101C

ADMIN STUFF

IMPORTANT STUFF

- Read the syllabus (on CCLE)
- Office Hours: Thursday 5:00-6:00pm
- Midterm in class: November 12
- Final: TBD
- Send all academic questions to Piazza, not email. Feel free to answer/participate in discussions.

101C

PREAMBLES

STRATEGY VS TACTICS

Strategy without tactics is the slowest route to victory.
Tactics without strategy is the noise before defeat.

– Sun Tzu

LECTURE R CODE

- Please visit
- https://github.com/davezes/fall2019_101C
- Clone/DL repo

101C

1

HEADS UP

- Advanced

WHAT'S THE POINT

- Inference
- 2 views, "System", Population
  - "Probability space"
  - "System"

SUPERVISED VS UNSUPERVISED

- Data modeling can be divided into two paradigms.
- **Supervised**. Simply, we have at least one or more responses that we wish to predict from explanatory variables.
- **Unsupervised**. We do not have a response, but rather seek to find "patterns" amongst our variables.
- Most of ISLR, and this course, is dedicated to exploring the **supervised** paradigm.

COST FUNCTIONS

- **How do we assess the "quality" of our prediction?**
- It depends.
- We don't spend a great deal of time studying/considering different cost functions in 101C.
- Actuarial sciences.
- Quantitative Response, MSE, RMSE (same minimum)
- Example, AC "attainment"

COST FUNCTIONS

- Very common cost function for quantitative response is the RMSE.
- RV version: $\sqrt{\mathrm{E}\left[\left(Y - \widehat{Y}\right)^2\right]}$
- Data version: $\sqrt{\frac{1}{n}\sum_{i}^{n}\left[\left(y_i - \widehat{y}_i\right)^2\right]}$

STRATEGERY

- Our general business in this course, and others like it, is to create $\hat{f}(\mathbf{x})$ from data, $(\mathbf{X}, \mathbf{y})$.
- This process itself is a *function*.

FUNCTIONS THAT CREATE FUNCTIONS

- For example, consider the multivariate linear solution:
- $\hat{f}(\mathbf{x}) = \mathbf{x}\,(\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{y} = \mathbf{x}\,\widehat{\boldsymbol{\beta}}$
- Notice that this itself is a function of our data, $(\mathbf{X}, \mathbf{y})$.

FUNCTIONS THAT CREATE FUNCTIONS

- We can think of this process as:
- $h(\mathbf{X}, \mathbf{y}) = x\,(\mathbf{X}^T\mathbf{X})^{-1}\,\mathbf{X}^T\mathbf{y} = x\,\widehat{\boldsymbol{\beta}} = \hat{f}(x)$
- The important thing to notice is that h is a function of our observations, $(\mathbf{X}, \mathbf{y})$, whereas the function it creates, $\hat{f}$, need not be a function of observations.

FUNCTIONS THAT CREATE FUNCTIONS

- h may require constants for the creation of $\hat{f}$.
- Such constants may be called “hyper-parameters”.
- Just for example, in Ridge regression, we have $h(\mathbf{X}, \mathbf{y}, \lambda) = x\,(\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\,\mathbf{X}^T\mathbf{y} = x\,\widehat{\boldsymbol{\beta}} = \hat{f}(x)$.

FUNCTIONS THAT CREATE FUNCTIONS

- The ISLR book we're using this quarter, and others like it, are really a collection of different h's. In some sense, there's little more to it than that.
- Understanding, at the most fundamental level, that estimation techniques are ultimately build atop functions of our data, makes dealing with "Testing", "Training", "Validation", much more natural.

FUNCTIONS THAT CREATE FUNCTIONS

- It also makes dealing with some of the key theoretical concepts much easier.
- If you don't like the word "theory", you can substitute "how/why it works".

KEYS

- Two of the keys to the theory of estimation are found in Chapter 2.
- Total Estimation Error (the square of which is called the Prediction Variance), pg 19
- The Bias-Variance Trade-Off, pg 34

SYSTEM

- $y = f(x) + \varepsilon \ , \quad \varepsilon \sim \mathcal{N}[0, \sigma]$

TOTAL ESTIMATION ERROR

- Total Estimation Error (the square of which is called the Prediction Variance):
- $\mathrm{E}\left[\left(Y-\widehat{Y}\right)^2\right] = \mathrm{E}\left[\left(f(x)+\varepsilon-\hat{f}(x)\right)^2\right] = \left(f(x)-\hat{f}(x)\right)^2 + \mathrm{Var}[\varepsilon]$
- This is easily demonstrated by expanding and factoring terms …
- However, to pull this off, we need to recognize two basic properties …

TOTAL ESTIMATION ERROR

- The first, that E $\left[f(x)\ \varepsilon\right] = 0$. This follows directly from the definition of Y.
- The second is much more interesting, that E $\left[\hat{f}(x)\ \varepsilon\right] = 0$.
- Technically, this requirement **may not in general be true**.
- $\hat{f}$ **could be** a stochastic function of some correlate of ε.
- It is not – not in 101C.

TOTAL ESTIMATION ERROR

- This becomes entirely clear if we turn back to our good friend, the meta function, h.
- There are two ways to conceptualize how h creates $\hat{f}$.
- One, likely the most comfortable, is to view h as a function of **data**, $h(\mathbf{X}, \mathbf{y})$, and perhaps some hyper-parameters. While these data may be realized through a stochastic process, once they are realized they are considered fixed – and so too the function $\hat{f}$ that h has created.

TOTAL ESTIMATION ERROR

- The other is to allow h to be a function of Y as a random variable, $h(\mathbf{X}, Y)$, so that, interestingly, $\hat{f}(x)$ can be thought of as a random variable.
- Either way, the result is the same:
- Since ε are **mutually independent**, then E $\left[\hat{f}(x)\ \varepsilon\right] = 0.$

TOTAL ESTIMATION ERROR

$$\mathrm{E}\left[\left(Y - \widehat{Y}\right)^2\right] = \underbrace{\left(f(x) - \hat{f}(x)\right)^2}_{\text{Reducible}} + \underbrace{\mathrm{Var}[\varepsilon]}_{\text{Irreducible}}$$

BIAS-VARIANCE TRADE-OFF

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

$$\mathrm{bias}\left[\hat{f}(x_0)\right] = \mathrm{E}\left[\hat{f}(x_0)\right] - f(x_0)$$

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

BIAS-VARIANCE TRADE-OFF

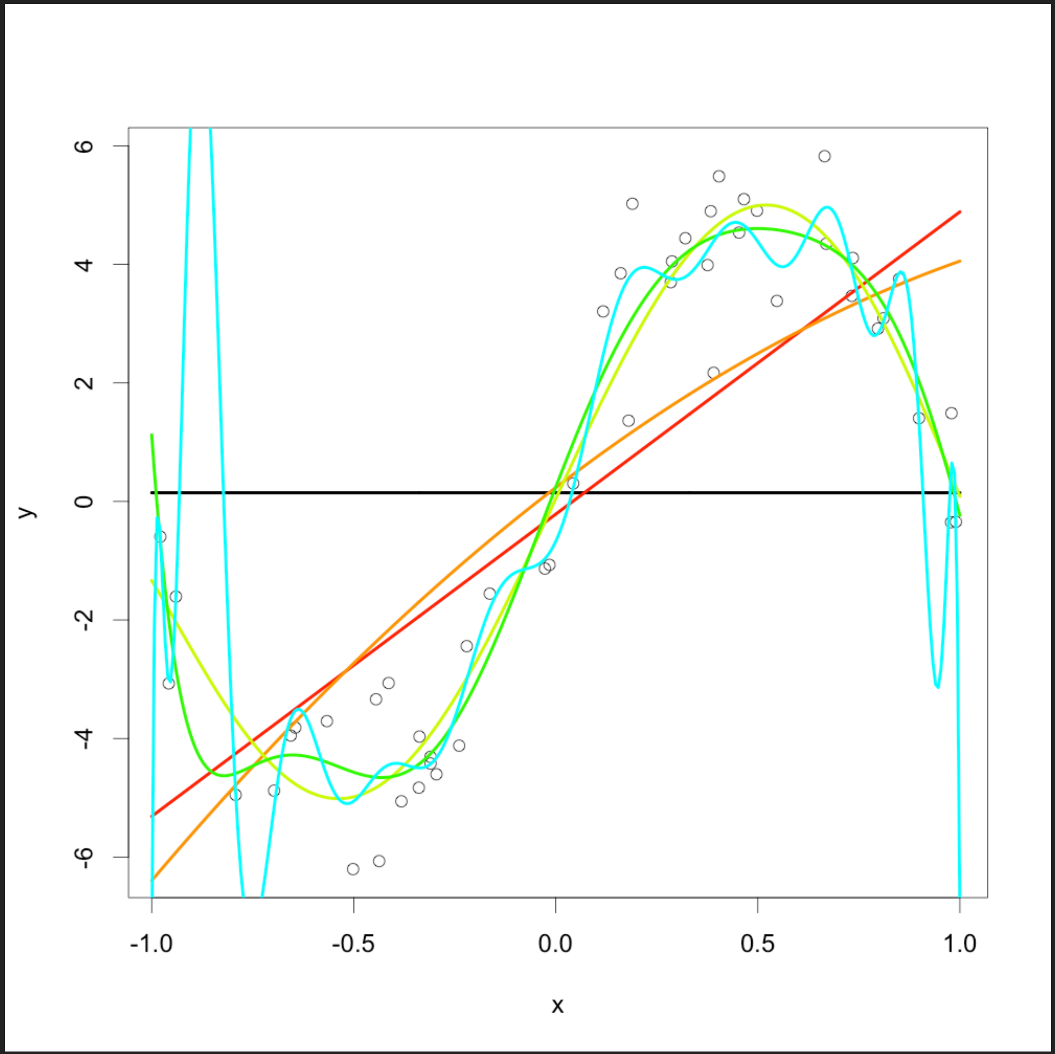
- From a data perspective, how do we deal with E $\left[\hat{f}(x_0)\right]$?
- We can approximate it. Easy to do, e.g., especially using simulation.
- We again turn to our good buddy, h.
- Imagine we have m (preferably equal-sized) data sets. So now …
- $h\left(\mathbf{X}_{(j)}, \mathbf{y}_{(j)}\right) = \hat{f}_j(x)\,, \quad j \in \{1,2,3,\ldots,m\}$
- For each, we choose some $(x_0, y_0)_i$ …

BIAS-VARIANCE TRADE-OFF

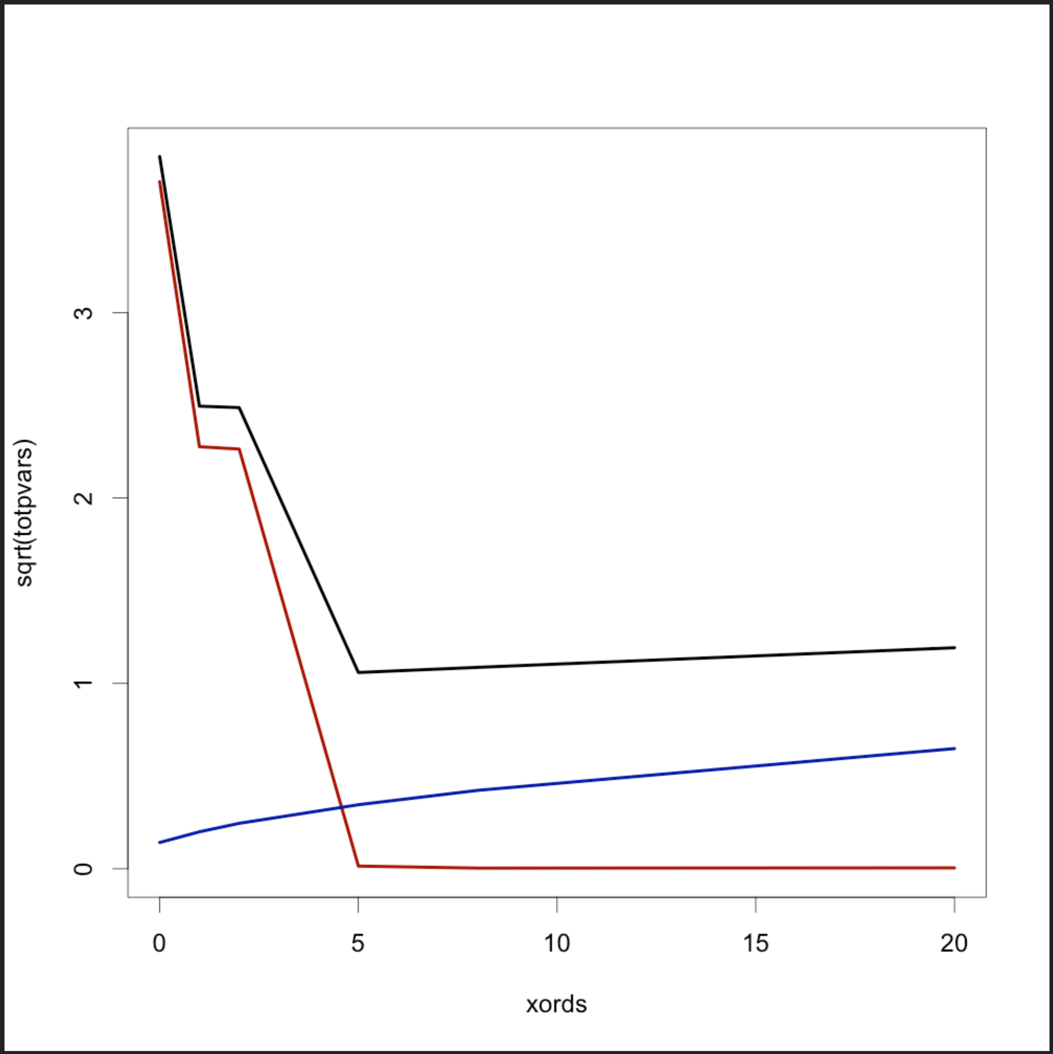
- Using simulation, we would choose x_0, then simulate – for each j – $y_{0,j}$, i.e., $y_{0,j} = f(x_0) + \varepsilon$
- So then,
- $\mathrm{E}\left[\hat{f}(x_0)\right] \approx \frac{1}{m}\sum_{j}^{m} \hat{f}_j(x_0)$
- $\mathrm{E}\left[\left(y_0 - \hat{f}(x_0)\right)^2\right] \approx \frac{1}{m}\sum_{j}^{m} \left(y_{0,j} - \hat{f}_j(x_0)\right)^2$

BIAS-VARIANCE TRADE-OFF

- File: _01_bias_variance.R

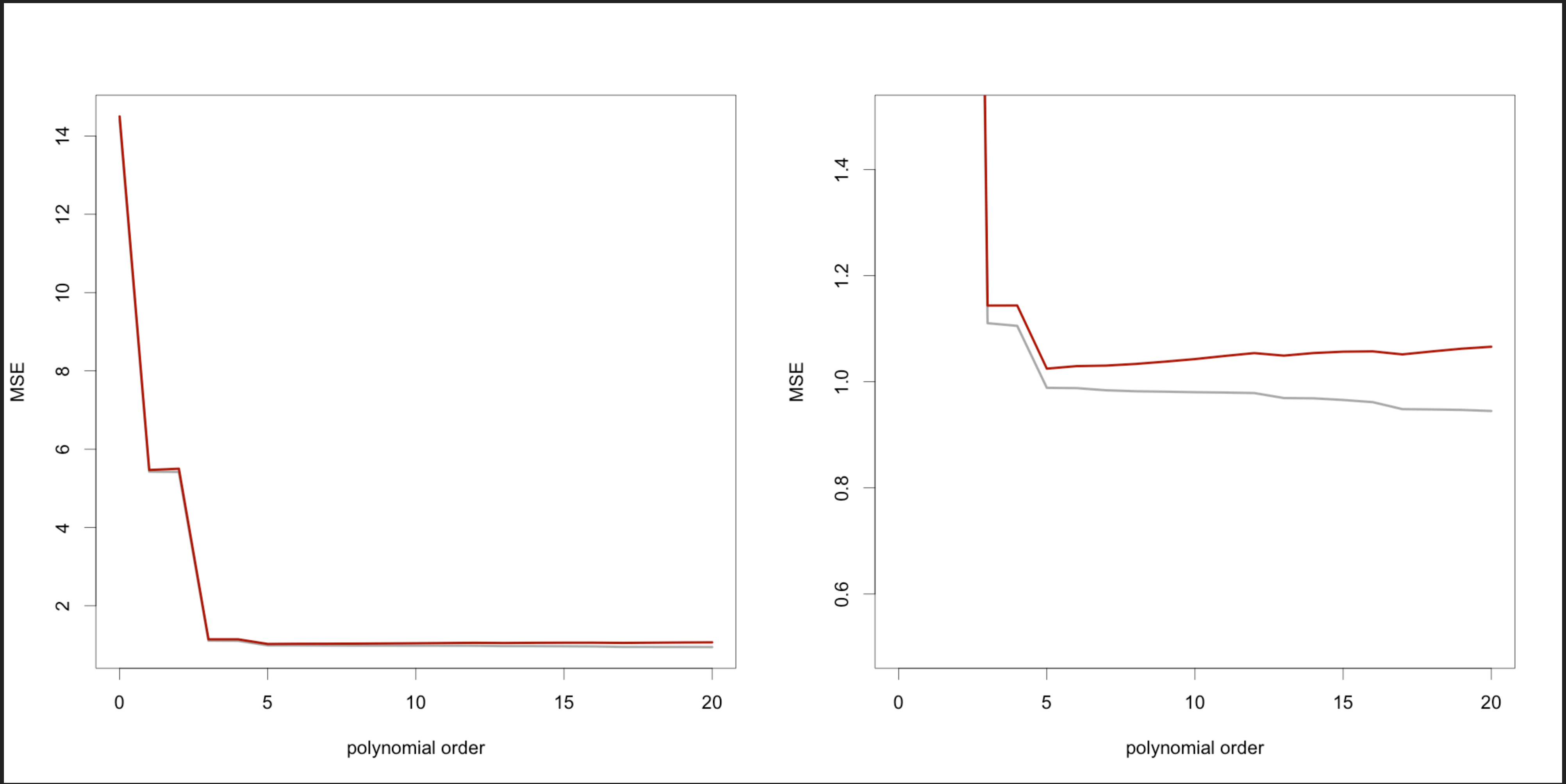


6
4
2
0
-2
-4
-6
y
-1.0
-0.5
0.0
0.5
1.0
x

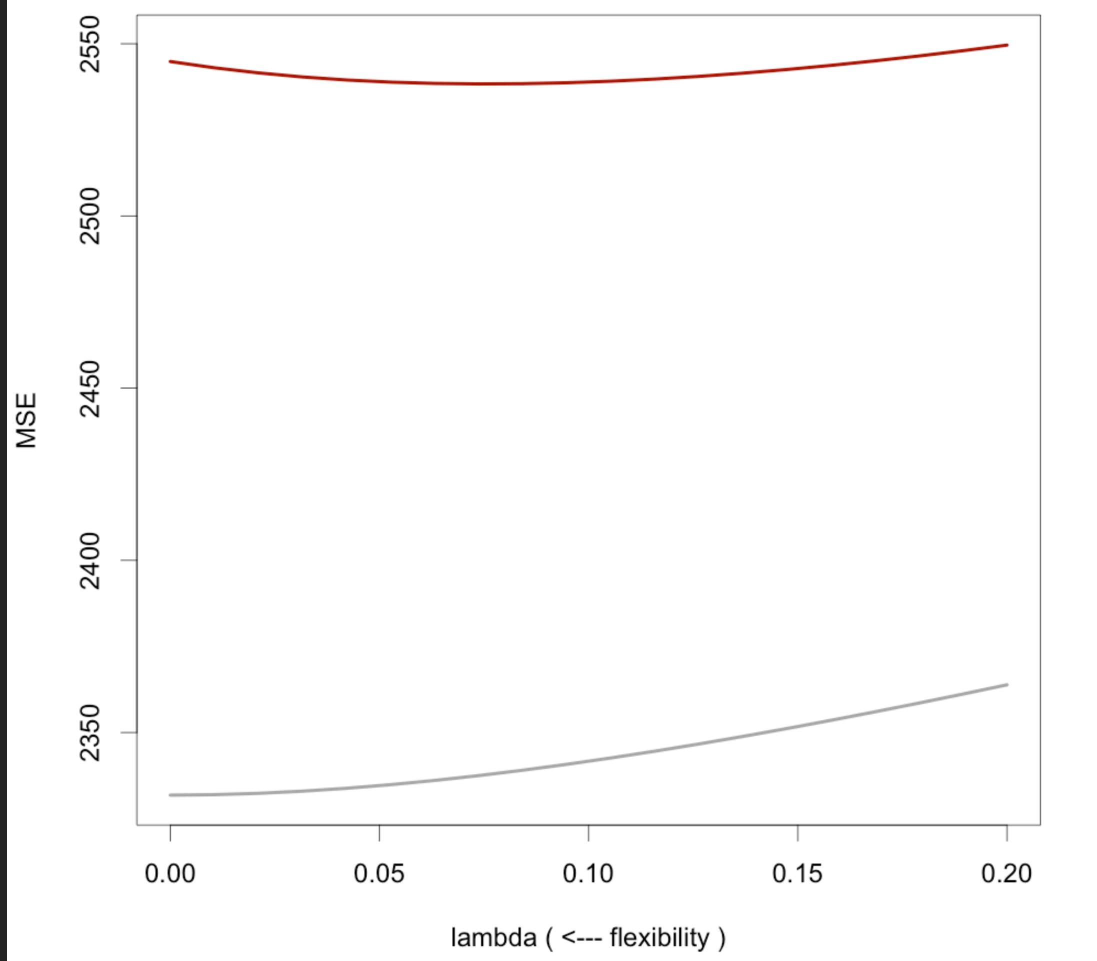

sqrt(totpvars)
0
1
2
3
0
5
10
15
20
xords

BIAS-VARIANCE TRADE-OFF

- File: _01_poly_and_Ridge_Train_Test.R



MSE
polynomial order
0
5
10
15
20
2
4
6
8
10
12
14
0.6
0.8
1.0
1.2
1.4


2550
2500
2450
2400
2350
MSE
0.00
0.05
0.10
0.15
0.20
lambda (<--- flexibility)

CLASSIFICATION

- Our response of interest may be categorical.
- In such a case the common assumption for modeling a quantitative response, $y = f(x) + \varepsilon$, simply doesn't make sense.
- Although in truth it could be that

$$y = \begin{cases} 1 & f(x) + \varepsilon > \alpha \\ 0 & f(x) + \varepsilon \leq \alpha \end{cases}$$

CLASSIFICATION

- As we saw previously, the joint distribution, $\phi(x, y)$, if it is known, is **always** the best way to inferentially relate y to x.
- The true joint density eliminates reducible error (provided we use it correctly).

CLASSIFICATION

- For this reason, one possible model choice for predicting a categorical response is to estimate the joint density, $\widehat{\phi}(x, y)$.
- However, if there are many variables, $\phi(x, y)$, and hence, $\widehat{\phi}(x, y)$ will be a surface over many dimensions, and unless we have an enormous number of observations, estimating $\widehat{\phi}(x, y)$ may be impractical (or at least unacceptably imprecise).

K-NEAREST NEIGHBORS

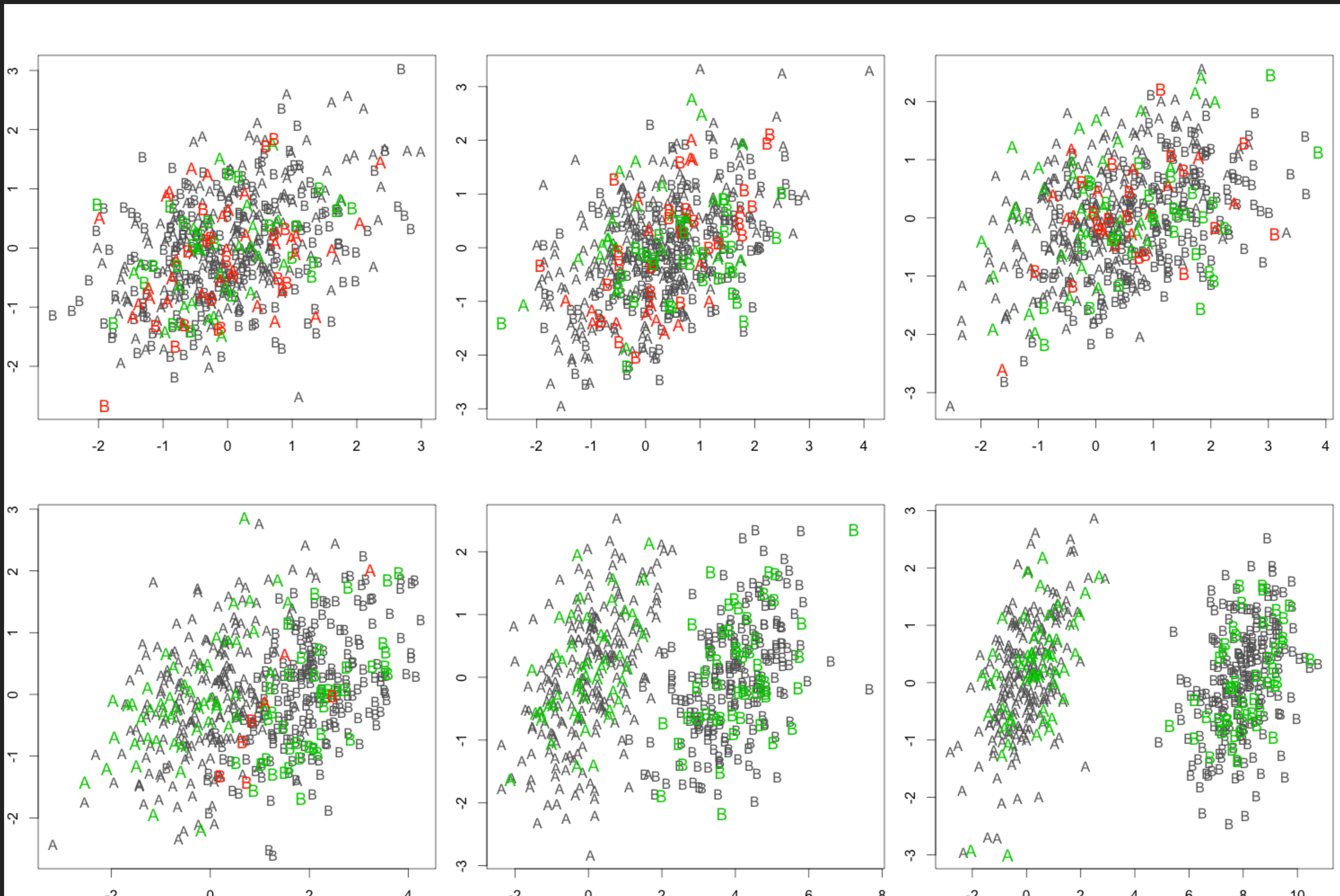
- When our predictors are quantitative, a very common, and very simple, and potentially very effective (precise) way to make predictions is k-nearest neighbors, or KNN.
- KNN is usually regarded as non-parametric, and aside from a pre-defined distance function, requires only a single hyper-parameter, $k \in \{1, 2, 3, \ldots\}$.

K-NEAREST NEIGHBORS

- A new point will locate the k-closest points in the data set.
- It will then ask each of them, “hey, buddy, what category are you?”
- It will then declare that its category is the preponderance of the categories of its neighbors.

K-NEAREST NEIGHBORS

- File: _01_categorical_response.R



CONFUSION TABLE

- AKA, “error table”
- Just to note, the terms “false positive”, “false negative”, “true positive”, “true negative”, in statistics circles often refer to decision processes in statistical tests.
- But we can use them here to refer to individual observations/predictions.

ACTUAL VS PREDICTION 2X2

Our Prediction / Actual	Yes	No
Yes	True Positive	False Negative
No	False Positive	True Negative

ACTUAL VS PREDICTION 2X2

- Empiric “false positive rate”, AKA “false alarm ratio”, AKA “false positive ratio”:
- $$\frac{N_{FP}}{N_{FP} + N_{TN}}$$
- Empiric “false negative rate”:
- $$\frac{N_{FN}}{N_{FN} + N_{TP}}$$

MISC

- Before our next get together:
  - Read Ch 3
  - Skim Ch 5 (again)
  - HW 1 due Sunday – 2019-10-06