Subject: Re: Assessing Performance
Date: Thursday, March 22, 2018 at 10:16:53 PM Pacific Daylight Time
From: Amar Pendala
Attachments: image001.png, image002.png, image003.png, image004.png, image005.png, image006.png, image007.png, image008.png, image009.png, image010.png, image011.png, image012.png, image013.png

Error Types

1. Model + algorithm -> fitted function
2. Use fitted function to make Predictions
3. Use Predictions to make decisions

But … how much am I losing due to inaccurate predictions?
Ie. How much am I loosing compared to perfection?

- Perfect predictions: Loss = 0
- My predictions: Loss = ???

Measure Loss:

- Loss function: $L(\underbrace{y}_{\text{actual value}}, \underbrace{f_{\hat{w}}(\mathbf{x})}_{\hat{f}(\mathbf{x}) = \text{predicted value } \hat{y}})$ — Cost of using $\hat{w}$ at x when y is true
- Common errors:
    - Absolute error: $L(y, f_{\hat{w}}(\mathbf{x})) = |y - f_{\hat{w}}(\mathbf{x})|$
      Squared error: $L(y, f_{\hat{w}}(\mathbf{x})) = (y - f_{\hat{w}}(\mathbf{x}))^2$

Part 1: Training error:

1. Define training data set
2. Fit model w/ training data set (quadratic to minimize RSS)
3. Compute training error
    a. Define a loss function
    b. Calculate training error == avg. loss in training set
        i. $= \frac{1}{N} \sum_{i=1}^{N} L(y_i, f_{\hat{w}}(\mathbf{x}_i))$
    c. Calculate RSME (Root Mean Square Error)
        i. RMSE = $\sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - f_{\hat{w}}(\mathbf{x}_i))^2}$

Training error decreases with increased model complexity

Small training error ≠> good predictions
unless training data includes everything you might ever see

Part 2: Generalization error

What we really want is to get estimate of loss over all possible data – even outside our data set. Not possible though …
So what we will do is give weights to our data points based on how likely it would have occurred in our data set

Generalization error: average value of loss weighted by how likely those pairs were in our data set

Formally:

average over all possible (x,y) pairs weighted by how likely each is

$$\text{generalization error} = E_{\mathbf{x},y}[L(y, f_{\hat{\mathbf{w}}}(\mathbf{x}))]$$

fit using training data

For very simple models, error is bad, but gets better as models get more complex. But at some point, as you get more highly crafted model for your training data set, generalization error starts to increase again.

Part 3: Test error

Approximates Generalization Error. Does so by looking at data not in our training set (ie. look at test set)

Test error

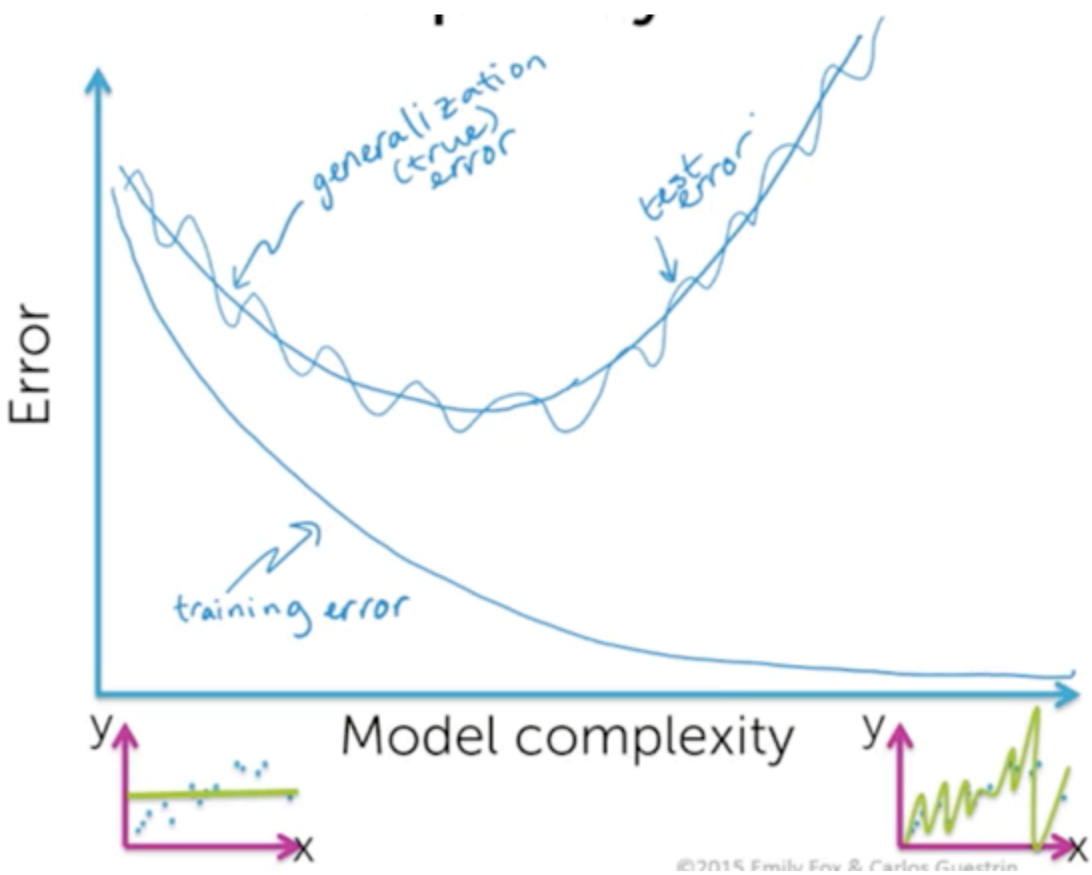
= avg. loss on houses in test set

$$= \frac{1}{N_{test}} \sum_{i \text{ in test set}} L(y_i, f_{\hat{\mathbf{w}}}(\mathbf{x}_i))$$

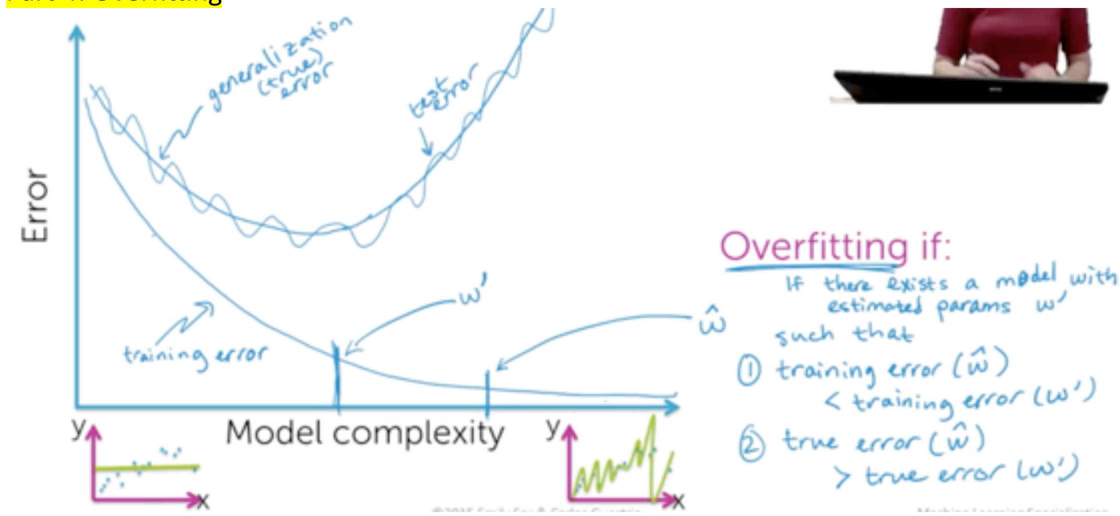
test points

fit using training data

NOTICE: test data is what you divide by, but loss is for training data



Part 4: Overfitting



Sources of Error

There are three sources of errors when forming predictions:

1. Noise
2. Bias
3. Variance

Noise:

Data is inherently noisy (ie. personal relationship/feelings, etc.. that affects value)

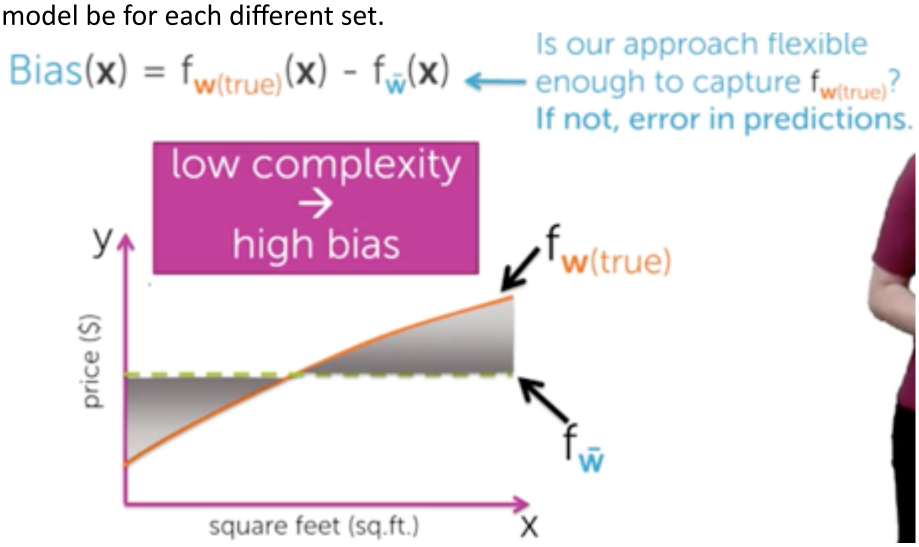
$$y_i = f_{\mathbf{w}(true)}(\mathbf{x}_i) + \varepsilon_i$$

variance of noise (aka. Irreducible error) = spread of the noise (ie. variable of value you'll likely see at a specific x value)

- This is a property of the data that you cannot handle. This is irreducible error as you cannot change model to address it

Bias == how well my model can fit the true relationship between x and y.

- If you have multiple training data sets, you'll likely get a somewhat slightly different model for each
- If you compute an average model over the different data sets, you try to determine how off will my model be for each different set.

- 

Variance = how much do specific fits (for a specific training data set) vary from the expected (average) fit

- high complexity models have high variance
  - because changing a few data points to a highly complex (ie. overly fitted model) can change each model dramatically, so average fit will have higher variance
- low complexity models have lower variance

High complexity models

- high variance
- low bias

Low complexity models

- low variance
- high bias

Now you have **a bias-variance** tradeoff – MSE (Mean Squared Error) = bias^2+variance