EECS E6690: Statistical Learning for Biological and Information Systems
Lecture 3: Model Selection and Regularization

Prof. Predrag R. Jelenković
Time: Tuesday 4:10-6:40pm
303 Seeley W. Mudd Building

Dept. of Electrical Engineering
Columbia University , NY 10027, USA
Office: 812 Schapiro Research Bldg.
Phone: (212) 854-8174
Email: predrag@ee.columbia.edu
URL: http://www.ee.columbia.edu/∼predrag

Big Picture: Estimation, Testing and Model Selection

Estimation:

- Select a class of function for f: Hypothesis class $\mathcal{H}$
  say, $\mathcal{H}$ are linear functions, i.e., linear regression
- Optimization: find $\hat{f} \in \mathcal{H}$ which minimizes the error/loss function

Testing: How good is $\hat{f}$ on unseen data? Two approaches:

- *Analytical*: Make some analytical assumptions, e.g. Gaussian, and compute distributions for the parameters of interest.
  Develop statistical tests to characterize $\hat{f}$: t-test, F-test, etc.
- *Numerical* (coming soon): Split data into training and testing.
  Use training data to find $\hat{f}$ and testing data to evaluate it.

Model selection and regularization: Find the smallest/simplest model?

- *Analytical*: Use F/t-tests to select the smallest model
- *Numerical*:
  - Model selection: Fit and test models with less predictors
  - Regularization: Modify the loss function such that it penalizes more complex models. This also helps with overfitting.

Last lecture: Multidimensional linear regression

- p *predictors (features, independent variables)*
- n observations: $(y_i, x_{i,1}, x_{i,2}, \ldots, x_{i,p})$
- Given estimates $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_p$, the prediction is

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x_1 + \hat{\beta}_2 x_2 + \cdots + \hat{\beta}_p x_p,$$

  or in matrix form

$$\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \begin{bmatrix} 1 & x_{1,1} & \cdots & x_{1,p} \\ 1 & x_{2,1} & \cdots & x_{2,p} \\ \vdots & & & \vdots \\ 1 & x_{n,1} & \cdots & x_{n,p} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \\ \vdots \\ \hat{\beta}_p \end{bmatrix}$$

- Minimize (over $\beta_1, \ldots, \beta_p$) the residual sum of squares (l_2 norm)

$$\mathrm{RSS}(\boldsymbol{\beta}) = \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 = (\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta})$$

Last lecture: l_2 solution

- Differentiating RSS($\boldsymbol{\beta}$), we get

$$-2\boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}) = \boldsymbol{0}$$

- If $\boldsymbol{X}^\top \boldsymbol{X}$ has a full rank

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$$
$$\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top \boldsymbol{y}$$

- *Hat matrix*: $\boldsymbol{P} := \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$; it puts a hat on $\boldsymbol{y}$
  $\boldsymbol{P}$ is the l_2-projection matrix of $\boldsymbol{y}$ onto $C(\boldsymbol{X})$
- $\hat{\boldsymbol{y}}$ and $(\boldsymbol{y} - \hat{\boldsymbol{y}})$ are ortogonal; $\hat{\boldsymbol{y}}$ is in $C(\boldsymbol{X})$
- $\sum_{i=1}^n (y_i - \hat{y}_i) = (\boldsymbol{y} - \hat{\boldsymbol{y}})\boldsymbol{1} = 0$
  (since $\boldsymbol{1} \in C(\boldsymbol{X})$ and $(\boldsymbol{y} - \hat{\boldsymbol{y}})$ orthogonal to $C(\boldsymbol{X})$)

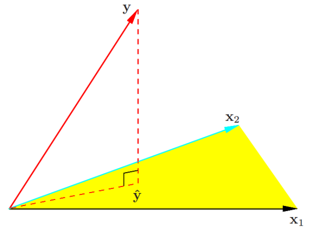
Geometry of 1D linear regression

Consider a data set $(x_1, y_1), \ldots, (x_n, y_n), n \geq 2$, and let

$$\mathbf{1} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \quad \boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \boldsymbol{X} = [\mathbf{1}, \boldsymbol{x}], \quad \boldsymbol{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \hat{\boldsymbol{y}} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}$$

Then, $\hat{\boldsymbol{y}}$ is projection of vector $\boldsymbol{y}$ onto a plane spanned by vectors $(\mathbf{1}, \boldsymbol{x})$, and can be computed algebraically as

$$\hat{\boldsymbol{y}} = \boldsymbol{P}\boldsymbol{y}, \quad \text{where } \boldsymbol{P} := \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top.$$



Geometry of 1D linear regression

Let us now compute this projection geometrically. First let us convert vectors $(\mathbf{1}, \boldsymbol{x})$ into orthonormal basis $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ using Gram-Schmidt method. Assume that $\mathbf{1}$ and $\boldsymbol{x}$ are linearally independent, i.e., $\boldsymbol{x} \neq c\mathbf{1}, c \neq 0$. First, we normalize $\boldsymbol{u}_1 = \mathbf{1}/\sqrt{n}$, and then

$$\boldsymbol{u}_2 = \frac{x - (\boldsymbol{x} \cdot \boldsymbol{u}_1)\boldsymbol{u}_1}{\|x - (\boldsymbol{x} \cdot \boldsymbol{u}_1)\boldsymbol{u}_1\|} = \frac{\boldsymbol{x} - \bar{x}\mathbf{1}}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}}, \quad \text{where } \bar{x} = \frac{1}{n}\sum x_i.$$

Now, we project $\boldsymbol{y}$ onto $(\boldsymbol{u}_1, \boldsymbol{u}_2)$ using

$$\begin{aligned} \hat{\boldsymbol{y}} &= (\boldsymbol{y} \cdot \boldsymbol{u}_1)\boldsymbol{u}_1 + (\boldsymbol{y} \cdot \boldsymbol{u}_2)\boldsymbol{u}_2 \qquad (1) \\ &= \bar{y}\mathbf{1} + \frac{\boldsymbol{x} \cdot \boldsymbol{y} - \bar{x}\mathbf{1} \cdot \boldsymbol{y}}{\sum(x_i - \bar{x})^2}(\boldsymbol{x} - \bar{x}\mathbf{1}), \quad \text{where } \bar{y} = \frac{1}{n}\sum y_i \\ &= \hat{\beta}_0\mathbf{1} + \hat{\beta}_1\boldsymbol{x}, \quad \text{where } \hat{\beta}_1 = \frac{\sum(y_i - \bar{y})(x_i - \bar{x})}{\sum(x_i - \bar{x})^2}, \hat{\beta}_0 = \bar{y} - \hat{\beta}_1\bar{x}. \end{aligned}$$

Moreover, using (1), argue that the projection matrix $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^\top$, where $\boldsymbol{Q}$ is a matrix with columns $(\boldsymbol{u}_1, \boldsymbol{u}_2)$, i.e., $\boldsymbol{Q} = [\boldsymbol{u}_1\boldsymbol{u}_2]$. Show that $\boldsymbol{P} = \boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{X}(\boldsymbol{X}^\top\boldsymbol{X})^{-1}\boldsymbol{X}^\top$.

Dual solution: dot products and kernels

Note that $\hat{\boldsymbol{\beta}}$ can be represented as a linear combination of data

$$\hat{\boldsymbol{\beta}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y} = \boldsymbol{X}^\top \left(\boldsymbol{X} (\boldsymbol{X}^\top \boldsymbol{X})^{-2} \boldsymbol{X}^\top \boldsymbol{y} \right) =: \boldsymbol{X}^\top \boldsymbol{\alpha} = \sum_{i=1}^{n} \alpha_i \boldsymbol{x}_i,$$

where $\boldsymbol{x}_i = (1, x_{i,1}, \ldots x_{i,p})$ is the ith data point, which implies

$$\hat{\boldsymbol{y}} = \boldsymbol{X} \hat{\boldsymbol{\beta}} = \boldsymbol{X} \boldsymbol{X}^\top \boldsymbol{\alpha} =: \boldsymbol{K} \boldsymbol{\alpha},$$

where $\boldsymbol{K}$ is a matrix of dot products, also known as Kernel or Gram matrix, which is symmetric and positive definite

$$K_{kj} = \langle \boldsymbol{x}_k, \boldsymbol{x}_j \rangle := \sum_{l=0}^{p} x_{k,l} x_{j,l}.$$

Hence, by minimizing the dual problem $\|\boldsymbol{y} - \hat{\boldsymbol{y}}\|_2^2 = \|\boldsymbol{y} - \boldsymbol{K}\boldsymbol{\alpha}\|_2^2$, one finds (assuming $\boldsymbol{K}$ being non-singular)

$$\boldsymbol{\alpha} = \boldsymbol{K}^{-1} \boldsymbol{y}$$

which has computational complexity $O(n^3)$. Direct computation of K requires $O(n^2 p)$ operations, resulting in total complexity $O(n^2(p+n)) \ll O(p^3)$ when $n \ll p$.

We will be back to dual (Kernel) solution throughout the course.

Last lecture: Goodness of fit

- Total sum of squares: $\mathsf{TSS} = (\boldsymbol{y} - \bar{y}\mathbf{1})^\top (\boldsymbol{y} - \bar{y}\mathbf{1})$
- Explained sum of squares: $\mathsf{ESS} = (\hat{\boldsymbol{y}} - \bar{y}\mathbf{1})^\top (\hat{\boldsymbol{y}} - \bar{y}\mathbf{1})$
- Then

$$\begin{aligned}
\mathsf{TSS} &= (\boldsymbol{y} - \hat{\boldsymbol{y}} + \hat{\boldsymbol{y}} - \bar{y}\mathbf{1})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}} + \hat{\boldsymbol{y}} - \bar{y}\mathbf{1}) \\
&= \mathsf{RSS} + \mathsf{ESS} + 2(\boldsymbol{y} - \hat{\boldsymbol{y}})^\top (\hat{\boldsymbol{y}} - \bar{y}\mathbf{1}) \\
&= \mathsf{RSS} + \mathsf{ESS}
\end{aligned}$$

- A measure of quality of the model

$$R^2 = \frac{\mathsf{ESS}}{\mathsf{TSS}} = 1 - \frac{\mathsf{RSS}}{\mathsf{TSS}}$$

- R^2 - *Coefficient of determination*: better fit as $R^2 \uparrow 1$
  i.e., more data explained by the model
  $R^2 = 1$ perfect linear fit

Last lecture: Computing distributions

- Normal/Gaussian assumption: i.i.d. $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- **Normal/Gaussian + Linear**: Can compute anything
  Check EC in HW1 for the derivation of χ^2, t, F distributions.
- $\hat{\boldsymbol{\beta}} \sim \mathcal{N}(\boldsymbol{\beta}, \sigma^2(\boldsymbol{X}^\top \boldsymbol{X})^{-1})$
- RSS $= (\boldsymbol{y} - \hat{\boldsymbol{y}})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}) = \boldsymbol{\epsilon}^\top (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{\epsilon}$, with zero-mean, normal residuals

$$\mathbb{E}[\boldsymbol{y} - \boldsymbol{X}\hat{\boldsymbol{\beta}}] = \mathbb{E}[\boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} - \boldsymbol{X}\hat{\boldsymbol{\beta}}] = \boldsymbol{0}$$

- Then it can be shown

$$\frac{\text{RSS}}{\sigma^2} \sim \chi^2_{n-p-1}$$

- Estimator: when n is large, $\chi^2_{n-p-1} \approx n - p - 1$, and

$$\hat{\sigma} = \sqrt{\frac{\text{RSS}}{n-p-1}}$$

Last lecture: F-test

- Could use $\hat{\beta}$ and t-test, but F (Fisher)-test is easier
- F-test idea: **use RSS** to test instead of $\hat{\boldsymbol{\beta}}$
- $\mathcal{H}_0 : \beta_1 = \beta_2 = \cdots = \beta_p = 0$
- $\mathcal{H}_1$: exists j such that $\beta_j \neq 0$
- Under $\mathcal{H}_0$, we have a null model: $Y = \beta_0 + \epsilon$
- Let RSS_0 be the residual sum of squares under $\mathcal{H}_0$
- Under $\mathcal{H}_0$:

$$\frac{\text{RSS}_0 - \text{RSS}}{\sigma^2} = \frac{\text{TSS} - \text{RSS}}{\sigma^2} \sim \chi^2_p$$

and

$$\frac{\frac{\text{TSS}-\text{RSS}}{p}}{\frac{\text{RSS}}{n-p-1}} \sim F_{p,n-p-1}$$

F-distribution computed explicitly in Extra Credit, HW1.

Last lecture: F-test genral

- (m) denotes a sub-model obtained by a linear constraint on $\boldsymbol{\beta}$
- Examples
  - $\beta_1 = \beta_2 = \ldots = \beta_p$: $Y = \beta_0 + \beta_1(X_1 + X_2 \cdots + X_p) + \epsilon$
  - $\beta_1 = \beta_2$: $Y = \beta_0 + \beta_1(X_1 + X_2) + \beta_3 X_3 + \ldots + \beta_p X_p + \epsilon$
- Testing: $\mathcal{H}_0$ (reduced model) vs. $\mathcal{H}_1$ (complete model)
- $q < p$ is the number of explanatory variables in the reduced model
- Under $\mathcal{H}_0$:

$$\frac{\mathrm{RSS}_{(m)} - \mathrm{RSS}}{\sigma^2} \sim \chi^2_{p-q}$$

and

$$\frac{\frac{\mathrm{RSS}_{(m)} - \mathrm{RSS}}{p-q}}{\frac{\mathrm{RSS}}{n-p-1}} \sim F_{p-q,n-p-1}$$

Small digression: RSS, χ^2 and Cochran's Theorem

- Several times in the class we said that the distribution of RSS/σ^2 for linear regression has χ^2 distribution for Gaussian noise ϵ $(Y = f(X) + \epsilon)$

$$\frac{RSS}{\sigma^2} \sim \chi^2_{n-p-1},$$

  where $n - p - 1$ represents the degrees of freedom and p is the number of predictors

- How can we show/prove this? (*Not needed for the grade.*)

- **Cochran's Theorem (1934)** If $y_i, 1 \leq 1 \leq n$ are i.i.d. $\mathcal{N}(0, \sigma^2)$ gaussian and $\boldsymbol{A}_j, j = 1, 2$ are idempotent ($\boldsymbol{A}_j^2 = \boldsymbol{A}_j$) and symmetric ($\boldsymbol{A}_j^\top = \boldsymbol{A}_j$) matrices such that rank$(\boldsymbol{A}_j) = r_j, r_1 + r_2 = n$ and $\boldsymbol{A}_1 + \boldsymbol{A}_2 = \boldsymbol{I}_n$, where $\boldsymbol{I}_n$ is $n \times n$ identity matrix, then the following variables are independent and have χ^2 distribution

$$\boldsymbol{y}^\top \boldsymbol{A}_j \boldsymbol{y} \sim \sigma^2 \chi^2_{r_j}$$

  Proof: For example, it can be found here.

Application of Cochran's Theorem: Distribution of RSS

- Recall that $\hat{\boldsymbol{y}} = \boldsymbol{P}\boldsymbol{y}$, where $\boldsymbol{P}$ is the hat matrix

$$\boldsymbol{P} = \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top$$

- Now, it is easy to check that $\boldsymbol{P}$ is symmetric and idempotent, i.e. $\boldsymbol{P} = \boldsymbol{P}^\top$ and $\boldsymbol{P}^2 = \boldsymbol{P}$
- Next, the same is true for $\boldsymbol{I} - \boldsymbol{P}$ since it is a difference of 2 symmetric matrices and

$$(\boldsymbol{I} - \boldsymbol{P})^2 = \boldsymbol{I} - 2\boldsymbol{P} + \boldsymbol{P}^2 = \boldsymbol{I} - 2\boldsymbol{P} + \boldsymbol{P} = \boldsymbol{I} - \boldsymbol{P}$$

- **Rank:** From linear algebra, it is known that rank of symmetric and indempotent matrices is equal to their trace ($\text{tr}(AB) = \text{tr}(BA)$)

$$\begin{aligned} \text{rank}(\boldsymbol{P}) &= \text{tr}(\boldsymbol{P}) = \text{trace}(\boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}\boldsymbol{X}^\top) \\ &= \text{trace}(\boldsymbol{X}^\top \boldsymbol{X}(\boldsymbol{X}^\top \boldsymbol{X})^{-1}) = \text{trace}(\boldsymbol{I}_{p+1}) = p + 1 \end{aligned}$$

Application of Cochran's Theorem: Distribution of RSS

- Then,

$$\text{rank}(\boldsymbol{I} - \boldsymbol{P}) = \text{tr}(\boldsymbol{I} - \boldsymbol{P}) = \text{tr}(\boldsymbol{I}) - \text{tr}(\boldsymbol{P}) = n - p - 1$$

- Finally, by applying Cochran's Theorem,

$$\begin{aligned} RSS &= (\boldsymbol{y} - \hat{\boldsymbol{y}})^\top (\boldsymbol{y} - \hat{\boldsymbol{y}}) \\ &= (\boldsymbol{y} - \boldsymbol{P}\boldsymbol{y})^\top (\boldsymbol{y} - \boldsymbol{P}\boldsymbol{y}) \\ &= \boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{P})^\top (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{y} \\ &= \boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{P})^2 \boldsymbol{y} \\ &= \boldsymbol{y}^\top (\boldsymbol{I} - \boldsymbol{P})\boldsymbol{y} \sim \sigma^2 \chi^2_{n-p-1} \end{aligned}$$

- Similarly, we can apply Cochran's Theorem to compute the distribution of TSS and ESS

Linear Model Selection: Analytical Approach

- Recall advertising example from last lecture
- Now, that we know the meaning of: standard error, t-value, F-value, p-value, R^2, we can completely understand the output of the linear model fit function, lm$(\cdot)$.

```
> lm2<-lm(adv$Sales~adv$TV+adv$Radio+adv$Newspaper)
> summary(lm2)

Call:
lm(formula = adv$Sales ~ adv$TV + adv$Radio + adv$Newspaper)

Residuals:
    Min      1Q  Median      3Q     Max 
-8.8277 -0.8908  0.2418  1.1893  2.8292 

Coefficients:
               Estimate Std. Error t value   Pr(>|t|)    
(Intercept)    2.938889   0.311908   9.422   <2e-16 ***
adv$TV         0.045765   0.001395  32.809   <2e-16 ***
adv$Radio      0.188530   0.008611  21.893   <2e-16 ***
adv$Newspaper -0.001037   0.005871  -0.177     0.86    
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.686 on 196 degrees of freedom
Multiple R-squared:  0.8972, Adjusted R-squared:  0.8956 
F-statistic: 570.3 on 3 and 196 DF,  p-value: < 2.2e-16
```

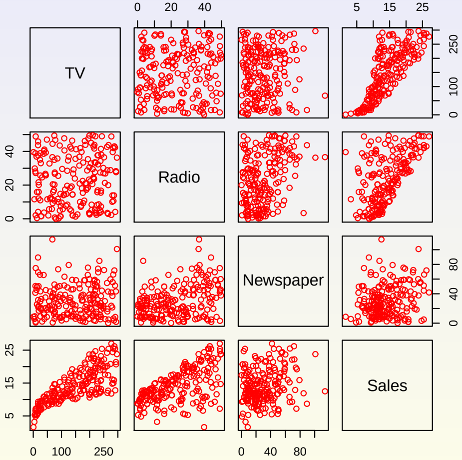
Linear Model Selection: Analytical Approach

- Hence, the model with one less predictor (without the newspaper) might be just as good

```
> summary(lm(adv$Sales~adv$TV+adv$Radio))

Call:
lm(formula = adv$Sales ~ adv$TV + adv$Radio)

Residuals:
    Min      1Q  Median      3Q     Max
-8.7977 -0.8752  0.2422  1.1708  2.8328

Coefficients:
            Estimate Std. Error t value   Pr(>|t|)
(Intercept)  2.92110    0.29449   9.919    <2e-16 ***
adv$TV       0.04575    0.00139  32.909    <2e-16 ***
adv$Radio    0.18799    0.00804  23.382    <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Residual standard error: 1.681 on 197 degrees of freedom
Multiple R-squared:  0.8972, Adjusted R-squared:  0.8962
 F-statistic: 859.6 on 2 and 197 DF,  p-value: < 2.2e-16
```

Linear Model Selection: Visual Examination

- We made a lot of assumptions in this model
- Note: examine the data visually to see if the model makes sense



- **How do we find the best model in general?**

Linear Model Selection and Regularization

Additional motivation:

- Prediction
  - High-dimensional data, $p \gtrapprox n$ – overfitting to the training data
  - Cannot use the plain vannila least squares
- Model interpretability
  - Hard to interpret model with many predictors
  - Focus on most important variables
- **Idea:** Modify least squares
- Agenda:
  - Subset selection
  - Shrinkage methods
  - Dimension reduction techniques (next class)

Model Selection: Bias-Variance Trade-off

Test Error, aka Generalization Error can be decomposed as:

- Let x_0 be a test (unseen) point and $y_0 = f(x_0) + \epsilon_0$

$$\begin{aligned}
\mathsf{Err}(x_0) &= \mathbb{E}\left(y_0 - \hat{f}(x_0)\right)^2 \\
&= \mathbb{E}\left(f(x_0) + \epsilon_0 - \hat{f}(x_0)\right)^2 \\
&= \sigma^2 + \mathbb{E}\left(f(x_0) - \mathbb{E}\hat{f}(x_0) - \hat{f}(x_0) + \mathbb{E}\hat{f}(x_0)\right)^2 \\
&= \sigma^2 + \left(f(x_0) - \mathbb{E}\hat{f}(x_0)\right)^2 + \mathsf{Var}(\hat{f}(x_0)) \\
&= \sigma^2 + \left(\mathsf{Bias}(\hat{f}(x_0))\right)^2 + \mathsf{Var}(\hat{f}(x_0))
\end{aligned}$$

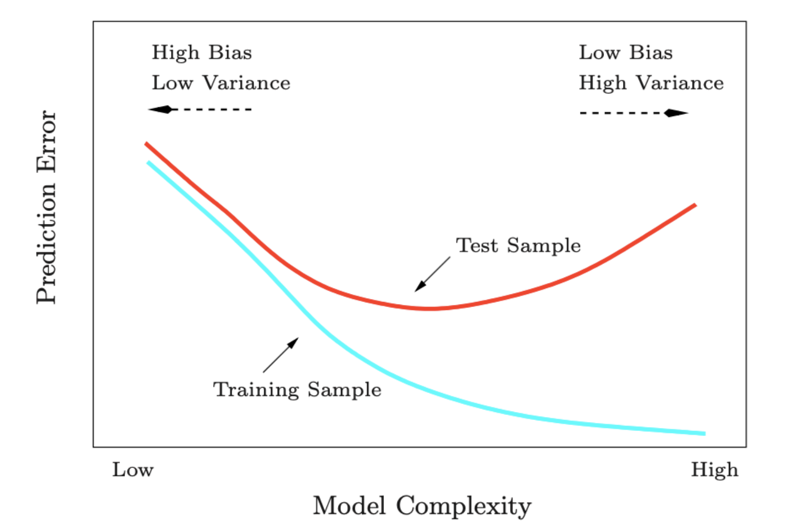
- Linear regression: recall $\hat{f}(x_0) = x_0^\top \hat{\boldsymbol{\beta}} = x_0^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$

$$\mathsf{Var}(\hat{f}(x_0)) = \sigma^2 \mathbb{E}\left(x_0^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1} x_0\right) \approx \frac{p}{n}\sigma^2$$

  assuming $\boldsymbol{X}$ random, zero mean and n large: $\boldsymbol{X}^\top \boldsymbol{X} \approx n\mathsf{Cov}(X)$.

  Increasing p increases the variance and in general reduces the bias

Testing and Training Error versus Model Complexity



Model Selection: Deciding on the important variables

We have seen analytical approaches via F/t-statistics

Numerical approaches:

- Need a criteria that balance training error and model size
- Several approaches
  - Best subsets selection
    - Consider all 2^p models
    - Infeasible when p is large
  - Forward selection
    - Start with a null model – no predictors
    - Add predictors one-by-one
    - Stopping criterion
  - Backward selection
    - Start with a full model – p predictors
    - Eliminate predictors one-by-one
    - Stopping criterion

Best Subset Selection

- Algorithm
  - Let $\mathcal{M}_0$ denote the null model (no predictors, sample mean prediction)
  - For $k = 1, 2, \ldots, p$:
    - Fit all $\binom{p}{k}$ models that contain exactly k predictors
    - Let $\mathcal{M}_k$ be the best of these $\binom{p}{k}$ models in terms of the smallest RSS (equivalently the largest R^2)
  - Select the best model from among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ using some criterion
- Number of models is 2^p
- Example: if $p = 20$, number of models to check $2^{20} = 1,048,576$

Prostate cancer data set

Data frame with 97 observations on the following 10 variables:

- **lcavol** - log cancer volume
- **lweight** - log prostate weight
- **age** in years
- **lbph** - log of the amount of benign prostatic hyperplasia
- **svi** - seminal vesicle invasion
- **lcp** - log of capsular penetration
- **gleason** - a numeric vector
- **pgg45** - percent of Gleason score 4 or 5
- **lpsa** - response: log of prostate specific antigen (PSA)
- **train** logical True/False vector

Loading libraries and prostate data

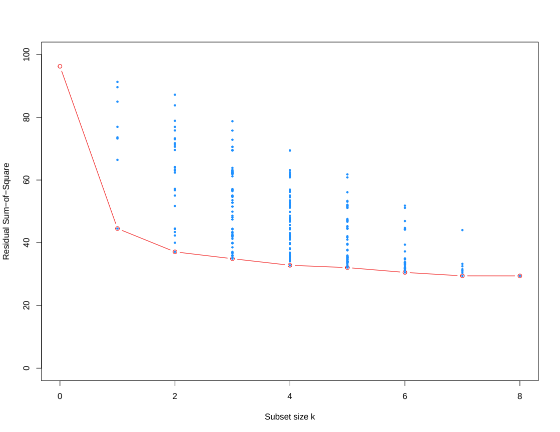
```
# Loading libraries/packages and data:

library(ISLR)
library(ElemStatLearn)
data(prostate)

# Checking "prostate" data
# Data manuals:
# https://cran.r-project.org/web/packages/ElemStatLearn/ElemStatLearn.pdf
# https://cran.r-project.org/web/packages/ISLR/ISLR.pdf

fix(prostate)
str(prostate)
cor(prostate[,1:8])
pairs(prostate[,1:9], col="violet")

# Separating the training and test data:
train <- subset( prostate, train==TRUE )[,1:9]
test <- subset( prostate, train=FALSE )[,1:9]
```

Best Subset Selection

```
# Loading "leaps" library/package for the best subset selection
library(leaps)

# Computing all combinations using "regsubsets"
prostate.leaps <- regsubsets( lpsa ~ . , data=train, nbest=70, really.big=TRUE )
prostate.leaps.sum <- summary( prostate.leaps )
prostate.models <- prostate.leaps.sum$which
prostate.models.size <- as.numeric(attr(prostate.models, "dimnames")[[1]])
hist( prostate.models.size )

#Extracting all and the best RSS
prostate.models.rss <- prostate.leaps.sum$rss
prostate.models.best.rss <-tapply( prostate.models.rss, prostate.models.size, min )
prostate.models.best.rss

# Adding the result with no X-s,  only intercept (beta0) model
prostate.dummy <- lm( lpsa ~ 1, data=train )
prostate.models.best.rss <- c(sum(resid(prostate.dummy)^2),prostate.models.best.rss)
```

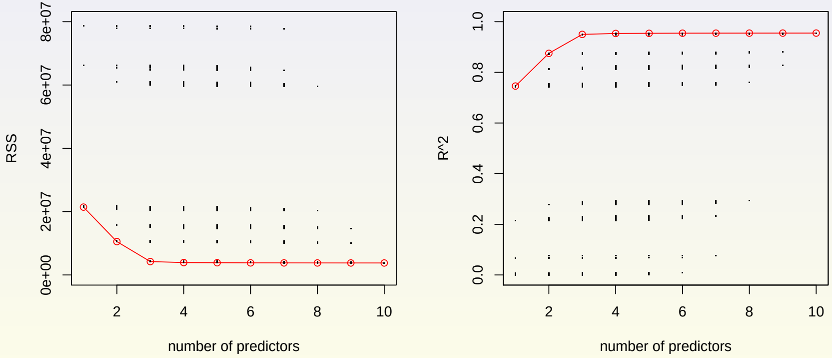
Best Subset Selection

```
# Making the plot
plot( 0:8, prostate.models.best.rss, ylim=c(0, 100),
        type="b", xlab="Subset size k", ylab="Residual Sum-of-Square", col="red2" )
points( prostate.models.size, prostate.models.rss, pch=20, col="dodgerblue",cex=0.7 )
```



Another Example

- Credit card data set



More Examples

```
> library(ISLR)
> names(Hitters)
 [1] "AtBat"     "Hits"      "HmRun"     "Runs"      "RBI"       "Walks"     "Years"     "CAtBat"
 [9] "CHits"     "CHmRun"    "CRuns"     "CRBI"      "CWalks"    "League"    "Division"  "PutOuts"
[17] "Assists"   "Errors"    "Salary"    "NewLeague"
> dim(Hitters)
[1] 322  20
> sum(is.na(Hitters$Salary))
[1] 59
> Hitters<-na.omit(Hitters)
> library(leaps)
> regfit.full<-regsubsets(Salary~.,data=Hitters)
> summary(regfit.full)

1 subsets of each size up to 8
Selection Algorithm: exhaustive
         AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI CWalks LeagueN DivisionW
1  ( 1 ) " "   " "  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
2  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
3  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
4  ( 1 ) " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
5  ( 1 ) "*"   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
6  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
7  ( 1 ) " "   "*"  " "   " "  " " "*"   " "   "*"    "*"   "*"    " "   " "  " "    " "     "*"
8  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   " "    " "   "*"    "*"   " "  "*"    " "     "*"
         PutOuts Assists Errors NewLeagueN
1  ( 1 ) " "     " "     " "    " "
2  ( 1 ) " "     " "     " "    " "
3  ( 1 ) "*"     " "     " "    " "
4  ( 1 ) "*"     " "     " "    " "
5  ( 1 ) "*"     " "     " "    " "
6  ( 1 ) "*"     " "     " "    " "
7  ( 1 ) "*"     " "     " "    " "
8  ( 1 ) "*"     " "     " "    " "
```

Forward Selection

- Reduce computational complexity by forfeiting optimality
- Algorithm
  - Let $\mathcal{M}_0$ denote the null model (no predictors, sample mean prediction)
  - for $k = 0, 1, \ldots, p-1$
    - Fit all $p-k$ models that augment the predictors in $\mathcal{M}_k$ with one additional predictor
    - Let $\mathcal{M}_{k+1}$ be the best of these $p-k$ models in terms of the smallest RSS (equivalently the largest R^2)
  - Select the best model from among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ using some criterion
- Greedy : not optimal, but tractable
- Number of models is $1 + p(p+1)/2 = O(p^2) \ll 2^p$
- If $p > n$, we can construct $\mathcal{M}_0, \ldots, \mathcal{M}_n$ models only

Example: Forward Selection is not optimal

```
> regfit.fwd<-regsubsets(Salary~.,data=Hitters,nvmax=19,method = "forward")
> summary(regfit.fwd)
.
.
Selection Algorithm: forward
          AtBat Hits HmRun Runs RBI Walks Years CAtBat CHits CHmRun CRuns CRBI CWalks LeagueN DivisionW
1  ( 1 )  " "   " "  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
2  ( 1 )  " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
3  ( 1 )  " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     " "
4  ( 1 )  " "   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
5  ( 1 )  "*"   "*"  " "   " "  " " " "   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
6  ( 1 )  "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    " "   "*"  " "    " "     "*"
7  ( 1 )  "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    " "   "*"  "*"    " "     "*"
8  ( 1 )  "*"   "*"  " "   " "  " " "*"   " "   " "    " "   " "    "*"   "*"  "*"    " "     "*"
9  ( 1 )  "*"   "*"  " "   " "  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    " "     "*"
10  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    " "     "*"
11  ( 1 ) "*"   "*"  " "   " "  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    "*"     "*"
12  ( 1 ) "*"   "*"  " "   "*"  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    "*"     "*"
13  ( 1 ) "*"   "*"  " "   "*"  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    "*"     "*"
14  ( 1 ) "*"   "*"  "*"   "*"  " " "*"   " "   "*"    " "   " "    "*"   "*"  "*"    "*"     "*"
15  ( 1 ) "*"   "*"  "*"   "*"  " " "*"   " "   "*"    "*"   " "    "*"   "*"  "*"    "*"     "*"
16  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   " "   "*"    "*"   " "    "*"   "*"  "*"    "*"     "*"
17  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   " "   "*"    "*"   " "    "*"   "*"  "*"    "*"     "*"
18  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   " "    "*"   "*"  "*"    "*"     "*"
19  ( 1 ) "*"   "*"  "*"   "*"  "*" "*"   "*"   "*"    "*"   "*"    "*"   "*"  "*"    "*"     "*"
          PutOuts Assists Errors NewLeagueN
1  ( 1 )  " "     " "     " "    " "
2  ( 1 )  " "     " "     " "    " "
3  ( 1 )  "*"     " "     " "    " "
4  ( 1 )  "*"     " "     " "    " "
5  ( 1 )  "*"     " "     " "    " "
6  ( 1 )  "*"     " "     " "    " "
7  ( 1 )  "*"     " "     " "    " "
8  ( 1 )  "*"     " "     " "    " "
9  ( 1 )  "*"     " "     " "    " "
10  ( 1 ) "*"     "*"     " "    " "
11  ( 1 ) "*"     "*"     " "    " "
12  ( 1 ) "*"     "*"     " "    " "
13  ( 1 ) "*"     "*"     "*"    " "
14  ( 1 ) "*"     "*"     "*"    " "
15  ( 1 ) "*"     "*"     "*"    " "
16  ( 1 ) "*"     "*"     "*"    " "
17  ( 1 ) "*"     "*"     "*"    "*"
18  ( 1 ) "*"     "*"     "*"    "*"
19  ( 1 ) "*"     "*"     "*"    "*"
```

Feature selection measures

- $p + 1$ models: $\mathcal{M}_0, \ldots, \mathcal{M}_p$. Which one is the best?
- RSS and R^2 estimate the training error, not the testing error

- Two approaches:
  - Indirect (adjust the training error)
  - Direct: validation, cross-validation (next week)

Indirect measures: C_p, AIC and BIC

- Model with $d \leq p$ predictors
- Mallow's C_p:
$$C_p = \frac{1}{n}(\text{RSS} + 2d\hat{\sigma}^2),$$
where $\hat{\sigma}$ is an estimator for the variance of noise (estimated on a model containing all predictors)
- Akaike information criteria (AIC):
$$\text{AIC} = \frac{1}{n\hat{\sigma}^2}(\text{RSS} + 2d\hat{\sigma}^2)$$
- Bayesian AIC (BIC):
$$\text{BIC} = \frac{1}{n}(\text{RSS} + d\hat{\sigma}^2 \log n)$$
- Heuristic: select a model with the lowest C_p, AIC, BIC

Indirect measures: Adjusted R^2

- Recall

$$R^2 = 1 - \frac{\mathsf{RSS}}{\mathsf{TSS}}$$

- Adjusted R^2

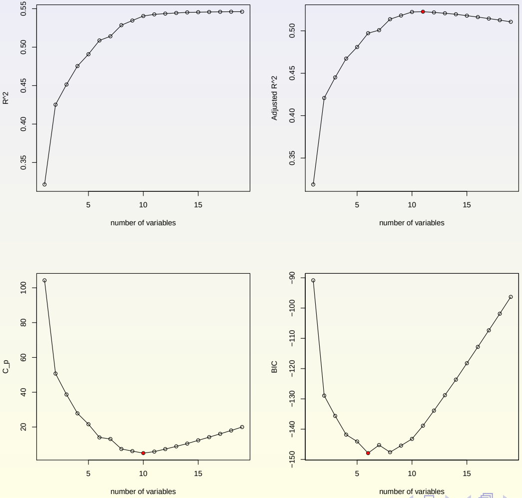
$$\mathsf{Adj}R^2 = 1 - \frac{\mathsf{RSS}/(n-d-1)}{\mathsf{TSS}/(n-1)}$$

- Heuristic: $\max \mathsf{Adj}R^2$
- Equivalent to

$$\min \frac{\mathsf{RSS}}{n-d-1}$$

Example

```
> regfit.full<-regsubsets(Salary~.,data=Hitters,nvmax = 19)
> reg.summary<-summary(regfit.full)
> names(reg.summary)
[1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
> reg.summary$rsq
 [1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146 0.5141227 0.5285569 0.5346124 0.5404950
[11] 0.5426153 0.5436302 0.5444570 0.5452164 0.5454692 0.5457656 0.5459518 0.5460945 0.5461159
```



Shrinkage methods

- An alternative to subset selection
- Idea: Regularize/constrain coefficients

- Two widely-used methods:
  - Ridge regression
  - LASSO (Least Absolute Shrinkage and Selection Operator)

Ridge Regression

- OLS: minimize RSS

$$\text{RSS} = \sum_{i=1}^{n}\left(y_i - \beta_0 - \sum_{j=1}^{p}\beta_j x_{i,j}\right)^2$$

- Ridge Regression: minimize (RSS + shrinkage penalty)

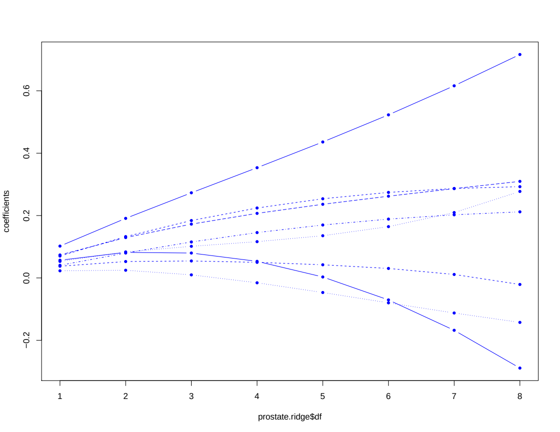
$$\text{RSS} + \lambda\sum_{j=1}^{p}\beta_j^2$$

- λ is a tuning parameter: β_j^λ for each λ
- β_0 is not in the penalty
- Data normalization:

$$\tilde{x}_{i,j} = \frac{x_{i,j} - \bar{x}_j}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_{i,j} - \bar{x}_j)^2}}$$

Prostate: Ridge regression

```
# Calling simple.ridge() function which is part of "ElemStatLearn" package
# Ridge functions in other packages:
# MASS: lm.ridge()
# mda : gen.ridge()
prostate.ridge <- simple.ridge( train[,1:8], train[,9], df=1:8 )

# plot
matplot( prostate.ridge$df, t(prostate.ridge$beta), type="b",
        col="blue", pch=20, ylab="coefficients" )
```



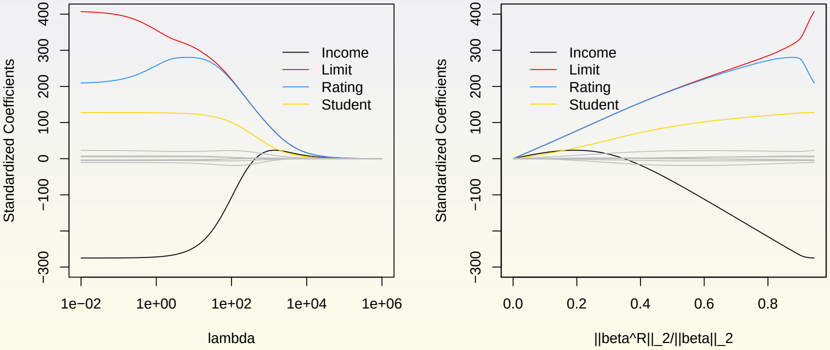
Another example: Credit

- `glmnet` function performs ridge regression, Lasso, ...
- Inputs should be numerical variables

```
> head(credit)
  X  Income Limit Rating Cards Age Education Gender Student Married Ethnicity Balance
1 1  14.891  3606    283     2  34        11   Male      No     Yes Caucasian     333
2 2 106.025  6645    483     3  82        15 Female     Yes     Yes     Asian     903
3 3 104.593  7075    514     4  71        11   Male      No      No     Asian     580
4 4 148.924  9504    681     3  36        11 Female      No      No     Asian     964
5 5  55.882  4897    357     2  68        16   Male      No     Yes Caucasian     331
6 6  80.180  8047    569     4  77        10   Male      No      No Caucasian    1151
> library(glmnet)
> y<-credit$Balance
> x<-model.matrix(Balance~.,credit)[,-c(1,2)]
> head(x)
   Income Limit Rating Cards Age Education GenderFemale StudentYes MarriedYes EthnicityAsian EthnicityCaucasian
1  14.891  3606    283     2  34        11            0          0          1              0                  1
2 106.025  6645    483     3  82        15            1          1          1              1                  0
3 104.593  7075    514     4  71        11            0          0          0              1                  0
4 148.924  9504    681     3  36        11            1          0          0              1                  0
5  55.882  4897    357     2  68        16            0          0          1              0                  1
6  80.180  8047    569     4  77        10            0          0          0              0                  1
> grid<-10^seq(-2,6,length=100)
> credit.ridge<-glmnet(x,y,alpha=0,lambda=grid)


> credit.ridge$lambda[50]
[1] 109.7499
> coef(credit.ridge)[,50]
     (Intercept)           Income            Limit           Rating            Cards              Age
   -295.64236303      -2.80221111       0.09282459       1.37158760      16.34674532      -1.10359021
       Education     GenderFemale       StudentYes       MarriedYes   EthnicityAsian EthnicityCaucasian
     -0.14925505       0.13149217     328.63578560     -12.37346466       8.29714568         7.20848754
> sqrt(sum(coef(credit.ridge)[-1,50]^2))
[1] 329.4747
> credit.ridge$lambda[80]
[1] 0.4132012
> coef(credit.ridge)[,80]
     (Intercept)           Income            Limit           Rating            Cards              Age
    -488.0819858       -7.7661932        0.1634360        1.5382779       15.7906711       -0.6229323
       Education     GenderFemale       StudentYes       MarriedYes   EthnicityAsian EthnicityCaucasian
      -0.9901752      -10.5859576      423.9301301       -9.5031779       17.4513942         10.1743680
> sqrt(sum(coef(credit.ridge)[-1,80]^2))
[1] 425.0184
```

Example: Credit

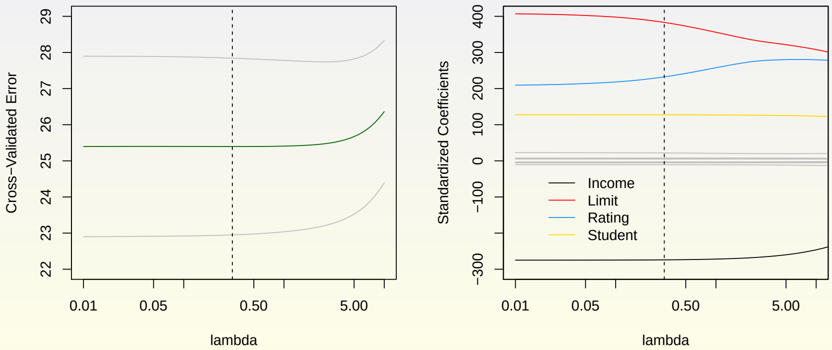
- Standardized coefficients



Example: Credit

- Optimal λ

```
> set.seed(200)
> grid<-10^seq(1,-2,length=100)
> cvridge.out<-cv.glmnet(x,y,alpha=0,lambda = grid)
> cvridge.out$lambda.min
[1] 0.3053856
```



Lasso

- Ridge regression: Still p (shrunk) predictors
- Inference

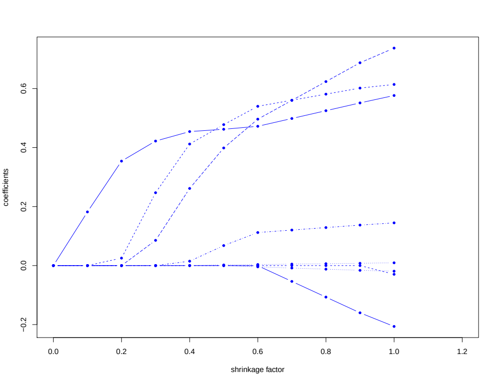
- Lasso: minimize

$$\text{RSS} + \lambda \sum_{j=1}^{p} |\beta_j|$$

- Rationale for absolute values: Some of β_j's will be equal to 0
- Data normalization

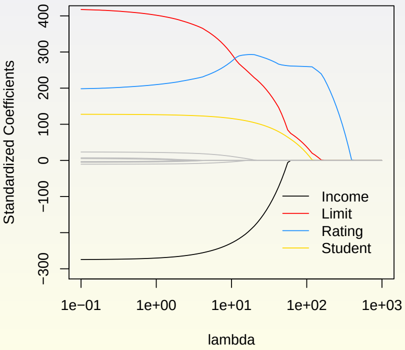
Prostate: Lasso

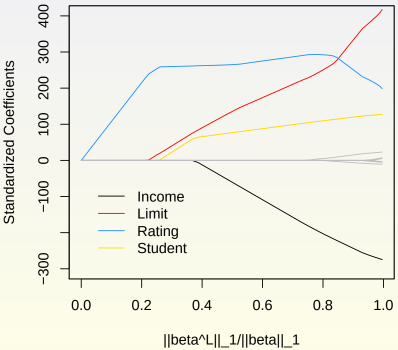
```
# loading package "lasso2"
library(lasso2)
prostate.lasso <- l1ce( lpsa ~ ., data=train, trace=TRUE, sweep.out=~1,
        bound=seq(0,1,by=0.1) )
prostate.lasso.coef <- sapply(prostate.lasso, function(x) x$coef)
colnames(prostate.lasso.coef) <- seq( 0,1,by=0.1 )

# plot
matplot( seq(0,1,by=0.1), t(prostate.lasso.coef[-1,]), type="b",
        xlab="shrinkage factor", ylab="coefficients",
        xlim=c(0, 1.2), col="blue", pch=20 )
```



Another example: Credit

```
> grid<-10^seq(-1,3,length=100)
> credit.lasso<-glmnet(x,y,alpha=1,lambda=grid)
> credit.lasso$lambda[50]
[1] 10.47616
> coef(credit.lasso)[,50]
  (Intercept)        Income         Limit        Rating         Cards               Age
 -468.5205454    -6.4263083     0.1254060     1.7922956     7.7155425        -0.2187646
    Education  GenderFemale    StudentYes    MarriedYes EthnicityAsian EthnicityCaucasian
    0.0000000     0.0000000   384.5532767     0.0000000     0.0000000         0.0000000
```





Ridge vs. Lasso vs. Best subset

- Equivalent formulations
  - Ridge:

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

  - Lasso:

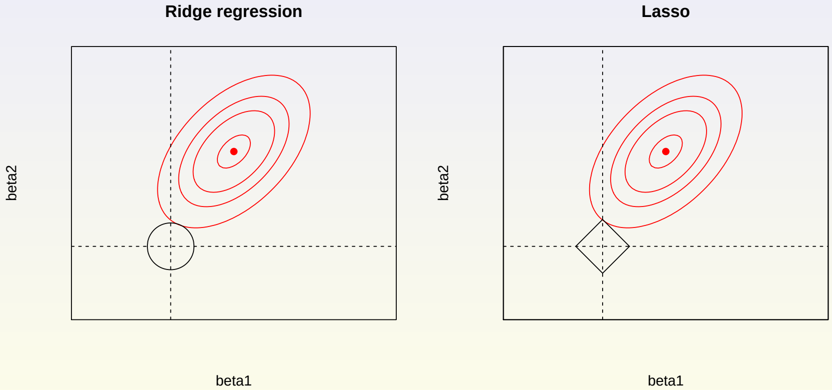
$$\min_{\boldsymbol{\beta}} \text{RSS} \quad \text{subject to } \sum_{j=1}^{p} |\beta_j| \leq s$$

  - Best subset selection:

$$\min_{\boldsymbol{\beta}} \text{RSS} \quad \text{subject to } \sum_{j=1}^{p} 1_{\{\beta_j \neq 0\}} \leq s$$

Lasso vs. Ridge regression

- Geometry



Bias-variance trade-off for linear ridge regression

- Normality assumption: i.i.d. $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$
- Least squares: $\boldsymbol{\beta}_{\mathsf{LS}} = (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ is unbiased, but might have high variance:

$$\mathbb{E}\boldsymbol{\beta}_{\mathsf{LS}} = \boldsymbol{\beta} \quad \text{and} \quad \mathrm{Cov}(\boldsymbol{\beta}_{\mathsf{LS}}) = \sigma^2 (\boldsymbol{X}^\top \boldsymbol{X})^{-1}$$

- Ridge regression: $\boldsymbol{\beta}_{\mathsf{RR}} = (\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{y}$ is biased, but might have lower variance:

$$\mathbb{E}\boldsymbol{\beta}_{\mathsf{RR}} = (\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{X} \boldsymbol{\beta}$$

and

$$\mathrm{Cov}(\boldsymbol{\beta}_{\mathsf{RR}}) = \sigma^2 \boldsymbol{Z} (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{Z}^\top,$$

where

$$\boldsymbol{Z} = (\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{X}^\top \boldsymbol{X}$$

- Note: $(\lambda \boldsymbol{I} + \boldsymbol{X}^\top \boldsymbol{X})$ is full rank: unique solution $\boldsymbol{\beta}_{\mathsf{RR}}$ always exists
Can be solved in dual dot product/kernel formulation for high-dim data ($p \gg n$): $\boldsymbol{\alpha} = (\lambda \boldsymbol{I} + \boldsymbol{K})^{-1} \boldsymbol{y}$ - more on that in the future.

Bias-variance tradeoff for linear regression

- Least squares or Ridge regression?

- How well our solution generalizes to new data?
  Let $(\boldsymbol{x}_0, y_0)$ be future data: $\boldsymbol{x}_0$ is known, but not y_0
- Predictions:
  - Least squares: $\boldsymbol{x}_0^\top \boldsymbol{\beta}_{\mathsf{LS}}$
  - Ridge regression: $\boldsymbol{x}_0^\top \boldsymbol{\beta}_{\mathsf{RR}}$

- Expected squared error of the prediction:

$$\mathbb{E}\left[(y_0 - \boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}})^2 \,|\, \boldsymbol{X}, \boldsymbol{x}_0\right]$$

  - $\boldsymbol{y}$ and y_0 are Gaussian with the true (but unknown) $\boldsymbol{\beta}$

Bias-variance tradeoff for linear regression

- Assuming conditioning on $\boldsymbol{X}$ and $\boldsymbol{x}_0$:

$$\mathbb{E}(y_0 - \boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}})^2 = \mathbb{E}y_0^2 - 2\mathbb{E}y_0\, \boldsymbol{x}_0^\top\, \mathbb{E}\hat{\boldsymbol{\beta}} + \boldsymbol{x}_0^\top\, \mathbb{E}[\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top]\, \boldsymbol{x}_0$$

- Since $\mathbb{E}y_0^2 = (\boldsymbol{\beta}^\top \boldsymbol{x}_0)^2 + \sigma^2$ and

$$\mathbb{E}\hat{\boldsymbol{\beta}}\hat{\boldsymbol{\beta}}^\top = \mathbb{E}\hat{\boldsymbol{\beta}}\, \mathbb{E}\hat{\boldsymbol{\beta}}^\top + \mathsf{Cov}(\hat{\boldsymbol{\beta}})$$

- We have

$$\begin{aligned}\mathbb{E}(y_0 - \boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}})^2 &= \sigma^2 + \boldsymbol{x}_0^\top (\boldsymbol{\beta} - \mathbb{E}\hat{\boldsymbol{\beta}})(\boldsymbol{\beta} - \mathbb{E}\hat{\boldsymbol{\beta}})^\top \boldsymbol{x}_0 + \boldsymbol{x}_0^\top \mathsf{Cov}(\hat{\boldsymbol{\beta}})\, \boldsymbol{x}_0 \\ &= \text{noise} + \text{bias}^2 + \text{variance}\end{aligned}$$

  - LS: $\mathbb{E}(y_0 - \boldsymbol{x}_0^\top \hat{\boldsymbol{\beta}})^2 = \sigma^2 + \sigma^2 \boldsymbol{x}_0^\top (\boldsymbol{X}^\top \boldsymbol{X})^{-1} \boldsymbol{x}_0$

Reading:

ISL: Finish reading Chapter 6

ESL: Chapter 3: Sections 3.3 - end of chapter.

Homework: Homework 1 next Tue - Sep 27.