What our model needs to do

- Usually, we are not just trying to explain observed data
- We want to uncover meaningful trends
- And predict future observations

Our questions then are

- Is $\beta$ a good estimate of $\beta$ (consistent, minimizes error)
- Will $X\beta$ fit future observations well (generalizes well)

Ridge regression

- If the $\beta$s are unconstrained
  - They can get very large
  - As they get larger, they are more susceptible to high variance
- Regularize the coefficients: add constraints to keep them small

$$\min_{\beta} \sum_{i=1}^{n} (y_i - X\beta)^2 \text{ s.t. } \sum_{j} \beta_j \leq c$$

- Necessary for $Y$ to be centered, $X$'s to be standardized.
  - Centered: mean 0
  - Standardized: mean 0, variance 1

Ridge Regression: L2 penalty

- New loss function: (instead of MSE or RSS)
  - Penalized residual sum of squares

$$PRSS(\beta)_{\text{ridge}} = \sum_{i=1}^{n} (y_i - X\beta)^2 + \lambda \sum_{j} \beta_j^2$$

- This is a convex optimization problem: There’s a unique solution
  - Solution is a function of $\lambda$

$\lambda$

- $\lambda$ is known as the shrinkage parameter
  - $\lambda$ controls the size the $\beta$ coefficients can take
  - Controls the amount of regularization
  - As $\lambda \downarrow 0$ we obtain $\beta_{\text{OLS}}$
  - As $\lambda \uparrow \infty$ we obtain $\beta_{\text{ridge}} = 0$ (intercept-only model)

Ridge coefficients
Why does Ridge Regression help?

- Bias-variance tradeoff
- We accept bias to turn down the variance.

How do we choose $\lambda$?

- We need a systematic and principled way of choosing $\lambda$.
- We want to choose $\lambda$ that minimizes the PRSS
  - Usually it’s not the OLS solution
- We want to minimize the size of $\beta$s while minimizing the MSE.

How do we choose $\lambda$? (geometric proof)

- The blue ball is the beta contribution, the red the OLS MSE.
- Just like in the two dimensional case, we want the cross over point

Choosing lambda in practice

Lasso Regression

- Ridge regression: keep the size of the $\beta$s small
- Lasso regression: keep the $\beta$s zero.

$$\sum_{j=1}^{p} (y_j - \hat{y}_j)^2 + \lambda_{ridge} \sum_{j=1}^{p} \beta_j^2 = \lambda_{lasso} \sum_{j=1}^{p} |\beta_j|$$

- Similar to Ridge, except different penalty. (and thus different interpretation)

Ridge $\lambda \sum_{j=1}^{p} \beta_j^2$ | Lasso $\lambda \sum_{j=1}^{p} |\beta_j|$
We could show Lasso is biased. Unless $\lambda = 0$.
Analytical solution is less clear than either Ridge or OLS, but it is again a function of $\lambda$.
Similar problem to before, we have to choose $\lambda$.

Lasso

Performance of Lasso

We want the crossover point.

How do we choose $\lambda$? (geometric proof)

- The teal square is the beta contribution, the red the MSE.
- Just like in the two dimensional case, we want the crossover point.
Performance: Ridge vs Lasso

Probability of seeing a given value of $\beta$

Machine learning overview

Deciding which algorithm to use

- First consideration: (supervised or unsupervised)
  Is there an underlying truth? Do you know exactly what values $Y$ should be taking?
- Second consideration: (classification vs regression)
  Are you predicting a probability, class/cluster or value
- Third consideration:
  Are you interested in predictive accuracy or understanding influence of predictors

A whirlwind tour of some common algorithms
(not instruction, just exposure... come back Spring ’18 for instruction)
Unsupervised vs supervised

- Mostly we’ll work in supervised....
- Considering both regression and classification
- With parametric models

Regression/Classification/Clustering

- Values vs labels
- All regression problems can be turned into a classification problem (e.g. Does the house cost more than 100k)
- Classification problems are either predicting labels or predicting probabilities of events
- Clustering is detecting communities/groups within a dataset

Linear regression

Logistic Regression

K nearest neighbors

K means clustering
Naïve Bayes Classifier

- Assume everything is independent given class.

![Naïve Bayes Classifier Diagram]

Support Vector Machines

- Support Vector Machines: Kernels

![Support Vector Machines: Kernels Diagram]

Artificial Neural Networks

![Artificial Neural Networks Diagram]