LECTURE 12:

LINEAR MODEL SELECTION PT. 3

October 23, 2017

SDS 293: Machine Learning

Announcements 1/2

Computer Science

Silvana, Artemis, Marina and Kyra present their research posters at the Collaborations event, 4/22/17.

Presentation of the
CS Major & Minors

TODAY @ lunch
Ford 240
FREE FOOD!

Announcements 2/2

CS Internship Lunch Presentations

Come hear where Computer Science majors interned in Summer 2017!

Employers range from companies in the tech industry to research labs.

All are welcome! Pizza lunch provided.

Thursday,
October 26th
12:10 - 1 pm
Ford Hall 241

Extra credit opportunity
Want to drop a missing lab? Attend and post to #talks!

Outline

- Model selection: alternatives to least-squares

✓Subset selection

  ✓Best subset

  ✓Stepwise selection (forward and backward)

  ✓Estimating error using cross-validation

- Shrinkage methods
  - Ridge regression and the Lasso
  - Dimension reduction
- Labs for each part

Flashback: subset selection

- **Big idea:** if having too many predictors is the problem maybe we can get rid of some

- Three methods:
  - **Best subset:** try all possible combinations of predictors
  - **Forward**: start with no predictors, greedily add one at a time
  - **Backward**: start with all predictors, greedily remove one at a time

Common theme of subset selection:

ultimately, individual predictors are either **IN** or **OUT**

Discussion

- **Question:** what potential problems do you see?
- **Answer:** we're exploring the space of possible models as if there were only finitely many of them, but there are actually infinitely many (why?)

New approach: “regularization”

subset selection

$$Y \approx \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

constrain the coefficients

Another way to phrase it:

reward models that **shrink** the coefficient estimates **toward zero**

(and still perform well, of course)

Approach 1: ridge regression

- **Big idea**: minimize RSS plus an additional penalty that rewards small (sum of) coefficient values

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)^2 + \lambda \sum_{j=1}^{p}\beta_j^2$$

RSS

Shrinkage penalty

Rewards coefficients close to zero

Sum over all observations

Observed value

Predicted value

Tuning parameter

Sum over all predictors

* In statistical / linear algebraic parlance, this is an ℓ_2 penalty

Approach 1: ridge regression

- For each value of λ, we only have to fit one model

RSS

Shrinkage penalty

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right) + \lambda\sum_{j=1}^{p}\beta_j^2$$

Tuning parameter

- Substantial computational savings over best subset!

Approach 1: ridge regression

- **Question**: what happens when the tuning parameter is **small**?

RSS

Shrinkage penalty

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right) + \lambda \sum_{j=1}^{p}\beta_j^2$$

Tuning parameter

- **Answer:** just minimizing RSS; simple least-squares

Approach 1: ridge regression

- **Question**: what happens when the tuning parameter is **large**?

RSS

Shrinkage penalty

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)}_{\text{RSS}} + \lambda \underbrace{\sum_{j=1}^{p}\beta_j^2}_{\text{Shrinkage penalty}}$$

Tuning parameter

- **Answer:** all coefficients go to zero; turns into null model

Ridge regression: caveat

- RSS is scale-invariant*
- **Question**: is this true of the shrinkage penalty?

$$\underbrace{\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right)}_{\text{RSS}} + \lambda \underbrace{\sum_{j=1}^{p}\beta_j^2}_{\text{Shrinkage penalty}}$$

- **Answer:** no! This means having predictors at different scales would influence our estimate… need to first **standardize** the predictors by dividing by the standard deviation

* multiplying any predictor by a constant doesn't matter

Discussion

- **Question:** why would ridge regression improve the fit over least-squares regression?
- **Answer:** as usual, comes down to **bias-variance tradeoff**
  - As λ increases, flexibility decreases: ↓ variance, ↑ bias
  - As λ decreases, flexibility increases: ↑ variance, ↓ bias
  - **Takeaway:** ridge regression works best in situations where least squares estimates have high variance: trades a small increase in bias for a large reduction in variance

So what's the catch?

- Ridge regression doesn't actually perform variable selection
- Final model will include **all predictors**
  - If all we care about is **prediction accuracy**, this isn't a problem
  - It does, however, pose a challenge for **model interpretation**
- If we want a technique that actually performs variable selection, **what needs to change**?

Approach 2: the lasso

- **(same) Big idea**: minimize RSS plus an additional penalty that rewards small (sum of) coefficient values

RSS

Shrinkage penalty

Rewards coefficients close to zero

$$\sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{ij}\right) + \lambda \sum_{j=1}^{p}\left|\beta_j\right|$$

Tuning parameter

* In statistical / linear algebraic parlance, this is an ℓ_1 penalty

Discussion

- **Question:** why does that enable us to get coefficients exactly equal to **zero**?

Answer: let's reformulate a bit

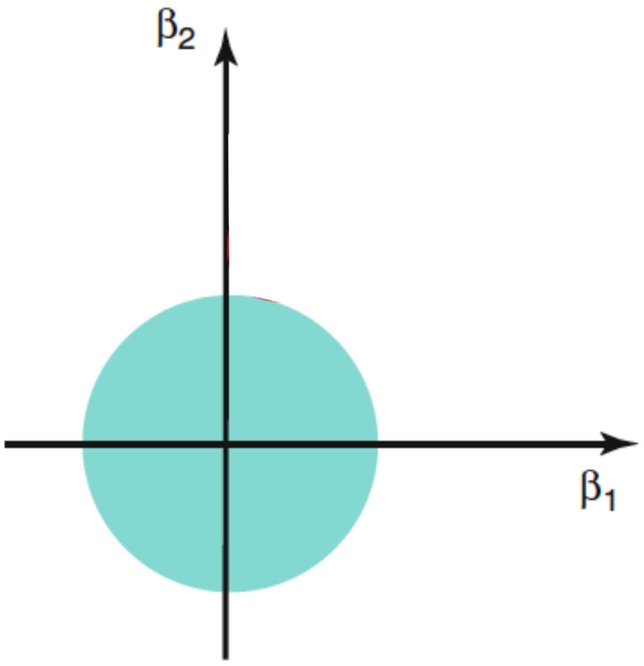
- For each value of λ, there exists a value for s such that:
- Ridge regression:

$$\min_{\beta}(RSS) \text{ subject to } \sum_{j=1}^{p} \beta_j^2 \leq s$$

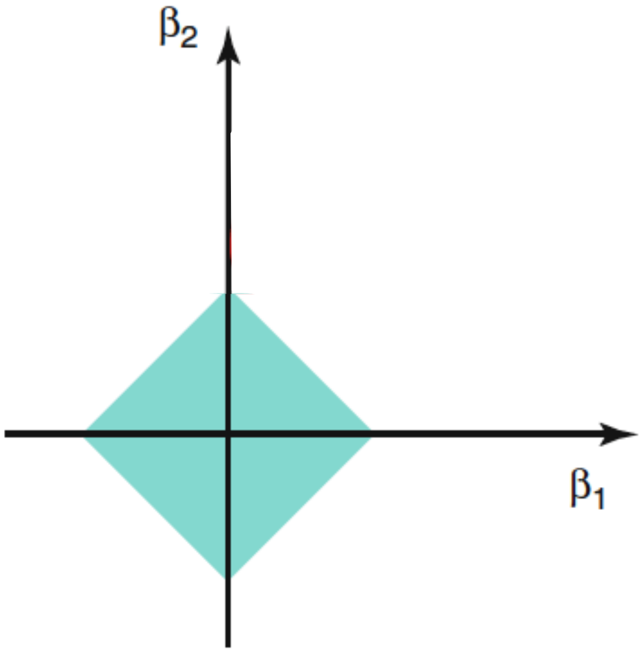
- Lasso:

$$\min_{\beta}(RSS) \text{ subject to } \sum_{j=1}^{p} |\beta_j| \leq s$$

Comparting constraint functions

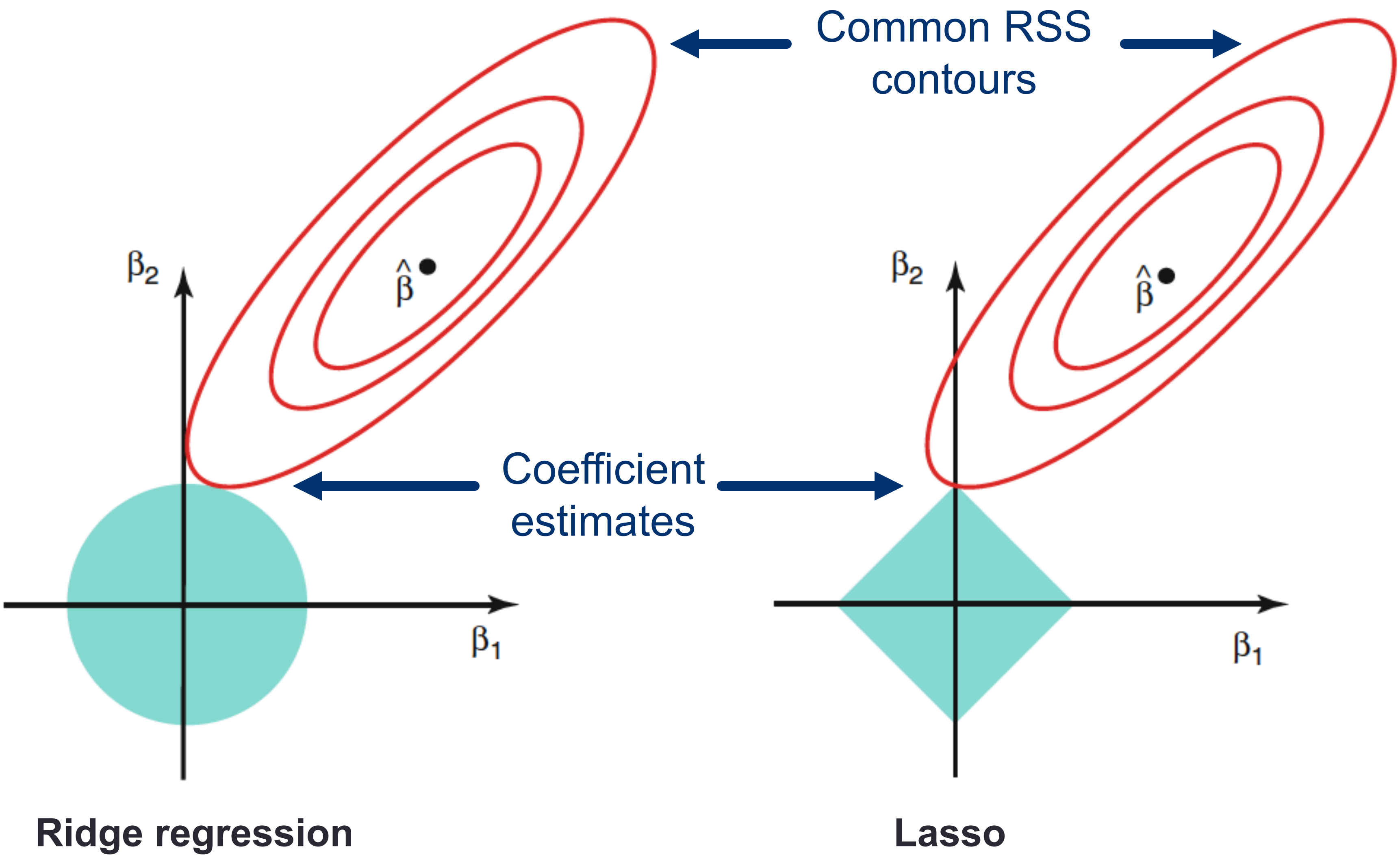


Ridge regression



Lasso

Comparing constraint functions



Comparing ridge regression and the lasso

- Efficient implementations for both (in R and python!)
- Both significantly reduce variance at the expense of a small increase in bias
- **Question**: when would one outperform the other?
- **Answer:**
  - When there are relatively many equally-important predictors, **ridge regression** will dominate
  - When there are small number of important predictors and many others that are not useful, **the lasso** will win

Lingering concern…

- **Question:** how do we choose the right value of λ?
- **Answer:** sweep and cross validate!
  - Because we are only fitting a single model for each λ, we can afford to **try lots of possible values** to find the best (“sweeping”)
  - For each λ we test, we’ll want to calculate the **cross-validation error** to make sure the performance is consistent

Lab: ridge regression & the lasso

- To do today's lab in R: `glmnet`
- To do today's lab in python: <nothing new>
- Instructions and code:

  [course website]/labs/lab10-r.html

  [course website]/labs/lab10-py.html

- Full version can be found beginning on p. 251 of ISLR

Coming up

- Jordan is traveling next week
- Guest lectures:
  - Tuesday: "Data Wrangling in Python" with Ranysha Ware, MITLL
  - Thursday: "ML for Population Genetics" with Sara Mathieson, CSC