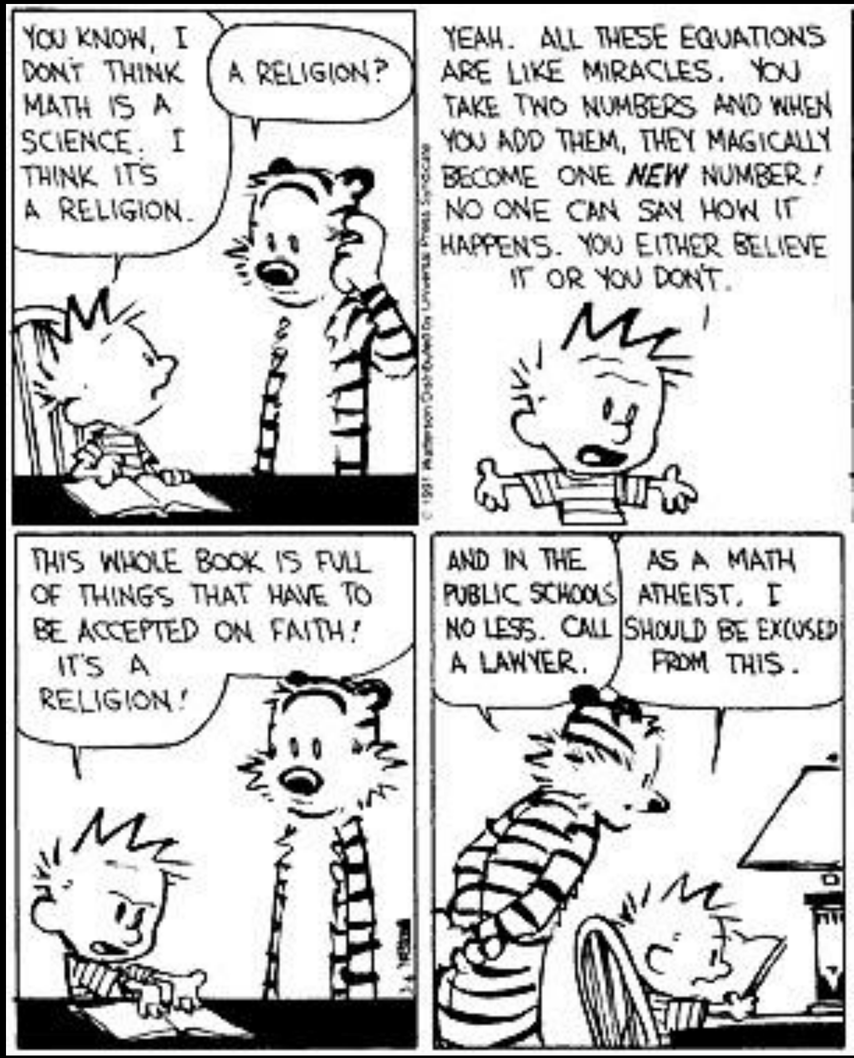
INFO 251: Applied Machine Learning



Regularization

Announcements

- Assignment 3 due Thursday
- Quiz 1 scheduled for March 4, first ~40 minutes of class
  - Closed book: No access to slides/internet/GenAI/reference materials
  - Must be present to take quiz!
  - 10-15 multiple choice and short-answer questions
  - I'll be providing a few sample questions in class

Course Outline

- Causal Inference and Research Design
  - Experimental methods
  - Non-experiment methods
- **Machine Learning**
  - Design of Machine Learning Experiments
  - **Linear Models and Gradient Descent**
  - Non-linear models
  - Fairness and Bias in ML
  - Neural models
  - Deep Learning
  - Practicalities
  - Unsupervised Learning
- Special topics

Key Concepts (last lecture)

- Cost Functions
- Gradient Descent
- Local and global minima
- Convex functions
- Incremental vs. Batch GD
- Learning rates
- Feature scaling

Stopping conditions

```
Choose an initial vector of parameters α, β
Choose learning rate R
Repeat until convergence (i.e., until an approximate minimum is obtained):
```

$$\alpha \leftarrow \alpha - R\frac{\partial}{\partial\alpha}J(\alpha,\beta)$$

$$\beta \leftarrow \beta - R\frac{\partial}{\partial\beta}J(\alpha,\beta)$$

- How to know a minimum has been obtained?
  - Look for small changes in the gradient
  - Look for small improvements in cost
  - Look for no changes in parameters
  - Set a stopping condition!

Example Quiz Question: Gradient descent

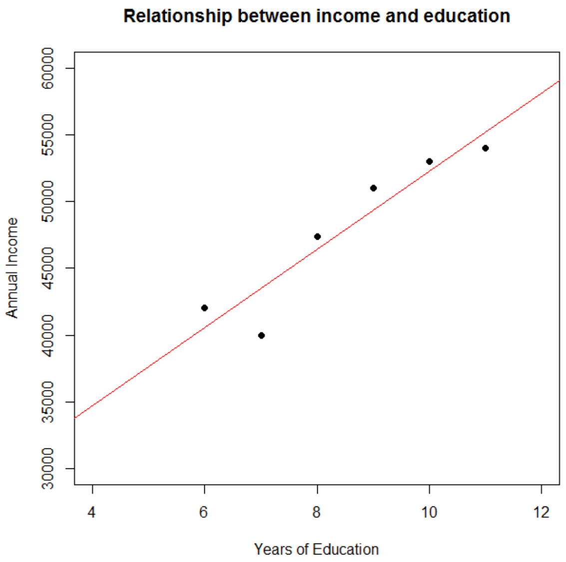
- To ensure that your gradient descent algorithm is properly converging to a minimum:
  1. Plot $J(\theta)$ as a function of θ, and ensure $J(\theta)$ is decreasing
  2. Plot $J(\theta)$ as a function of number of iterations, and ensure $J(\theta)$ is decreasing
  3. Plot $J(\theta)$ as a function of θ, and make sure $J(\theta)$ is convex
  4. Plot $J(\theta)$ as a function of learning rate R, and make sure $J(\theta)$ in monotonic (either constantly increasing or constantly decreasing) in R

Today's Outline

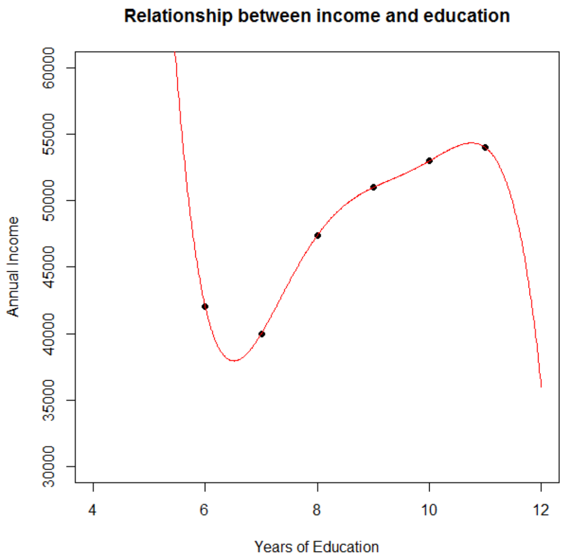
- Overfitting
- Regularization: Intuition and overview
- Ridge
- Lasso
- Logistic regression: Intro

Overfitting revisited

- Overfitting: If we have too many features, our model may fit the training set very well, but fail to generalize to new examples



$$wages_i = \theta_0 + \theta_1 Educ_i + \epsilon_i$$



$$wages_i = \theta_0 + \theta_1 Educ_i + \cdots + \theta_5 Educ_i^5 + \epsilon_i$$

Overfitting: Solutions

- Later in the course:
  - Feature selection
  - Model selection
  - Dimensionality reduction
- Now: Regularization
  - For instance, ridge regularization: Keep all the features, but reduce magnitude of specific parameters

Regularization: Intuition

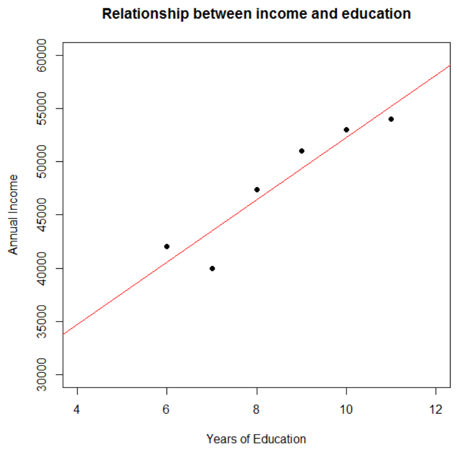
- Occam's Razor
  - A principle of parsimony, economy, or succinctness. It states that among competing hypotheses, the hypothesis with the fewest assumptions should be selected

Ockham chooses a razor

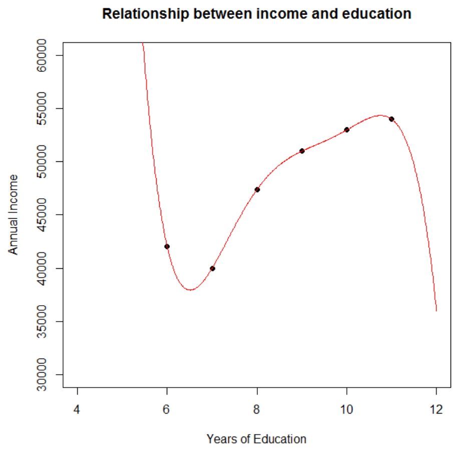
See: Domingos, P. The role of Occam's razor in knowledge discovery. *Data Mining and Knowledge Discovery 3* (1999), 409–425.

Regularization: Intuition

- Idea: Add a cost penalty for additional complexity in the model
- Example: polynomial regression
  - Model: $Y_i = \theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k$
  - Parameters: $\theta_0, \ldots, \theta_k$
  - Original "Cost": $J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2$



$wages_i = \ \theta_0 + \theta_1 Educ_i + \epsilon_i$



$wages_i = \ \theta_0 + \theta_1 Educ_i + \cdots + \theta_5 Educ_i^5 + \epsilon_i$

Regularization: Intuition

- Original cost from OLS (polynomial regression example)
  - $J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2$
- (Intuitive) Goal behind regularization
  - $J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2 + C(\theta_1, \ldots, \theta_k)$
- Example: Ridge regularization
  - $J(\theta) = \frac{1}{2N}\left[\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2 + \boxed{\boxed{\lambda}\sum_{j=1}^{k}\theta_j^{\boxed{2}}}\right]$

New penalty

"Ridge" coefficient

Regularization parameter (λ)

Regularization and Linear Regression

- Original Gradient Descent update rule

$$\alpha \text{ <- } \alpha - R\frac{\partial}{\partial\alpha}J(\alpha,\beta)$$

$$\beta \text{ <- } \beta - R\frac{\partial}{\partial\beta}J(\alpha,\beta)$$

  - Original derivative of J (in linear regression, $Y_i = \alpha + \beta X_i$)

$$\alpha \text{ <- } \alpha - R\frac{1}{N}\sum_{i=1}^{N}(\alpha + \beta X_i - Y_i)$$

$$\beta \text{ <- } \beta - R\frac{1}{N}\sum_{i=1}^{N}(\alpha + \beta X_i - Y_i) \quad X_i$$

- Regularized version has new partial derivatives:

$$\beta \text{ <- } \beta - R\left[\frac{1}{N}\sum_{i=1}^{N}(\alpha + \beta X_i - Y_i) \quad X_i + \frac{\lambda}{N}\beta\right]$$

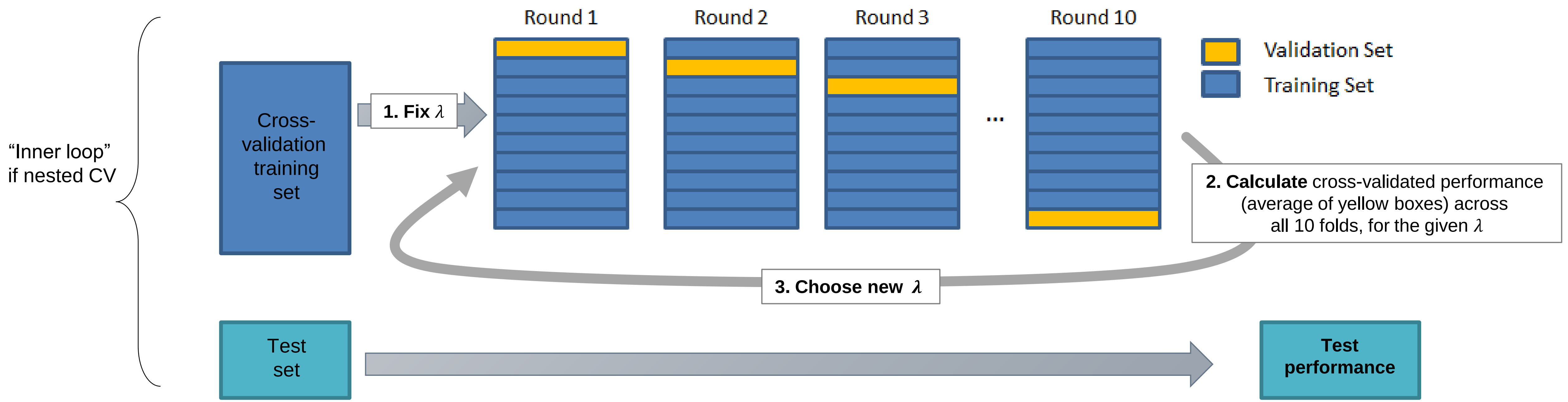
- Rewritten:

$$\beta \text{ <- } \beta\left(1 - R\frac{\lambda}{N}\right) - R\frac{1}{N}\sum_{i=1}^{N}(\alpha + \beta X_i - Y_i) \quad X_i$$

Regularization: Some notes

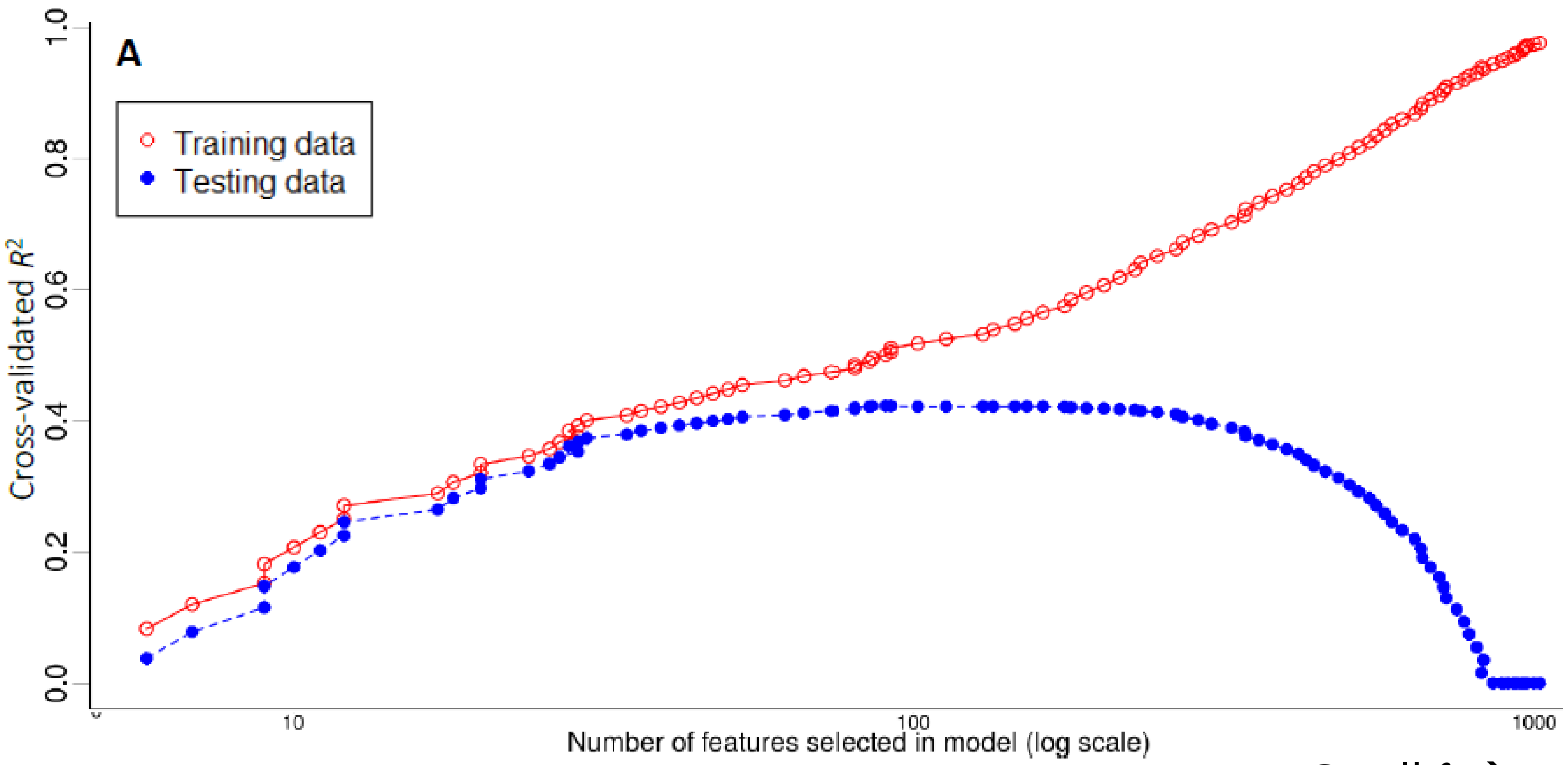
- How to select λ?
  - Cross validation!
    - Choose λ that minimizes cross-validated performance (yellow boxes)
    - i.e., repeat dark blue process for a variety of candidate values of λ

$$J(\theta) = \frac{1}{2N}\left[\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \dots + \theta_k X_i^k - Y_i\right)^2 + \lambda \sum_{j=1}^{k} \theta_j^2\right]$$



Regularization: Some notes

- Example from an early paper I wrote



← Large λ

Small λ →

Regularization: Some notes

Polynomial regression example: $J(\theta) = \frac{1}{2N}\left[\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2 + \lambda \sum_{j=1}^{k} \theta_j^2\right]$

- What happens in regularization if features are in different units?
  - Penalty on different scales
  - One solution: Normalize features
- Do we penalize the intercept?
  - Typically, no.
    - The intercept is typically not a sign of overfitting, it indicates the global intercept
  - A common alternative: center the data around zero (Y is mean zero), regularize all coefficients
    - This can simplify implementation

Outline

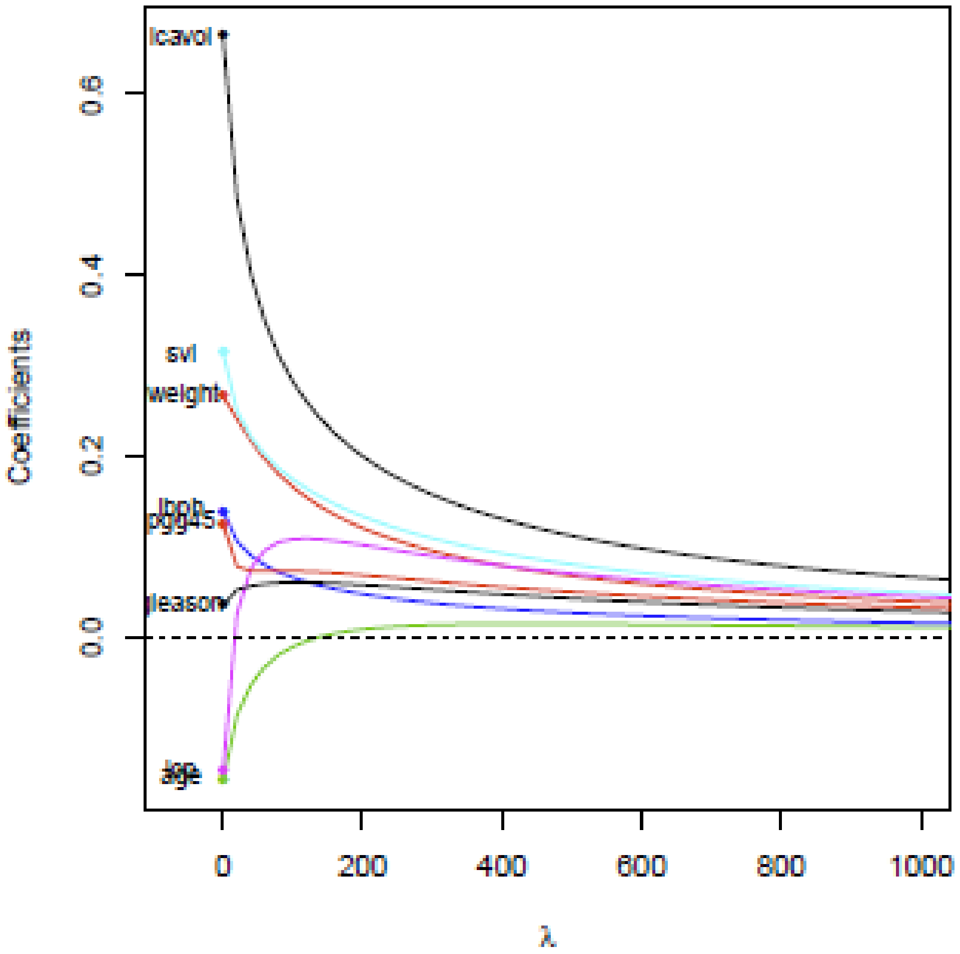
- Overfitting
- Regularization: Intuition and overview
- **Ridge**
- Lasso
- Logistic regression: Intro

"Ridge"

$$J(\theta) = \frac{1}{2N}\left[\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \ldots + \theta_k X_i^k - Y_i\right)^2 + \lambda \sum_{j=1}^{k} \boxed{\theta_j^2}\right]$$

- L_2 norm (ridge regression): penalty proportional to θ^2
  - Works best when a subset of the true coefficients are small
  - Will never set coefficients to zero exactly
  - Cannot perform variable selection in the linear model
  - Coefficients don't have same natural interpretation as OLS
  - Convex and differentiable

Ridge: Coefficient plot



Source: Ryan Tibshirani

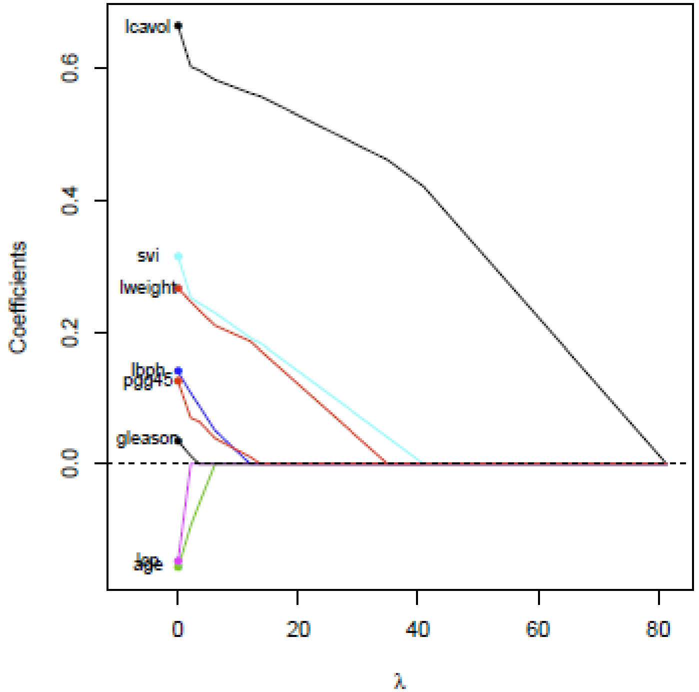
LASSO

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ \dots + \theta_k X_i^k - Y_i\right)^2 + \lambda\sum_{j=1}^{k}|\theta_j|$$

- L_1 norm (lasso regression): penalty proportional to $|\theta|$
  - Selects more relevant features and discards the others, vs. Ridge regression which reduces parameters but doesn't drive to zero
  - Not differentiable
  - Coefficients still difficult to interpret, though "post-lasso" versions can reduce bias (e.g., Belloni & Chernozhukov)

LASSO: Coefficient plot

- Least Absolute Selection and Shrinkage Operator
  - See ESL section 3.4
  - Tibshirani (1996), “Regression Shrinkage and Selection via the Lasso"



Other forms of Regularization

- Elastic net: combines Lasso and Ridge w/two hyperparameters (α, λ)

$$J(\theta) = \frac{1}{2N}\sum_{i=1}^{N}\left(\theta_0 + \theta_1 X_i + \ ... + \theta_k X_i^k - Y_i\right)^2 + \alpha\lambda\sum_{j=1}^{k}|\theta_j| + (1-\alpha)\lambda\sum_{j=1}^{k}\theta_j^2$$

penalty	function	optimizer	reference
ridge	$p(x_j) = \lambda x_j^2$	glmnet, ista	(Hoerl & Kennard, 1970)
lasso	$p(x_j) = \lambda \lvert x_j \rvert$	glmnet, ista	(Tibshirani, 1996)
adaptiveLasso	$p(x_j) = \frac{1}{w_j}\lambda \lvert x_j \rvert$	glmnet, ista	(Zou, 2006)
elasticNet	$p(x_j) = \alpha\lambda \lvert x_j \rvert + (1-\alpha)\lambda x_j^2$	glmnet, ista	(Zou & Hastie, 2005)
cappedL1	$p(x_j) = \lambda \min(\lvert x_j \rvert, \theta); \theta > 0$	glmnet, ista	(Zhang, 2010)
lsp	$p(x_j) = \lambda \log(1 + \lvert x_j \rvert/\theta); \theta > 0$	glmnet, ista	(Candès et al., 2008)
scad	$p(x_j) = \begin{cases} \lambda \lvert x_j \rvert & \text{if } \lvert x_j \rvert \leq \lambda \\ \frac{-x_j^2 + 2\theta\lambda \lvert x_j \rvert - \lambda^2}{2(\theta-1)} & \text{if } \lambda < \lvert x_j \rvert \leq \lambda\theta \ ; \theta > 2 \\ (\theta+1)\lambda^2/2 & \text{if } \lvert x_j \rvert \geq \theta\lambda \end{cases}$	glmnet, ista	(Fan & Li, 2001)
mcp	$p(x_j) = \begin{cases} \lambda \lvert x_j \rvert - x_j^2/(2\theta) & \text{if } \lvert x_j \rvert \leq \theta\lambda \\ \theta\lambda^2/2 & \text{if } \lvert x_j \rvert > \lambda\theta \end{cases} ; \theta > 0$	glmnet, ista	(Zhang, 2010)

Outline

- Overfitting
- Regularization: Intuition and overview
- Ridge
- Lasso
- **Logistic regression: Intro**

Logistic regression: Basics

- Logistic regression
  - Models the (linear) relationship between one or more independent variables and one **binary** dependent variable
  - As with linear regression, can be used for inference and prediction; used to predict (and classify) binary outcomes

Inference	**Prediction**
What is the effect of an additional year of schooling on whether an individual is eligible for welfare?	Do we predict that an individual with 6 years of education will be eligible for welfare?
Why did the UCB datacenter go down this week?	Will the datacenter go down next week?
How big a factor is “home court advantage” in whether our team will win or lose?	Are we going to win this week?

Logistic Regression: Idea

- Logistic Regression: Model
  - The logistic regression model assumes that the independent variables have a linear relationship with the logit transformation of the dependent variable
    - i.e., $\mathrm{logit}(y) = \alpha + \beta X + \cdots$

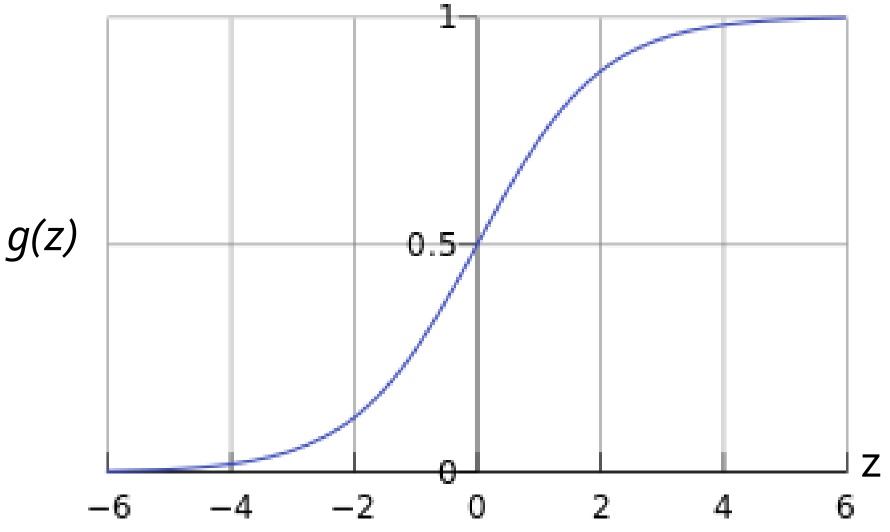
Logistic Regression: Idea

- Logit transformation maps probabilities to log of odds ratios
  - Odds ratio: probability success / probability failure, or $\frac{p}{1-p}$
  - Example: Probability success = 0.8
    - Odds ratio is 4
    - "Odds of success are 4 to 1"
- With logistic regression, our model/hypothesis is that the log of odds ratio is linear function of independent variables
  - $\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \alpha + \beta X + \cdots$
  - Rewritten, this implies $p = \frac{e^{\alpha+\beta X+\cdots}}{1+e^{\alpha+\beta X+\cdots}}$, or more generally, $= \frac{e^{z}}{1+e^{z}}$

Logistic Regression: The logistic function

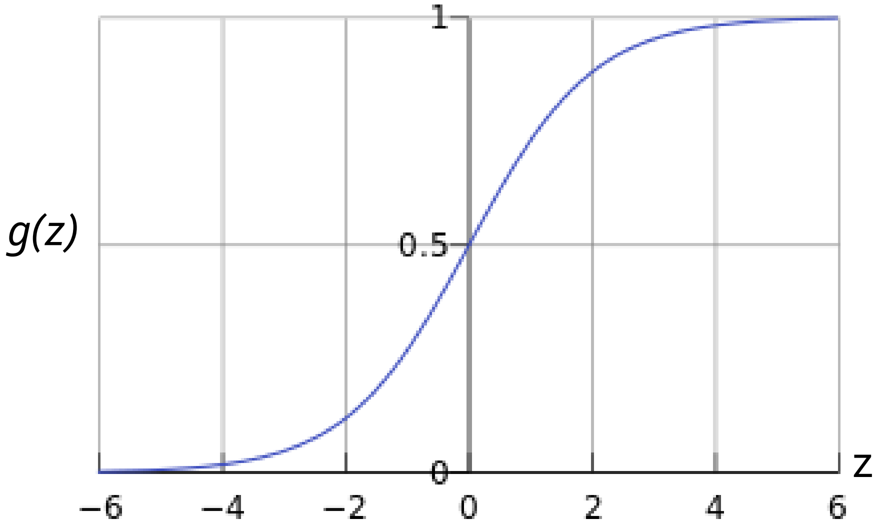
- What does this (sigmoid) function do? $g(z) = \frac{e^z}{1+e^z} = \frac{1}{1+e^{-z}}$
  - Transforms [-∞,+∞]=>[0,1]
  - "Squashing function": constrains output to be between 0 and 1
  - (We'll come back to the sigmoid function when we talk about neural nets)

- In logistic regression,
  - Z is a linear function of parameters
    - i.e., $z = \alpha + \beta X + \ldots$
  - In other words, $g(z) = \frac{e^{\alpha+\beta X+\cdots}}{1+e^{\alpha+\beta X+\cdots}}$



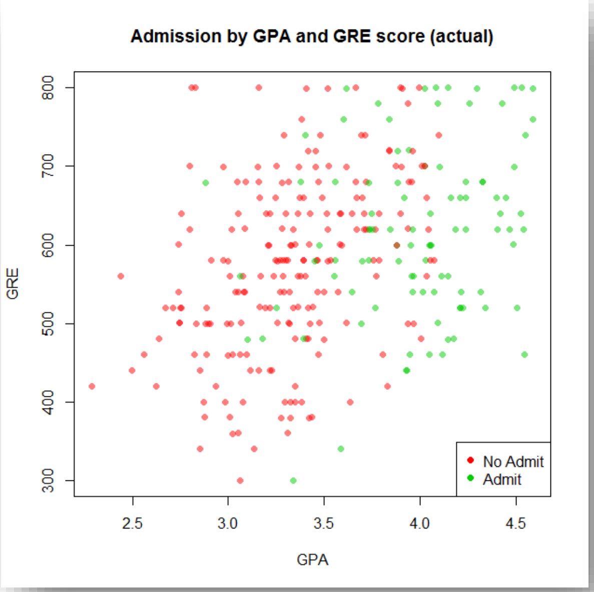
Logistic Regression: Decision Boundary

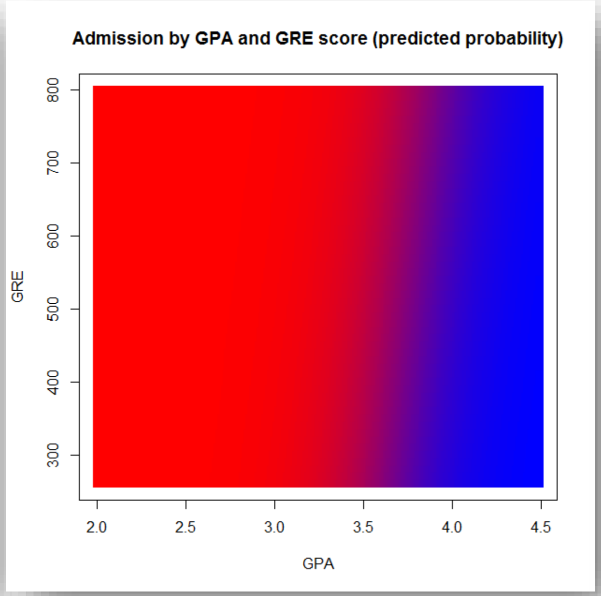
- How to interpret g(z)?
  - Probability that y=1
  - $P(y = 1|x; \alpha, \beta)$
- Simple classifier
  - Predict y=1 if g(z)≥0.5
  - Predict y=0 if g(z)<0.5
- How does this relate to values of z?
  - Predict y=1 if z≥0
  - Predict y=0 if z<0

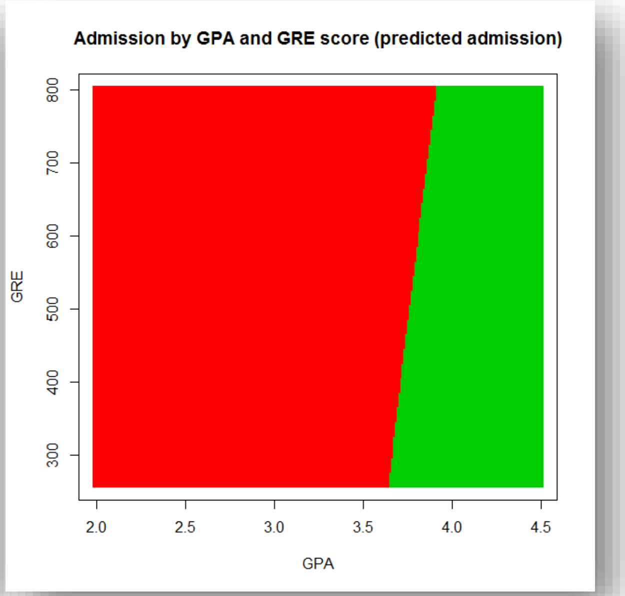


Logistic Regression: Example

- Example: admission vs. GRE and GPA
  1. Start with raw data
  2. Fit logistic regression
  3. Threshold converts g(z) to classification







Logistic Regression: Coefficients

- How do we interpret the coefficients from a logistic regression?
  - The coefficient tells you what change to expect in the *log odds ratio* of your dependent variable, for a one-unit increase in your independent variable.

- Ways to make this more intelligible
  - Convert from log odds ratio to odds ratio
    - $\exp(\beta)$
  - Convert from odds ratio to probability
    - $\frac{odds}{1+odds}$

Logistic Regression: Coefficients

- Example with no predictor variables
  - Likelihood of being honor student (model with intercept and no regressors)%
    - $\text{logit}(honor_i) = \alpha + \epsilon_i$

```
Logistic regression                               Number of obs   =        200
                                                  LR chi2(0)      =       0.00
                                                  Prob > chi2     =          .
Log likelihood = -111.35502                       Pseudo R2       =     0.0000

------------------------------------------------------------------------------
         hon |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
   intercept |   -1.12546   .1644101    -6.85   0.000    -1.447697   -.8032217
------------------------------------------------------------------------------
```

    - i.e., log(p/(1-p)) = -1.12546

```
        hon |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        151       75.50       75.50
          1 |         49       24.50      100.00
------------+-----------------------------------
      Total |        200      100.00
```

  - Note that p = exp(-1.12546)/(1+exp(-1.12546)) = .245

Logistic Regression: Coefficients

- Example with single predictor variable
  - Likelihood of honor student, by major
    - $\text{logit}(honor_i) = \alpha + \beta STEM_i + \epsilon_i$
    - Are STEM students more likely to be honors?
    - How much more likely?
  - exp(0.593) = 1.809
    - (this is the odds ratio)
  - The odds radio can also be seen in the cross-tabs:
    - Odds for non-STEM: 0.23 (17/74)
    - Odds for STEM: 0.42 (32/77)
    - Odds for STEM 81% higher
      - 0.42 / 0.23 = 1.809
      - 0.644 / (1-0.644) = 1.809

```
Logistic regression                               Number of obs   =        200
                                                  LR chi2(1)      =       3.10
                                                  Prob > chi2     =     0.0781
Log likelihood = -109.80312                       Pseudo R2       =     0.0139

------------------------------------------------------------------------------
         hon |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        stem |   .5927822   .3414294     1.74   0.083    -.0764072    1.261972
   intercept |  -1.470852   .2689555    -5.47   0.000    -1.997995   -.9437087
------------------------------------------------------------------------------
```

```
           |         stem
       hon |        no        yes |     Total
-----------+----------------------+----------
         0 |        74         77 |       151
         1 |        17         32 |        49
-----------+----------------------+----------
     Total |        91        109 |       200
```