Data, Environment and Society, Lecture 22: Exam Review

Instructor: Duncan Callaway

GSI: Salma Elmallah

November 14, 2019

Slides in this deck…

- Are just taken from slide decks from earlier in the semester.
- A few other notes on what the exam will cover:
  - Everything we've covered in lecture and in reading is fair game
  - I'll cover through Lecture 21 (Benami guest lecture)
  - Relative to last year, less emphasis on Python, more emphasis on how algorithms work, conceptualizing resource allocation problems.

Exploratory Data Analysis (EDA)

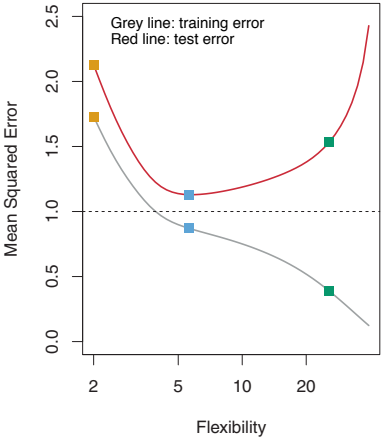
One can approach EDA by asking questions about the data:

- Structure
- Granularity
- Scope
- Temporality
- Faithfulness

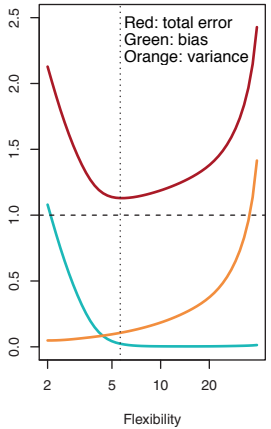
Principles of visualization

- Scale
- Conditioning
- Perception
- Transformation
- Context
- Smoothing

Decomposing bias-variance



Take a moment to think about how bias and variance add up to make the red curve on the left. Try to draw bias and variance separately.



How to evaluate how well a model performs? The *Cost function.*

- Cost functions can be used to describe how much of the variation in the data can be captured by the model.
- Example: The mean squared error:

$$MSE = \frac{1}{n}((y_1 - \hat{y}_1)^2 + (y_2 - \hat{y}_2)^2 + \cdots + (y_n - \hat{y}_n)^2)$$

$$= \frac{1}{n}(e_1^2 + e_2^2 + \cdots + e_n^2)$$

$$= \frac{1}{n}\sum_{i=1}^{n} e_i^2$$

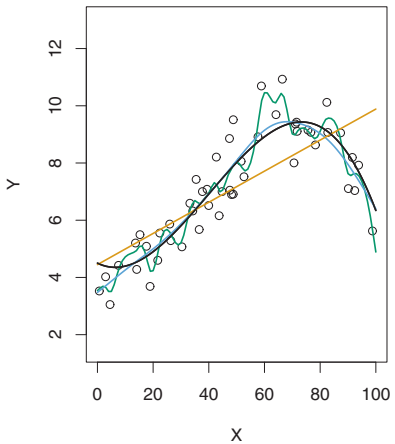
A major part of statistical learning lies in how the cost function is defined.

A thought experiment from ISLR Ch 2

Suppose you have four different model forms to choose from. When you fit them to the data, you get this figure.

Which model should you choose?

- The one that minimizes mean squared error?
- Careful! Doesn't the squiggly one minimize mean squared error?
- To do model selection we need to understand the concept of training and testing data.



Parametric vs. non-parametric models

The model examples we discussed so far are **parametric**, meaning they relate inputs to outputs with a mathematical function defined by parameters.

But **non-parametric** models are also possible.

- These don't use functions with coefficients
- Instead the data *become* the model

It's easiest to see this by example using the K-nearest neighbors algorithm.

Let's be clear...

What do people doing prediction care about, $\hat{\beta}$ or $\hat{y}$?

$\hat{y}$!

What measure should people doing prediction use to evaluate model performance, coefficient confidence intervals, RSS, R^2 or p?

RSS or R^2 are suitable. But there is much more to the story!

- Today we'll talk about adjustments to R^2 that attempt to address bias-variance tradeoff
- We'll discuss other approaches in the coming weeks.

Model selection methods

Two basic methods:

- Computationally heavy and theoretically robust:
  - repeated sampling of train and test data sets
  - build and test models with each sampled set
  - choose the model form that minimizes test error, on average.
  - the figure on the previous slide is an example of this approach.

- Easy to implement (no need for significant computing):
  - Use the full data set
  - Fit each candidate model once
  - Choose the model that minimizes an “adjusted” measure of R2 or mean squared error.

Qualitative predictors, defined

Quantitative predictors

- have a natural order, or
- values can be summed, and
- often have units of measurement.

Qualitative predictors do not have these characteristics.

Nonlinear predictors

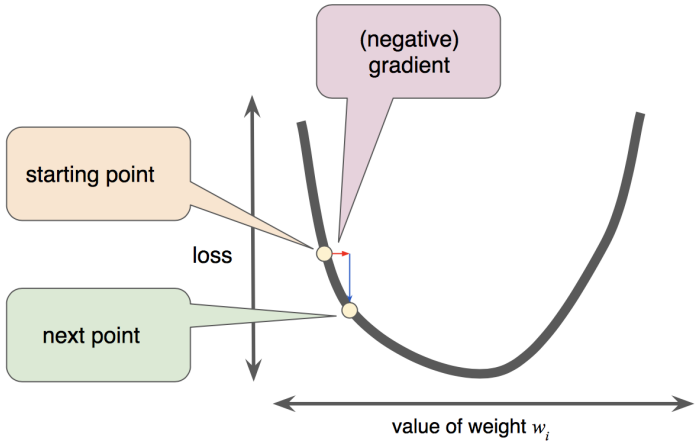
We can specify virtually any nonlinear model you can think of. For example:

$$\hat{y}_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^{\frac{1}{3}} + \beta_4 f(x_i)$$

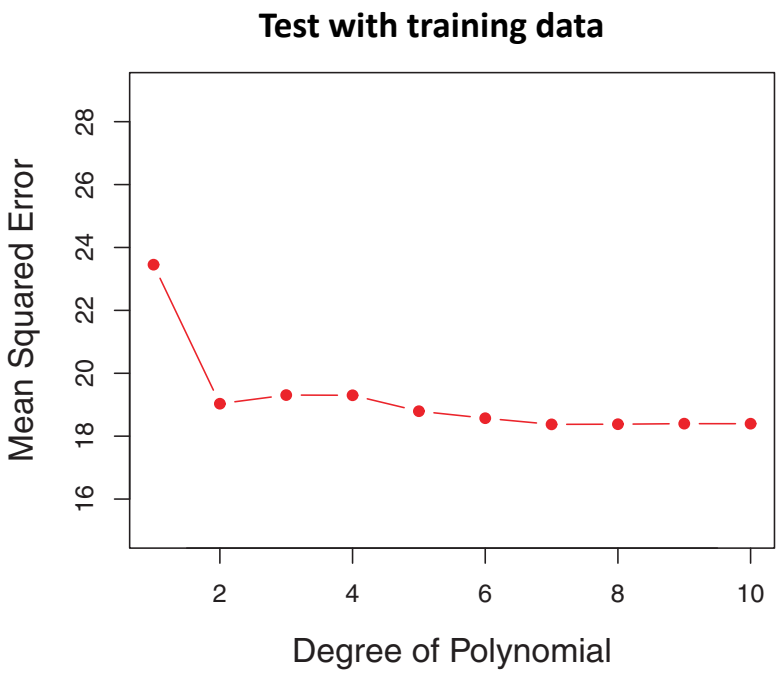
$f(x_i)$ can be any function you want!

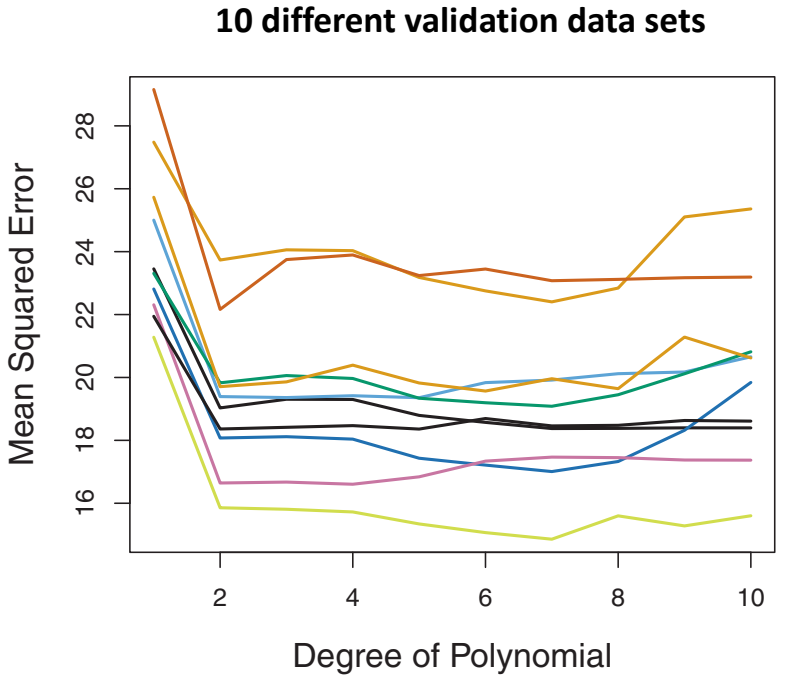
Let's see how this might play out in the Novotny data. Check out the Lecture 11 Jupyter notebook.

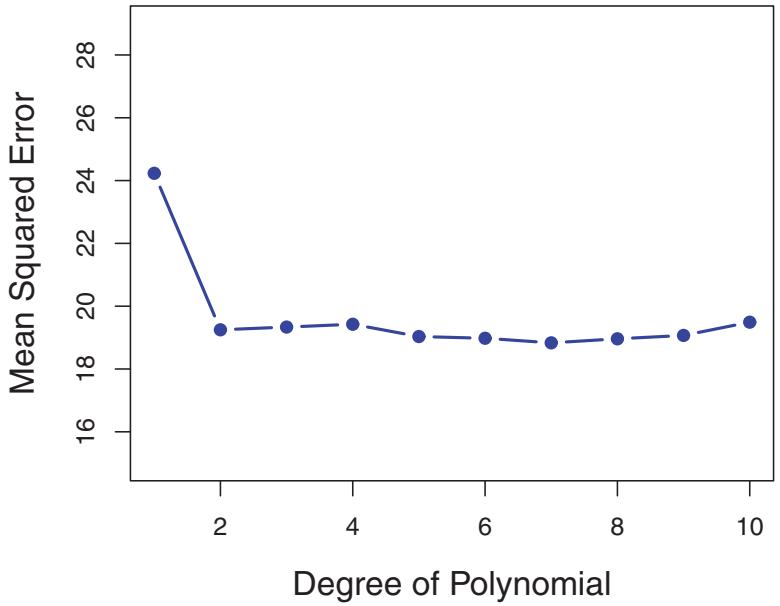
Gradient descent – sketch

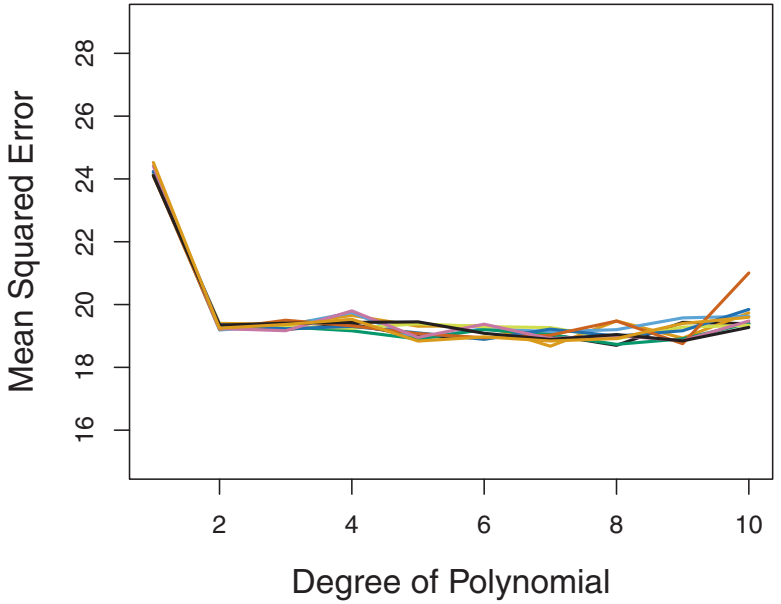


https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent









- Validation can be used for model selection.
- In these figures (Auto data set from the book) the minima are in different locations for different validation approaches
- 10-fold: each line is a different random split into 10 folds.

Recap lecture objectives from last time (from lecture 14-15)

- Refine our understanding of model identification as an optimization problem

$$\min_{\beta} \sum_{i=1}^{N} (Y_i - X_i\beta)^2 + \lambda \cdot R(\beta)$$

Important: We drop the λ term for prediction, i.e. predictions are just

$$\hat{y}_i = X_i\hat{\beta}$$

where X_i and $\hat{\beta}$ are vectors

- Understand what "regularization" is and why we do it
  - A tool for adapting optimization problems to be "well behaved"
  - In statistical learning, a tool to tradeoff bias and variance

But note, R causes you to solve a different problem than the original $\rightarrow$ parameter bias

Recap lecture objectives from last time, ctd (from lecture 14-15)

- Continue thinking about how to adjust errors to compare models with different p
  - k-fold cross validation, AIC, BIC, adjusted R^2...
- Learn the tradeoffs between subset selection, ridge and lasso
  - Speed (fastest to slowest): Ridge, Lasso, Subset
  - Subset selection and Lasso do feature selection. Ridge does not.
  - You can naturally tune prediction bias-variance with Ridge and Lasso

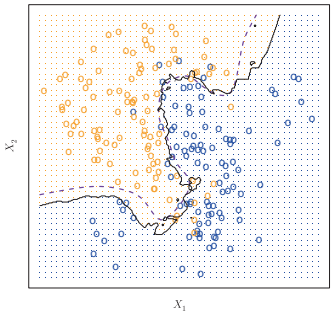
Today's objectives

(from lecture 14-15)

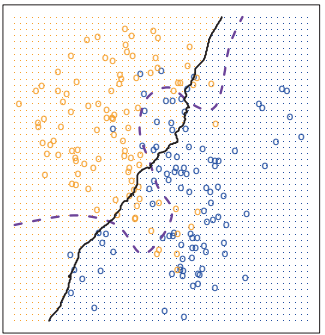
1. Quick review of the basic mechanics of Subset selection, Ridge and Lasso.
2. Build deeper intuition on how they work and how they differ.
3. Learn how the bias-variance tradeoff gets tuned with regularization term parameters.
4. Understand the tradeoffs between these methods in more detail
5. Understand the importance of standardizing your variables.
6. Epilogue: the elastic net, a machine learning mashup.

Which has the highest K? Which has the lowest?

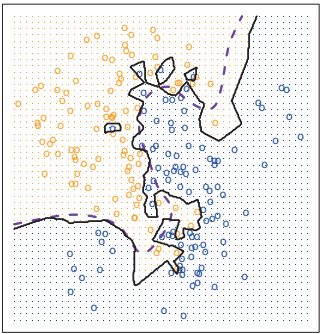
Dashed = Bayes decision boundary
Solid = KNN estimate of Bayes decision boundary



K = 10



K = 100



K = 1

Where should the splits be? (from lecture 16-17)

Then we partition any region by choosing j and s as follows:

$$\{j, s\} = \arg \min_{j \in J, s \in X_j} \sum_{i: x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

where $\hat{y}_{R_k}$ is the mean of all response variables in region k.

It would be tedious to identify j and s by hand, but it's actually very quick computationally.

Question: How many j-s pairs for p features and n observations?

Where should the splits be? (from lecture 16-17)

Then we partition any region by choosing j and s as follows:

$$\{j, s\} = \arg \min_{j \in J, s \in X_j} \sum_{i: x_i \in R_1(j,s)} (y_i - \hat{y}_{R_1})^2 + \sum_{i: x_i \in R_2(j,s)} (y_i - \hat{y}_{R_2})^2$$

where $\hat{y}_{R_k}$ is the mean of all response variables in region k.

It would be tedious to identify j and s by hand, but it's actually very quick computationally.

Question: How many j-s pairs for p features and n observations?

- No more than $p(n-1)$, since we can only choose $(n-1)$ boundaries between observations.
- There may be fewer, if separate observations share the same values for some of their features.

How much does this cow weigh? (from lecture 19)

According to James Surowiecki's book, *The Wisdom of Crowds*, in 1906 Frances Galton averaged all of a crowd's guesses for a heffer and they were only 1% off.

Three ways to build many trees from the same data (from lecture 19)

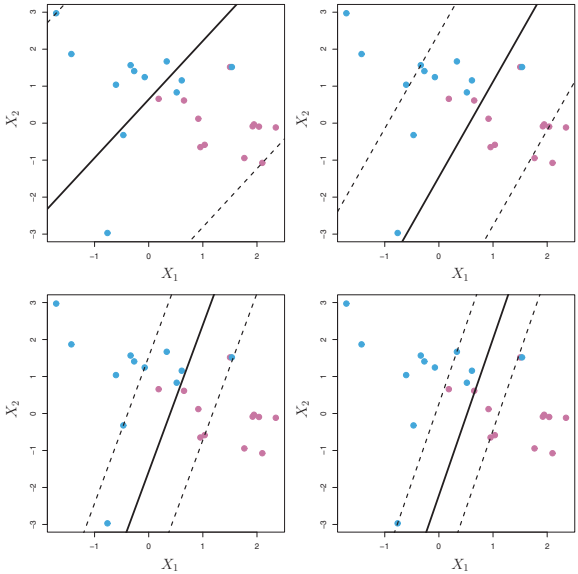
- **Bagging** (Bootstrap aggregation): Build many trees from random samples of the data
- **Random forests**: Build many trees from bootstrapped samples, but each binary split is chosen from a random subset of predictors
- **Boosting**: choose new trees to minimize the residual of an existing aggregation of trees.

Objectives for today (from lecture 20

- Some examples
- Introduce the idea of a **hyperplane** (it's really simple)
- Figure out what a **maximal margin hyperplane** (MMH) is and why we use it
  - Note, these only work for *separable data*
- Understand how **support vector classifiers** extend the MMH to cases when the data are not separable.
  - SVCs are *linear* separations of the feature space
- Open your horizons to the **support vector machine**
  - This provides nonlinear separations of the feature space!

Tuning C

(from lecture 20



Questions

1. Which plot has large C? Which is small?
2. What's going to have the highest variance? Large or small C?