# Data Mining (W4240 Section 001)
# Shrinkage

Giovanni Motta

Columbia University, Department of Statistics

November 11, 2015

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Subset Selection

**Pick the best $k$ ($\leq p$) covariates to use in linear regression**

# Subset Selection

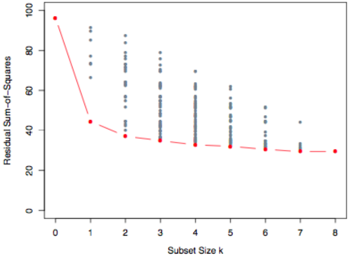
**Pick the best $k$ ($\leq p$) covariates to use in linear regression**

Why?

- *Predictive Accuracy:* Linear least squares estimator has low bias, high variance. Reduce number of covariates, get a bit more bias but much less variance.
- *Interpretability:* Which variables matter? Which do not? Interpretability allows your model to say something about the data vs. just giving a prediction.

# Subset Selection

How to pick the best $k$ ($\leq p$) covariates for linear regression?



Best Subset Selection:

- enumerate possible subsets in a smart way for each $k$
- for each $k$, select subset that minimizes RSS
- pick best $k$: cross-validation or other model selection methods
- good method for $p <$ 30 or 40

# Model Selection

Rather than enumerating all possible subsets, model selection can be done in a few ways

- Cross-validation:
    - possibly more accurate
    - no need for asymptotic approximations (is $n$ large enough to justify asymptotics?)
    - more flexible (can be used for things other than MLE)
- Model selection criteria (AIC, BIC, etc.):
    - often easy to compute
    - theoretically justifiable

# Model Selection

Rather than enumerating all possible subsets, model selection can be done in a few ways

- Cross-validation:
  - possibly more accurate
  - no need for asymptotic approximations (is $n$ large enough to justify asymptotics?)
  - more flexible (can be used for things other than MLE)
- Model selection criteria (AIC, BIC, etc.):
  - often easy to compute
  - theoretically justifiable

- **Today we will discuss shrinkage, a similar approach:**
  - can sometimes remove covariates (perform subset selection)
  - more generally, reduces estimator variance and complexity

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Multivariate Linear Regression

*Recall:*

The *design matrix* is an $n \times p$ matrix:

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \dots & x_{1,p} \\ x_{2,1} & \dots & x_{2,p} \\ \vdots & & \vdots \\ x_{n,1} & \dots & x_{n,p} \end{bmatrix}$$

The *response vector* is an $n \times 1$ column vector:

$$\mathbf{y} = \left[y_1 \; y_2 \; \dots \; y_n\right]^\top$$

The *parameter vector* is a $p \times 1$ column vector (as before):

$$\boldsymbol{\beta} = \left[\beta_1 \; \dots \; \beta_p\right]^\top$$

Today, we will center and scale the data; not use intercept

# Reminder: Centering/Scaling for Linear Regression

Scaling: we do not *need* an intercept

- rescale data:

$$\tilde{X}^\top = \left( \frac{X_1 - \bar{X}_1}{\hat{\sigma}_1}, \frac{X_2 - \bar{X}_2}{\hat{\sigma}_2}, \dots, \frac{X_p - \bar{X}_p}{\hat{\sigma}_p} \right),$$

$$\tilde{Y} = \frac{Y - \bar{Y}}{\hat{\sigma}_Y}$$

- all elements of rescaled data have mean 0... so no need for intercept
- in R, use the function `scale()` (ex: `> x.bar <- scale(x)`)
- then fit linear function

Note: we need to rescale both $X$ and $Y$! (Why is this?)

# Multivariate Linear Regression

Recall least squares regression:

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= \arg\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2 \\ &= \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \mathbf{X}^\top \mathbf{y}\end{aligned}$$

Problems:

- when the true relationship between $Y$ and $X$ is linear, the LS estimates will have low bias but high variance
- need at least $p$ observations, otherwise $(\mathbf{X}^\top \mathbf{X})^{-1}$ does not exist

# Multivariate Linear Regression and subset selection

Idea: if we have a large number of covariates compared to observations, say $n < 2p$, the best you can do is to

**estimate most coefficients as 0!**

- not enough info to determine all coefficients
- try to estimate ones with strong signal
- set everything else to 0 (or close)

Coefficients of 0 may not be a bad assumption...

*If we have 1,000s of coefficients, are they all equally important?*

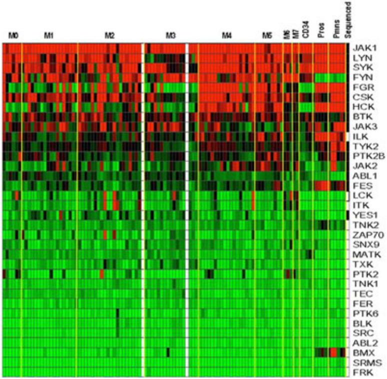
# Gene Expression

*Example: microarray gene expression data*

- gene expression: want to measure the level at which information in a gene is used in the synthesis of a functional gene product (usually protein)
- can use gene expression data to determine subtype of cancer (e.g. which *type* of Lymphoma B?) or predict recurrence, survival time, etc
- problem: thousands of genes, hundreds of patients, $p > n$!

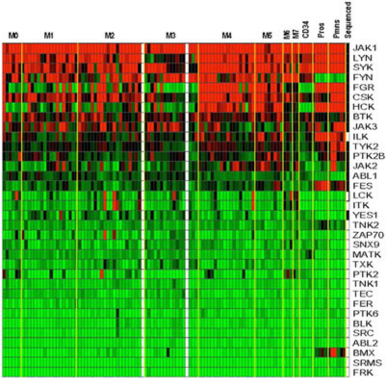
Intuition: only a handful of genes should affect outcomes

# Gene Expression



- gene expression levels are continuous values
- data: observation $i$ is gene expression levels from patient $i$, attached to outcome for patient (survival time)
- covariates: expression levels for $p$ genes

# Gene Expression



- collinearity: does it matter *which* gene is selected for *prediction*? No!
- overfitting: now fitting $p'$ non-0 coefficients to $n$ observations with $p' << n$ means less fitting of noise

# Forward Stepwise Linear Regression

Forward stepwise linear regression:

- sequentially adds in predictors based on $F-$statistics (or AIC or BIC or adjusted R$^2$)
- can handle data with $p > n$
- ...but problems with multiple testing ($F-$statistic computed on same data again and again)
- ...and a lot of parameter bias (either 0 or much greater magnitude than it should be)
- ...and model selection can be unstable

Is there a more principled way to force (**shrink**) values to 0?

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Regularized Linear Regression

**Regularization (a hugely important concept):**

- place a *penalty* on large values for $\beta_1$, ..., $\beta_p$ (why not $\beta_0$? can always easily estimate mean)
- add this penalty to the objective function
- solve for $\hat{\boldsymbol{\beta}}$!

New objective function:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \text{penalty}(\beta_j) \right]$$

$\lambda$ acts as a weight on penalty: low values mean few coefficients near 0, high values mean many coefficients near 0

# Regularized Linear Regression

New objective function:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \text{penalty}(\beta_j) \right]$$

**When and why can this be a better predictor?**

- It adds bias (we are not fitting the best $\boldsymbol{\beta}$ to the data)
- ...but it *greatly* reduces variance

So that beta only reflects the true importance and the penalty are same

Note 1: the data *need* to be centered and scaled. Why?

Note 2: will this always be a better predictor? Why not?

# Regularized Linear Regression

Suppose we have an estimator $\hat{f}(\mathbf{z}) = \mathbf{z}^\top \hat{\boldsymbol{\beta}}$. Two questions:

1. Is $\hat{\boldsymbol{\beta}}$ close to the true $\boldsymbol{\beta}$?
2. Will $\hat{f}(\mathbf{z})$ fit future observations well?

MSE of our estimate:

$$MSE(\hat{\boldsymbol{\beta}})_{\boldsymbol{\beta}} = \mathbb{E}[\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}\|^2] = \mathbb{E}[(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})]$$

MSE of the OLS estimator (answer to the first question):

$$\mathbb{E}[(\hat{\boldsymbol{\beta}}_{OLS} - \boldsymbol{\beta})^\top (\hat{\boldsymbol{\beta}}_{OLS} - \boldsymbol{\beta})] = \sigma^2 \mathrm{tr}[(\mathbf{Z}^\top \mathbf{Z})^{-1}]$$

# Will $\hat{f}(\mathbf{z})$ fit future observations well?

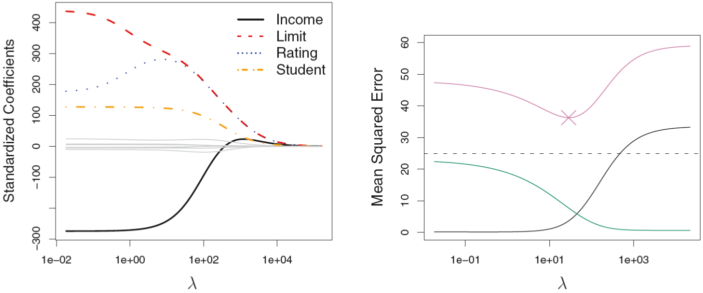
- Just because $\hat{f}(\mathbf{z})$ fits our data well, this doesnt mean that it will be a good fit to new data
- suppose that we take new measurements Y: $(\mathbf{z}_1, Y_1), \ldots, (\mathbf{z}_n, Y_n)$
- So if $\hat{f}$ is a good model, then $\hat{f}(\mathbf{z}_i)$ should also be close to $Y_i$

**Prediction error** (PE)

$$
\begin{aligned}
PE(\mathbf{z}_0) &= \mathbb{E}[\big(Y - \hat{f}(\mathbf{Z})\big)^2 | \mathbf{Z} = \mathbf{z}_0] \\
&= \underline{\sigma_\epsilon^2 + [Bias\hat{f}(\mathbf{z}_0)]^2 + \mathbb{V}\mathrm{ar}[\hat{\mathrm{f}}(\mathbf{z}_0)]}
\end{aligned}
$$

- As model becomes more complex (more terms included), local structure/curvature can be picked up. However, coefficient estimates suffer from higher variance
- Introducing a little bias in our estimate for $\boldsymbol{\beta}$ might lead to a substantial decrease in variance, and hence to a substantial decrease in the PE

# Ridge Regression



- Left: credit card data
- Right: simulated data with $n = 50$ and $p = 45$

# Regularized Linear Regression

- Best Subset Selection (card penalty): #parameters as penalty term

$$\hat{\boldsymbol{\beta}}^{BSS} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \underline{\sum_{j=1}^{p} \mathbf{1}_{\{\beta_j \neq 0\}}} \right]$$

- Ridge regression (squared penalty): Easy to work with

$$\hat{\boldsymbol{\beta}}^{Ridge} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right]$$

- Lasso regression (absolute value penalty):

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

# Regularized Linear Regression

- Best Subset Selection ($\mathrm{card}$ penalty):

$$\hat{\boldsymbol{\beta}}^{BSS} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \mathbf{1}_{\{\beta_j \neq 0\}} \right]$$

- Ridge regression (squared penalty):

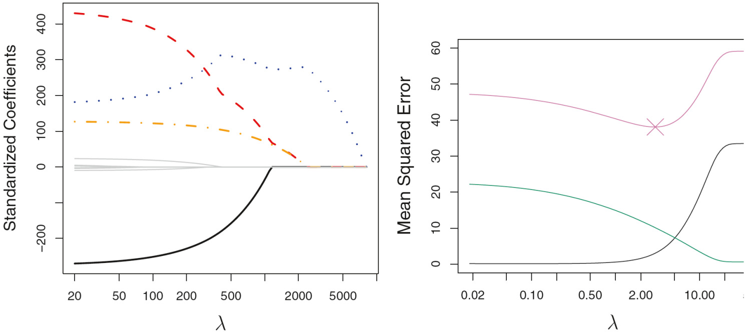
$$\hat{\boldsymbol{\beta}}^{Ridge} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right]$$

- Lasso regression (absolute value penalty):

$$\hat{\boldsymbol{\beta}}^{Lasso} = \arg\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right]$$

Ridge and Lasso regression produce estimators with different properties

# Ridge Regression



- Left: credit card data
- Right: (same) simulated data with $n = 50$ and $p = 45$

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection ≈ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Ridge Regression

Geometrically, what does a squared penalty do?

$$\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \boxed{\lambda \sum_{j=1}^{p} \beta_j^2} \right]$$

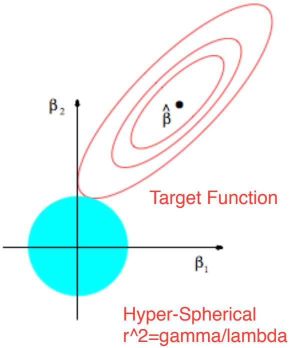
is equivalent to

$$\min_{\boldsymbol{\beta}, \gamma} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \gamma \right]$$

$$\text{subject to} : \lambda \sum_{j=1}^{p} \beta_j^2 \leq \gamma$$

# Ridge Regression

The estimator $\hat{\boldsymbol{\beta}}^{Ridge}$ is given by the first point at which an ellipse contacts the constraint region



$\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2$: residual sum of squares

$\lambda \sum_{j=1}^{p} \beta_j^2 \leq \gamma$: coefficients restricted sphere with radius $\sqrt{\frac{\gamma}{\lambda}}$

# Ridge Regression

Ridge Reg always have a closed form solution

$$\begin{aligned}
\mathrm{PRSS}(\boldsymbol{\beta}) &= (\mathbf{y}-\mathbf{X}\boldsymbol{\beta})^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_2^2 \\
\frac{\partial \mathrm{PRSS}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= -2\mathbf{X}^\top(\mathbf{y}-\mathbf{X}\boldsymbol{\beta}) + 2\lambda\boldsymbol{\beta} \\
\hat{\boldsymbol{\beta}}^{Ridge} &= \left(\mathbf{X}^\top\mathbf{X} + \underline{\lambda\mathbf{I}_p}\right)^{-1}\mathbf{X}^\top\mathbf{y}
\end{aligned}$$

- $\hat{\boldsymbol{\beta}}^{Ridge}$ takes values *near* 0, but not exactly 0 ($\lambda \to \infty$)
- We have a closed form solution for $\hat{\boldsymbol{\beta}}^{Ridge}$.
- the matrix $\left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}_p\right)^{-1}$ *always* exists

# Ridge Regression

Bias and Variance

$$\mathbb{E}[\hat{\boldsymbol{\beta}}^{Ridge}] = \left[\mathbf{I}_p + \lambda(\mathbf{X}^\top\mathbf{X})^{-1}\right]\boldsymbol{\beta}$$

$$\mathbb{V}\mathrm{ar}(\boldsymbol{\theta}^\top\hat{\boldsymbol{\beta}}_\lambda^{\mathrm{Ridge}}) \leq \mathbb{V}\mathrm{ar}(\boldsymbol{\theta}^\top\hat{\boldsymbol{\beta}}^{\mathrm{OLS}})$$

# Ridge Regression and PCA

Let $n > p = \mathrm{rk}(X)$. Singular value decomposition of $\mathbf{X}$

$$\underset{n\times p}{\mathbf{X}} = \underset{n\times p}{\mathbf{V}}\ \underset{p\times p}{\mathbf{\Lambda}^{1/2}}\ \underset{p\times p}{\mathbf{L}^\top}$$

where

- $\underset{p\times p}{\mathbf{L}}$ = eigenvectors of $\underset{p\times p}{\mathbf{X}^\top \mathbf{X}}$
- $\underset{p\times p}{\mathbf{\Lambda}}$ = eigenvalues
- $\underset{n\times p}{\mathbf{V}}$ = ortho-normal eigenvectors of $\underset{n\times n}{\mathbf{X}\,\mathbf{X}^\top}$

# OLS, Ridge Regression, & PCA

$$
\begin{aligned}
\hat{\mathbf{y}}^{OLS} = \mathbf{X}\hat{\boldsymbol{\beta}}^{OLS} &= \mathbf{X}(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{X}^\top\mathbf{y} \\
&= \mathbf{V}\mathbf{V}^\top\mathbf{y} \\
\hat{\mathbf{y}}_\lambda^{Ridge} = \mathbf{X}\hat{\boldsymbol{\beta}}_\lambda^{Ridge} &= \mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top\mathbf{y} \\
&= \mathbf{V}\boldsymbol{\Lambda}^{1/2}(\boldsymbol{\Lambda} + \lambda\mathbf{I})^{-1}\boldsymbol{\Lambda}^{1/2}\mathbf{V}^\top\mathbf{y} \\
&= \sum_{j=1}^{p}\mathbf{v}_j \underline{\left[\frac{\lambda_j}{\lambda_j + \lambda}\right]} \mathbf{v}_j^\top\mathbf{y} \\
&= \mathbf{V}\mathrm{diag}\{\frac{\lambda_j}{\lambda_j + \lambda}\}\mathbf{V}^\top\mathbf{y} \qquad 1 \le j \le p \\
\hat{\mathbf{y}}_\kappa^{PC} = \mathbf{X}\hat{\boldsymbol{\beta}}_\kappa^{PC} &= \mathbf{V}\mathrm{diag}\{\lambda_j\}\mathbf{V}^\top\mathbf{y} \qquad\qquad 1 \le j \le \kappa
\end{aligned}
$$

# OLS, Ridge Regression, & PCA

- If $\lambda = 0$ then $\hat{\mathbf{y}}^{Ridge} = \hat{\mathbf{y}}^{OLS}$.
- If $\lambda > 0$, then the larger the eigenvalue $\lambda_j$, the less it will be penalized in ridge regression.
- Like LinReg, RidReg computes the coordinates of $\mathbf{y}$ with respect to the orthonormal basis $\mathbf{V}$. It then shrinks these coordinates by the factors $\frac{\lambda_j}{\lambda_j+\lambda}$: **a greater amount of shrinkage is applied to the coordinates of basis vectors with smaller $\lambda_j$.**
- That is, since $\frac{\lambda_j}{\lambda_j+\lambda} \leq 1$, small eigenvalues are penalized the most.
- In contrast, in PCA regression, large singular values are kept intact, and the small ones (after certain number $\kappa$) are completely removed. This would correspond to $\lambda = 0$ for the first $\kappa$ ones and $\lambda = \infty$ for the rest.

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection ≈ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Lasso Regression

Geometrically, what does an absolute value penalty do?

$$\min_{\beta} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^{\top} \boldsymbol{\beta} \right)^2 + \boxed{\lambda \sum_{j=1}^{p} |\beta_j|} \right]$$

Double Exponential
exp(abs(beta)*lamda)

is equivalent to

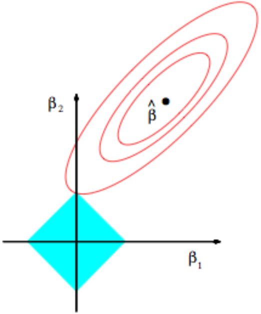
$$\min_{\boldsymbol{\beta},\gamma} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^{\top} \boldsymbol{\beta} \right)^2 + \gamma \right]$$

$$\text{subject to : } \lambda \sum_{j=1}^{p} |\beta_j| \leq \gamma$$

# Lasso Regression

The estimator $\hat{\boldsymbol{\beta}}^{Lasso}$ is given by the first point at which an ellipse contacts the constraint region

It only touches at one point and so it's a bit unstable in iteration



$\sum_{i=1}^{n} \left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2$: residual sum of squares

$\lambda \sum_{j=1}^{p} |\beta_j| \leq \gamma$: coefficients restricted square with radius $\frac{\gamma}{\lambda}$

# Lasso Regression

Reduced the complexity of the model——can be used as a subset selection

- Most of the time, the residual sum of squares is projected onto a vertex
- This forces many coefficient values to *exactly* 0
- Unfortunately, we don't have a closed form solution for $\hat{\boldsymbol{\beta}}^{Lasso}$
- Nevertheless, there are many R functions that solve this problem efficiently (even for large $n$ and $p$) through *convex optimization*

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

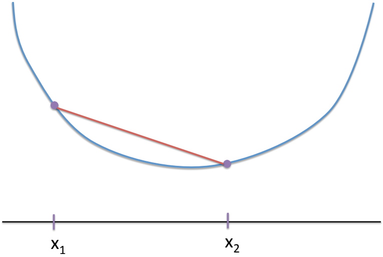
Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Convex Optimization

A function is *convex* if
$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$



Can minimize with hill-climbing algorithms and you are *guaranteed* to get optimal decision

# Convex Optimization

Convex optimization problem:

$$\begin{aligned} \min_{x} \ & f_0(x) \\ \text{subject to } & f_i(x) \leq 0 \\ & \mathbf{Ax} = \mathbf{b} \end{aligned}$$

Objective function: $f_0(x)$ is convex

Constraints: $f_i(x)$ is convex, $\mathbf{Ax} = \mathbf{b}$ is affine (linear)

- $f_i(x) = x^2$ is convex
- $f_i(x) = |x|$ is convex
- $f_i(x) = \mathbf{1}_{\{x \notin \{\ldots,-1,0,1,2,\ldots\}\}}$ is not convex
- $f_i(x) = \mathrm{card}(x)$ (number of non-zero elements) is not convex

# Convex Optimization

Subset selection:

$$\min_{\boldsymbol{\beta}} \left[ \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \boldsymbol{\beta} \right)^2 + \lambda \, \mathrm{card}(\boldsymbol{\beta}) \right]$$

In general,

- if penalty is norm $\|\boldsymbol{\beta}\|_p = \left( \sum_j \beta_j^p \right)^{\frac{1}{p}}$ with $p \geq 1$, then problem is convex
- if penalty is norm $\|\boldsymbol{\beta}\|_p$ with $p < 1$, then problem is not convex
- subset selection is not convex

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection ≈ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

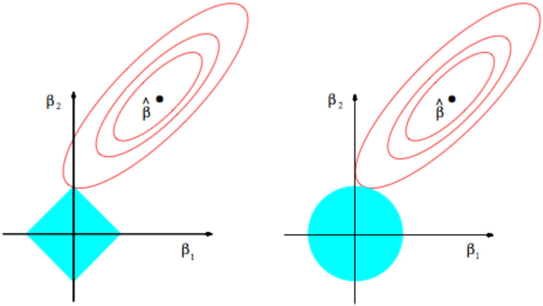
Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Comparing Ridge and Lasso



Ridge is *stable* to small changes in $\mathbf{X}$ and $\mathbf{y}$; Lasso is not (might be projected onto different vertex)

# Comparing Ridge and Lasso

| | **Ridge** | **Lasso** |
|---|---|---|
| Objective | $\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2 + \lambda \sum_{j=0}^{p} \beta_j^2$ | $\sum_{i=1}^{n}\left(y_i - \mathbf{x}_i^\top \boldsymbol{\beta}\right)^2 + \lambda \sum_{j=0}^{p} \lvert\beta_j\rvert$ |
| Estimator | $\left(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^\top\mathbf{y}$ | no closed form |
| Coefs | most close to 0 | When a parameter hits 0, it stays there<br>most exactly 0 |
| Stability | robust to changes in $\mathbf{X}$, $\mathbf{y}$ | not robust to changes in $\mathbf{X}$, $\mathbf{y}$ |

Regularized linear regression is fantastic for low signal datasets or those with $p >> n$

- Ridge: good when many coefficients affect value but not large (gene expression)
- Lasso: good when you want an *interpretable* estimator

# Choosing $\lambda$

Both Ridge and Lasso have a tunable parameter, $\lambda$

- use leave-one-out cross validation to find best $\lambda$

$$\hat{\lambda} = \arg\min_{\lambda} \sum_{i=1}^{n} \left( y_i - \mathbf{x}_i^\top \hat{\boldsymbol{\beta}}_{-i,\lambda} \right)^2$$

- this is really slow for large datasets
- have closed form approximation called *generalized cross validation*
- `R` functions implement this to automatically choose $\lambda$ for you

# Outline

Reminder from last time: Subset Selection

Motivation: subset Selection $\approx$ Shrinking coefficients

Regularization

Regularization 2: Ridge Regression

Regularization 1: LASSO Regression

Aside: Convex Optimization

Ridge Regression vs Lasso Regression

Example: Prostate Data

# Regularized Linear Regression in R

Read in data about prostate cancer:

```
> prostate <- read.csv("Prostate.csv")
> names(prostate)
 [1] "lcavol"  "lweight" "age"     "lbph"
 [5] "svi"     "lcp"     "gleason" "pgg45"
 [9] "lpsa"    "train"
```

Predictors (columns 1–8):
lcavol lweight age lbph svi lcp gleason pgg45

Response (column 9):
lpsa

Training/testing indicator (column 10):
train

# Regularized Linear Regression in R

There are 96 observations and 8 covariates

First, we center and scale the data (mean 0, var $= n$)

```
> xp <- scale(prostate[,1:9])
> prostate[1:2,]
      lcavol  lweight age      lbph svi       lcp gleason
1 -0.5798185 2.769459  50 -1.386294   0 -1.386294       6
2 -0.9942523 3.319626  58 -1.386294   0 -1.386294       6
  pgg45       lpsa train
1     0 -0.4307829  TRUE
2     0 -0.1625189  TRUE
> xp[1:2,]
        lcavol    lweight        age      lbph        svi
[1,] -1.637356 -2.0062118 -1.8624260 -1.024706 -0.5229409
[2,] -1.988980 -0.7220088 -0.7878962 -1.024706 -0.5229409
            lcp   gleason      pgg45      lpsa
[1,] -0.8631712 -1.042157 -0.8644665 -2.520226
[2,] -0.8631712 -1.042157 -0.8644665 -2.287827
```

# Regularized Linear Regression in R

Now break the data into training and testing sets:

```
> xp.train <- xp[(prostate$train==TRUE),]
> xp.test <- xp[(prostate$train==FALSE),]
> dim(xp.train)
[1] 67  9
> dim(xp.test)
[1] 30  9
```

We have 67 training observations and 30 testing observations

# Regularized Linear Regression in R

Let's begin by fitting ordinary least squares and least absolute deviation regression:

```
> xp.train.df <- data.frame(xp.train)
> names(xp.train.df)
[1] "lcavol"  "lweight" "age"     "lbph"    "svi"
[6] "lcp"     "gleason" "pgg45"   "lpsa"
> attach(xp.train.df)
> fit.ols <- lm(lpsa ˜ lcavol + lweight + age + lbph + svi + lcp
+ gleason + pgg45 - 1)
```

# Regularized Linear Regression in R

```
> fit.ols

Call:
lm(formula = lpsa ~ lcavol + lweight + age + lbph + svi + lcp +
    gleason + pgg45 - 1)

Coefficients:
  lcavol   lweight       age      lbph       svi       lcp
 0.58905   0.22825  -0.12455   0.18252   0.26395  -0.24848
 gleason     pgg45
-0.01566   0.22819

> y.pred.ols <- predict(fit.ols,data.frame(xp.test[,1:8]))
```

# Regularized Linear Regression in R

To fit a regularized linear model, we use the package `glmnet`

- `glmnet` regresses on matrices, not data frames
- has parameter `alpha`, where `alpha` =0 means Ridge, `alpha` =1 means Lasso

$$\min_{\beta} \sum_{i=1}^{n} \left(y_i - x_i^\top \boldsymbol{\beta}\right)^2 + \lambda \sum_{j=1}^{p} \left[(1-\alpha)\beta_j^2 + \alpha|\beta_j|\right]$$

- use the function `cv.glmnet( )` to find the right parameter for $\lambda$
- use the function `predict( )` to get a prediction

```
> library(glmnet)
> cv.fit.ridge <- cv.glmnet(xp.train[,1:8],lpsa,alpha=0)
> y.pred.ridge <- predict(cv.fit.ridge,xp.test[,1:8])
```

# Regularized Linear Regression in R

```
 > cv.fit.lasso <- cv.glmnet(xp.train[,1:8],lpsa,alpha=1)
 > y.pred.lasso <- predict(cv.fit.lasso,xp.test[,1:8])
 > # Compute MSE for test set
 > c(mean((y.pred.ols-xp.test[,9])^2),
 mean((y.pred.ridge-xp.test[,9])^2),
 mean((y.pred.lasso-xp.test[,9])^2))
[1] 0.3891581 0.3982256 0.3726441
```

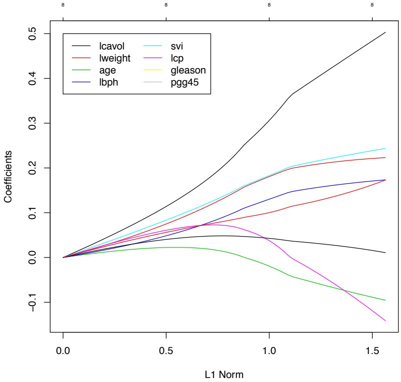
# Regularized Linear Regression in R

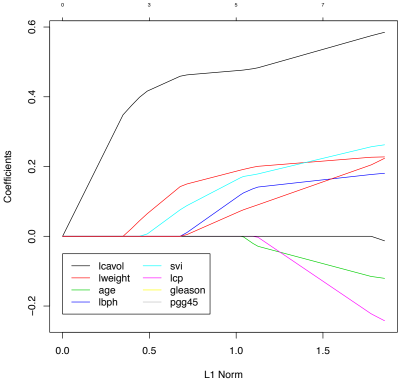
So how do the coefficients change with $\lambda$?

- can use glmnet to show these

```
 > fit.ridge <- glmnet(xp.train[,1:8],lpsa,alpha=0)
> plot(fit.ridge)
> legend(0,0.5,c("lcavol","lweight","age","lbph","svi","lcp",
"gleason", "pgg45"),col=1:8,lty=1,ncol=2)
> fit.lasso <- glmnet(xp.train[,1:8],lpsa,alpha=1)
> plot(fit.lasso)
> legend(0,-0.05,c("lcavol","lweight","age","lbph","svi","lcp",
"gleason", "pgg45"),col=1:8,lty=1,ncol=2)
```

# Regularized Linear Regression in R





# Regularized Linear Regression in R

What was that plot?

- $L_1$ norm on x-axis ($\sum |\beta_j|$)
- $\beta_{1:p}$ on y-axis

So, let's plot one coefficient...

```
> L1.norm <- function(x) sum(abs(x))
> plot(apply(fit.ridge$beta,2,L1.norm),fit.ridge$beta[1,],type="l")
> lines(apply(fit.ridge$beta,2,L1.norm),fit.ridge$beta[2,],col=2)
> lines(apply(fit.ridge$beta,2,L1.norm),fit.ridge$beta[3,],col=3)
> beta.min <- min(fit.ridge$beta)
> beta.max <- max(fit.ridge$beta)
> plot(apply(fit.ridge$beta,2,L1.norm),fit.ridge$beta[1,],type="l",
  ylim=c(beta.min,beta.max))
```

# Regularized Linear Regression in R

Figuring out the coefficients for the optimal models:

```
> cv.fit.ridge$lambda.min
[1] 0.1006497
> which(cv.fit.ridge$lambda == cv.fit.ridge$lambda.min)
[1] 97
> fit.ridge$beta[,97]
     lcavol     lweight         age        lbph
 0.48281519  0.22108049 -0.08846349  0.17067165
        svi         lcp     gleason       pgg45
 0.23836712 -0.11783893  0.01591430  0.16183770
> cv.fit.lasso$lambda.min
[1] 0.009607497
> which(cv.fit.lasso$lambda == cv.fit.lasso$lambda.min)
[1] 48
> fit.lasso$beta[,48]
    lcavol    lweight        age       lbph
 0.5605983  0.2225770 -0.1016259  0.1715339
       svi        lcp    gleason      pgg45
 0.2444484 -0.1878424  0.0000000  0.1867427
```