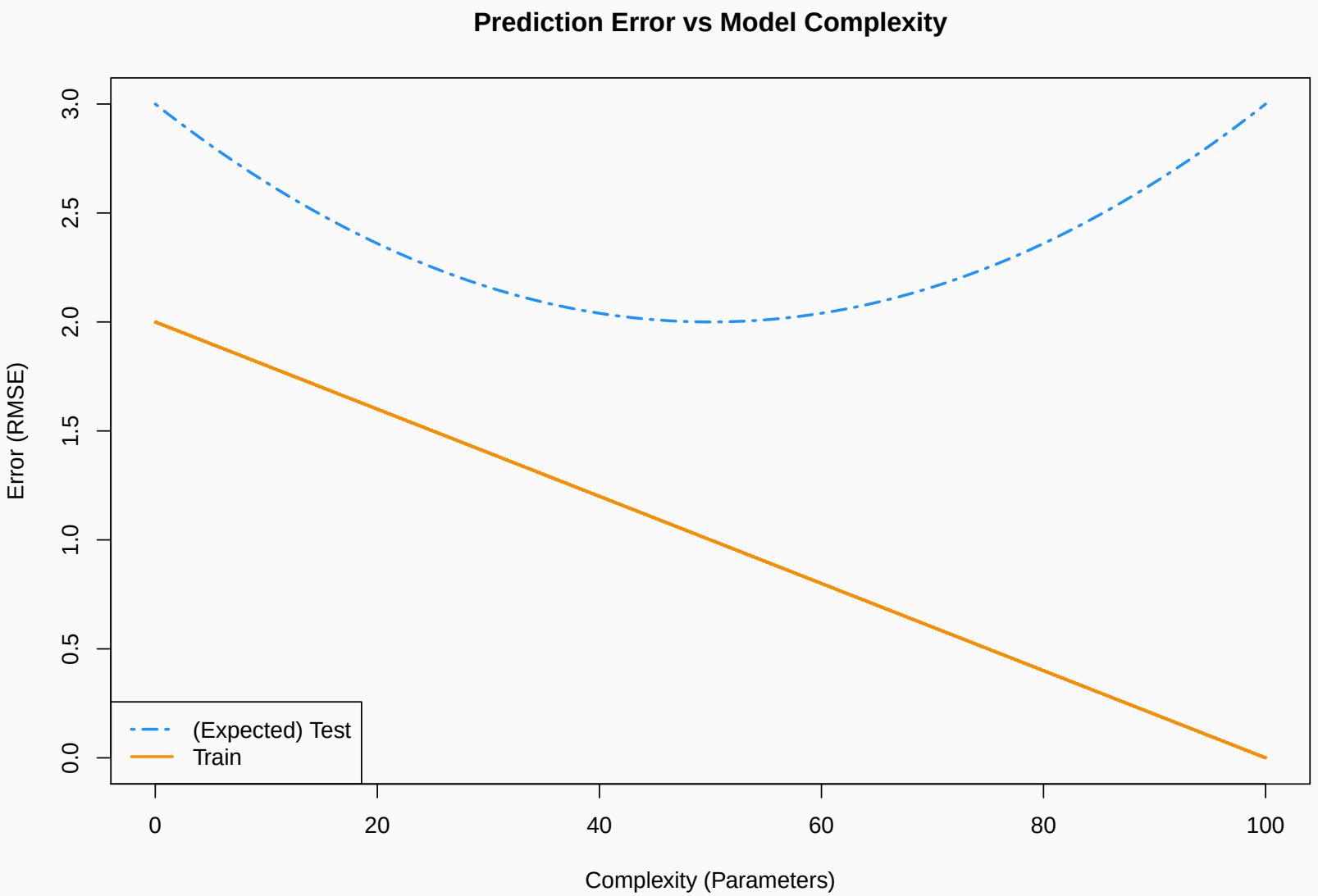
Overfitting

- **Overfitting** occurs when a model is *too complex* (too flexible) for the data
- **Underfitting** occurs when a model is *not complex enough* (too inflexible) for the data

Train RMSE


Prediction Error vs Model Complexity
Error (RMSE)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0
20
40
60
80
100
Complexity (Parameters)

(Expected) Test RMSE



The "Best" Model

- Pick the model with the lowest **Test** RMSE
- Compared to this...
  - *More complex* models with higher Test RMSE are **Overfitting**
  - *Less complex* models with higher Test RMSE are **Underfitting**
- This is only a "guess" of the "best" model based on available information
- In practice, Test RMSE might not be such a nice curve
  - This is due to the randomness of the split
  - You could get lucky, or unlucky

Explanation vs Prediction

- Sometimes we check model assumptions directly
- When predicting, we make assumptions and check them indirectly
    - If we assume a correct (or close to correct) form of the model, the Test RMSE will be low

If Time...

- `rmarkdown` Tables
- Using code from the Internet
- Back to Test-Train Split Lab
  - What would be a good Test RMSE?
  - Overfitting: n vs p
  - Randomness of Split
- Pseudo RNG