IEOR E4650 Business Analytics

Session 7: Linear Model Selection

Spring 2018

Prof. Adam Elmachtoub

Model Selection for Linear Regression

- Options for choosing predictors:
  - Domain-specific knowledge
  - Use everything
  - Statistical selection - let the data decide
- Subset Selection (predictor selection).
  - Best Subset Selection
  - Forward Stepwise Selection
- Shrinkage Method
  - Ridge Regression
  - Lasso Regression
  - Elastic Net

Why Linear Regression Can Fail

- If $n \gg p$ linear model tends to perform well on test data
- If n is not much larger than p, model can overfit data and result in poor predictions.
- In practice, p can be quite large
  - Genetic data and search data naturally have very large p
  - Start with 30 raw features, then add transformations and interaction terms leads to over a thousand independent variables
- When p is large, linear regression models tend to overfit and are difficult to interpret
- By constraining the number of predictors or shrinking the $\widehat{\beta}$'s, we can reduce variance at the cost of negligible increase in bias
- By reducing the number of predictors, we can improve the interpretability of the model

Alternative methods for linear regression

- Recall that we are trying to estimate true parameters β_0, $\beta = (\beta_1, \ldots, \beta_p)$
- Subset Selection: For every $t \leq p$, find the best model of size t:

$$RSS_{sub}(t) = \min_{\widehat{\beta}_0, \widehat{\beta}} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}' x_i)^2 \text{ subject to } \sum_{j=1}^{p} I(\widehat{\beta}_j \neq 0) \leq t.$$

- Shrinkage: For a budget t on the norm of $\widehat{\beta}$, find the best model for the L_1 and L_2 norms:

$$RSS_{lasso}(t) = \min_{\widehat{\beta}_0, \widehat{\beta}} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}' x_i)^2 \text{ subject to } \sum_{j=1}^{p} |\widehat{\beta}_j| \leq t.$$

$$RSS_{ridge}(t) = \min_{\widehat{\beta}_0, \widehat{\beta}} \sum_{i=1}^{n} (y_i - \widehat{\beta}_0 - \widehat{\beta}' x_i)^2 \text{ subject to } \sum_{j=1}^{p} \widehat{\beta}_j^2 \leq t.$$

Best-subset selection

- How many linear models can we build given p independent variables?
- Consider the set of linear models:

$$Y = \beta_0 + \beta_1 \delta_1 X_1 + \beta_2 \delta_2 X_2 + \cdots + \beta_p \delta_p X_p + \epsilon.$$

where $\delta_i \in \{0, 1\}$ for each $i = 1, \ldots, p$

- There are 2^p linear models
- Best-subset selection with parameter t aims to find best model with only t independent variables
- If $t = p$, this is just standard linear regression.

Why subset selection?

Chose a subset of the predictors

- Recall the bias-variance tradeoff:
  - A model with many predictors may have low bias but high variance
  - A model with few predictors may have high bias and low variance
- We want the right value of t that minimizes the test error, i.e., the sum of the bias squared plus the variance
- We will choose the value of t using the model selection techniques from last time

How to implement subset selection

- If t is small, just try all $\binom{p}{t}$ models
  - Best subset selection only works if t is relatively small
- When t is large, use forward stepwise regression
  - Add predictors to the model, *one-at-a-time*
  - At each step, the variable that gives the greatest additional improvement to the fit is added to the model.
  - We will first find a good models with $0, 1, 2, \ldots, p$ features, and call them $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$
  - Then we will choose one among these $p + 1$ models

Best Subset Selection: Traditional approach

1. For each $t = 1, \ldots, p$:
   a) Fit all $\binom{p}{t}$ models that contain exactly t predictors on training data $(75\%$ of data).
   b) Pick the best among these models and call it $\mathcal{M}_t$. Here best is defined as having the smallest MSE.

2. Select single best model among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ using adjusted R^2.

3. Evaluate MSE of final chosen model on test data $(25\%$ of data)

Best Subset Selection: Train-Validation-Test approach

1. For each $t = 1, \ldots, p$:
   a) Fit all $\binom{p}{t}$ models that contain exactly t predictors on training data (50% of data).
   b) Pick the best among these models and call it $\mathcal{M}_t$. Here best is defined as having the smallest MSE on training data.

2. Compute MSE of $\mathcal{M}_t$ on the validation data (25% of data) and call this MSE_t.

3. Let t^* be the size of the model with smallest MSE_t, i.e., the number of predictors in the best model.

4. Find the best model with t^* predictors using combined training and validation data (75% of data).

5. Evaluate MSE of final chosen model on test data (25% of data)

Best Subset Selection: k-Fold Cross-validation approach

1. For each $t = 1, \ldots, p$ and for each $j = 1, \ldots, k$
   a) Let training set j be the training data with fold j removed.
   b) Fit all $\binom{p}{t}$ models that contain exactly t predictors on training set j
   c) Pick the best among these models using MSE on training set j.
   d) Let MSE_{jt} be the MSE of the best model from c) on fold j
   e) Use average MSE to estimate performance of using t variables, i.e., $MSE_t = \frac{1}{k}\sum_{j=1}^{k} MSE_{jt}$

2. Let t^* be the size of the model with smallest MSE_t, i.e., the number of predictors in the best model.

3. Find the best model with t^* predictors using entire training data (75% of data).

4. Evaluate MSE of final chosen model on test data (25% of data)

Best Subset Selection: Example

- Use Hitters data set
- We will do only model selection, using traditional approach
- Want to predict a baseball player's Salary on the basis of various statistics associated with their performance
- Sample R code: data dimension, missing values

```
> library(ISLR)
> fix(Hitters)  #puts data into Excel sheet format
> names(Hitters)
> Hitters[1:5,]
> Hitters2=na.omit(Hitters) #removes rows with missing data
> dim(Hitters2)
[1] 263  20
```

Best Subset Selection: Example, Cont.

- The regsubsets() performs best subset selection by identifying the best model that contains a given number of predictors.
- regsubsets() is part of the leaps library; similar to lm()
- By default, regsubsets() only reports results up to the best eight-variable model. Use nvmax option to change it
    - Sample R code

```
> library(leaps)
> regfit.full=regsubsets(Salary~.,data=Hitters2,nvmax=19)
> reg.summary=summary(regfit.full)
> names(reg.summary)
[1] "which" "rsq"  "rss" "adjr2" "cp" "bic" "outmat" "obj"
```

Best Subset Selection: Example, Cont.

- Pick model with highest adjusted R^2

```
> names(reg.summary)
[1] "which" "rsq"  "rss" "adjr2" "cp" "bic" "outmat" "obj"
> reg.summary["rsq"]
$rsq
[1] 0.3214501 0.4252237 0.4514294 0.4754067 0.4908036 0.5087146
[7] 0.5141227 0.5285569 0.5346124 0.5404950 0.5426153 0.5436302
[13] 0.5444570 0.5452164 0.5454692 0.5457656 0.5459518 0.5460945
[19] 0.5461159
> which.max(reg.summary$adjr2)
[1] 11
> coef(regfit.full,11)
(Intercept)        AtBat         Hits        Walks       CAtBat
135.7512195   -2.1277482    6.9236994    5.6202755   -0.1389914
CRuns          CRBI       CWalks      LeagueN    DivisionW
1.4553310     0.7852528   -0.8228559   43.1116152 -111.1460252
PutOuts       Assists
0.2894087     0.2688277
```

Forward Stepwise Selection: Traditional approach

1. For $t = 0, \ldots, p-1$:
   a) Consider all $p-t$ models that augment the predictors in $\mathcal{M}_t$ by one predictor on training data (75%).
   b) Chose the best among these $p-t$ models, and call it $\mathcal{M}_{t+1}$. Here *best* is defined as having the smallest MSE.

2. Select single model among $\mathcal{M}_0, \mathcal{M}_1, \ldots, \mathcal{M}_p$ using adjusted R^2.

3. Evaluate MSE of final chosen model on test data $(25\%$ of data$)$

Forward Stepwise Selection: Train-Validation-Test approach

1. For each $t = 0, \ldots, p-1$:
   a) Consider all $p - t$ models that augment the predictors in $\mathcal{M}_t$ by one predictor on training data (50%)
   b) Pick the best among these models and call it $\mathcal{M}_{t+1}$. Here best is defined as having the smallest MSE.

2. Compute MSE of $\mathcal{M}_t$ on the validation data (25% of data) and call this MSE_t.

3. Let t^* be the size of the model with smallest MSE_t, i.e., the number of predictors in the best model.

4. Do forward stepwise selection using t^* predictors on combined training and validation data (75% of data).

5. Evaluate MSE of final model on test data (25% of data)

Forward Stepwise Selection: k-Fold Cross-validation approach

1. For each $j = 1, \ldots, k$
   a) Let training set j be the training data with fold j removed.
   b) Let $\mathcal{M}_{jt}$ be the size t model chosen using forward selection on training set j
   d) Let MSE_{jt} be the MSE of $\mathcal{M}_{jt}$ on fold j
   e) Use average MSE to estimate performance using t variables, i.e., $MSE_t = \frac{1}{k}\sum_{j=1}^{k} MSE_{jt}$

2. Let t^* be the size of the model with smallest MSE_t, i.e., the number of predictors in the best model.

3. Do forward stepwise selection using t^* predictors on entire training data (75% of data).

4. Evaluate MSE of final chosen model on test data (25% of data)

Forward Stepwise Selection: Example

- Use Hitters data set
- We will do only model selection, using training-validation approach (no model assessment-no test data)

```
> train=sample(c(TRUE,TRUE,FALSE),nrow(Hitters2),rep=TRUE)
> val=(!train)
> val.mat=model.matrix(Salary~.,data=Hitters2[val,])
> regfit.fwd=regsubsets(Salary~.,data=Hitters2[train,],nvmax=19,method="forward")
> val.mse=rep(NA,19)
> for(t in 1:19){        #regsubsets has no natural predict function :(
+     coefi=coef(regfit.fwd,id=t)
+     pred=val.mat[,names(coefi)]%*%coefi
+     val.mse[t]=mean((Hitters2$Salary[val]-pred)^2)
+ }
> val.mse
> t_star=which.min(val.mse)
> regfit.fwd=regsubsets(Salary~.,data=Hitters2,nvmax=t_star,method="forward")
> coef(regfit.fwd,t_star)
```

Computational Complexity Comparison

- Best Subset Selection involves 2^p models
- Forward Stepwise Selection involves $\sum_{k=0}^{p-1}(p-k) = 1 + p(p+1)/2$ models.
- For $p = 20$, it is $1,048,576$ vs. 211.
- Forward Stepwise Selection does well in practice.

Shrinkage Methods: Ridge Regression

- Ridge Regression is often formulated by dualizing the constraint $||\beta||_2 \leq t$, resulting in the problem

$$\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \beta' x_i)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right\},$$

  where the Lagrangian $\lambda \geq 0$ can be viewed as a tuning parameter.

- The red term is a shrinkage penalty.
  - If $\lambda = 0$, the penalty term has no effect, i.e., it produces the least squares estimates.
  - As λ increases, the flexibility decreases, i.e., variance decreases, but bias increases.
  - If $\lambda = \infty$, $\beta = 0$. Equivalent to the NULL model.

Ridge Regression on Credit Data Set

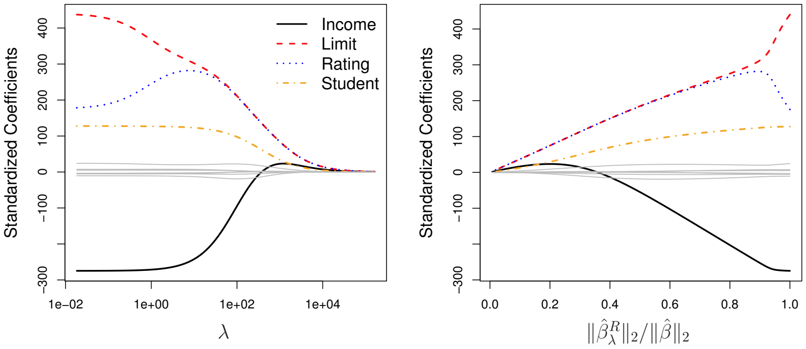


Figure: The standardized ridge regression coefficients are displayed for the Credit data set, as a function of λ and $||\hat{\beta}_\lambda^R||_2/||\hat{\beta}||_2$.

Ridge Regression and Bias Variance Tradeoff

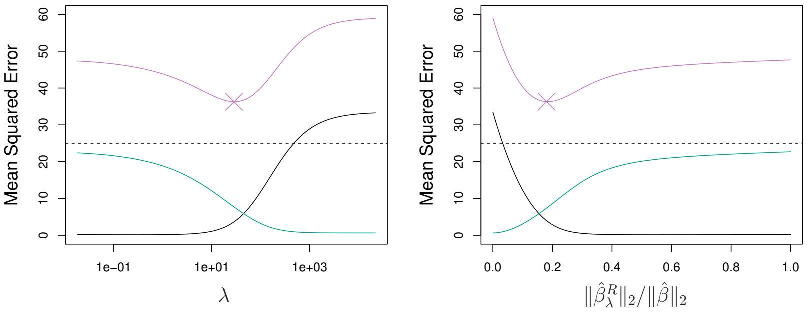


Figure: Squared bias (black), variance (green), and test mean squared error (purple) for the ridge regression predictions on a simulated data set.

Shrinkage Methods: Lasso Regression

- Lasso Regression is often formulated by dualizing the constraint $||\beta||_1 \leq t$, resulting in the problem

$$\min_{\beta_0,\beta} \left\{ \sum_{i=1}^{n} (y_i - \beta_0 - \beta' x_i)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \right\},$$

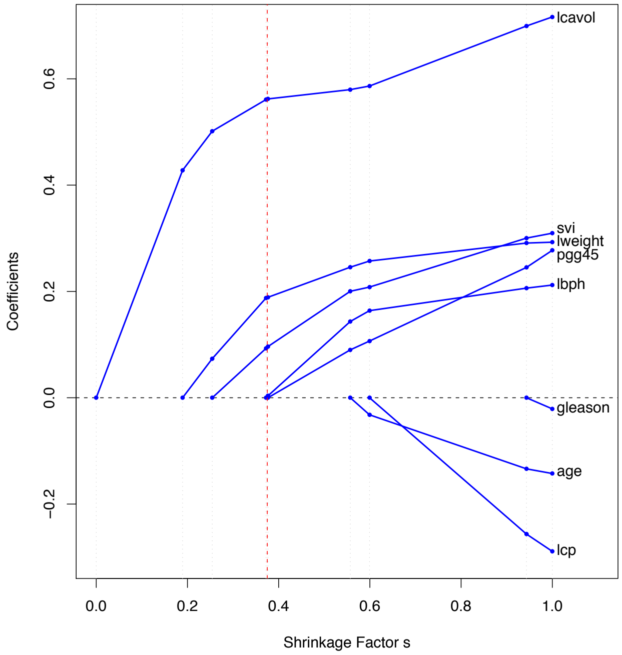
  where the Lagrangian $\lambda \geq 0$ can be viewed as a tuning parameter.

- The red term is a shrinkage penalty.
  - If $\lambda = 0$, the penalty term has no effect, i.e., it produces the least squares estimates.
  - As λ increases, the flexibility decreases, i.e., variance decreases, but bias increases.
  - If $\lambda = \infty$, $\beta = 0$. Equivalent to the NULL model.

LASSO, Cont.

- Let $\widehat{\beta}$ be the standard least squares estimate and let $t_0 = \sum_{j=1}^{p} |\widehat{\beta}_j|$ be its L_1 norm.
- Values $t \geq t_0$ do NOT affect the least squares minimization.
- $t < t_0$ leads to a shrinkage of the least squares solution;
- Some coefficients will be 0 exactly, leading to variable selection and a simplification of the model.
- If $t = 0$, all estimated coefficients are shrunk to 0

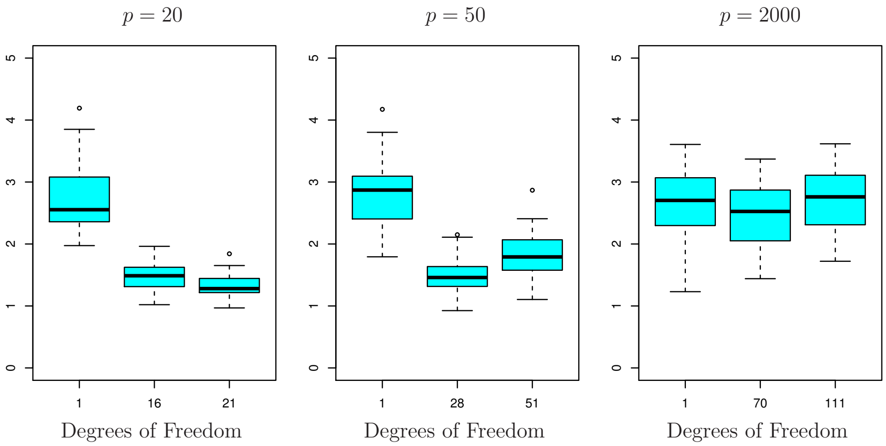
Shrinkage factor $s = t/t_0$ for heart data



$$s = \frac{t}{t_0} \geq 1, \tilde{\beta} = \hat{\beta}$$

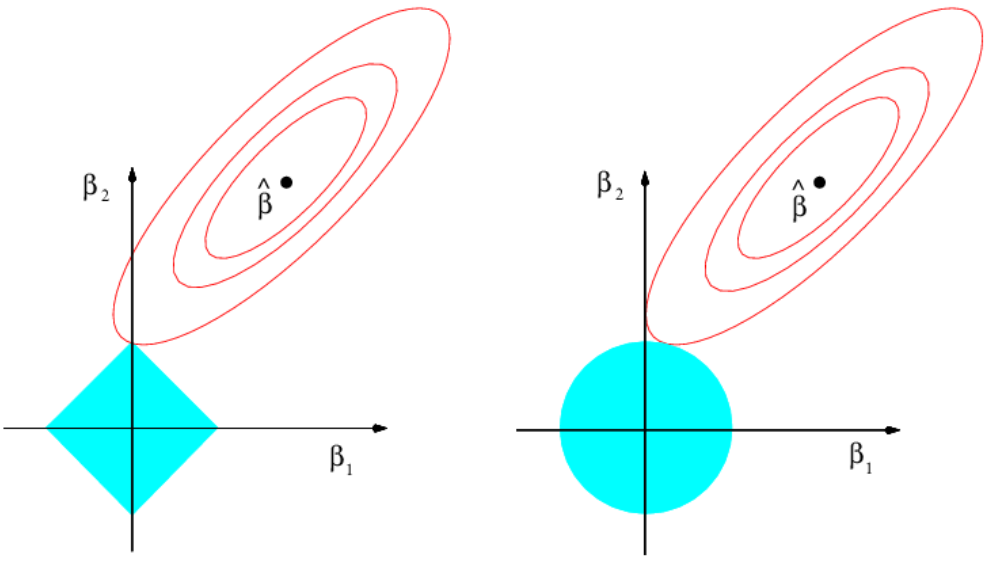
$$s = \frac{t}{t_0} < 1, |\tilde{\beta}| < |\hat{\beta}|$$

Regression in High Dimensions



- $n = 100$, only $p = 20$ of the predictors are truly associated with y.
- Plots show test errors on models selected by Lasso.

Lasso (left) vs Ridge (right): Graphical Solution



Shrinkage Method: Train-Validation-Test approach

1. Select candidate values for λ from 0 to ≈ 1000
2. Solve ridge/lasso for each value of λ on training data (50 %)
3. Compute MSE of each model on validation data (25% of data)
4. Let λ^* be the choice of λ corresponding to the smallest MSE on the validation data
5. Solve ridge/lasso using λ^* on combined training and validation data (75% of data).
6. Evaluate MSE of final chosen model on test data (25% of data)

Shrinkage Method: k-Fold Cross-validation approach

1. Select candidate values for λ from 0 to ≈ 1000
2. Let training set j be the training data with fold j removed.
3. Solve ridge/lasso for each value of λ on each training data j
4. Compute the MSE for each λ using the cross-validation technique
5. Let λ^* be the choice of λ corresponding to the smallest MSE on the validation data
6. Solve ridge/lasso using λ^* on entire training data (75%)
7. Evaluate MSE of final chosen model on test data (25% of data)

Elastic Net: A compromise between Lasso and Ridge

- Recall that Lasso use the L_1 norm and Ridge uses the L_2 norm
- Using L_q for $q \in (1, 2)$ suggest a compromise between Lasso and Ridge regression
- However, for $q > 1$, $|\beta_j|^q$ is differentiable at 0, so will not set coefficients to zero as Lasso.
- A compromise is to have the elastic net penalty

$$\lambda \sum_{j=1}^{p} \left((1-\alpha)\beta_j^2 + \alpha|\beta_j| \right).$$

- The elastic-net selects variables like the Lasso
- Shrinks together the coefficients of correlated predictors like Ridge.

Numerical Examples: Ridge and LASSO Regression

- We can use package `glmnet` to perform Ridge and LASSO regression.
    - Function `glmnet()` has an `alpha` argument
    - `alpha=0`, a ridge regression model is a fit
    - `alpha=1`, a lasso model is a fit
    - `lambda` is a tunning parameter.
- Package `lars` can perform LASSO.

Numerical Examples: Sample R Code

- Sample R code

```
> library(ISLR)
> fix(Hitters)
> Hitters2=na.omit(Hitters)
> dim(Hitters2)
[1] 263  20

grid=10^seq(10,-2,length=100)
x=model.matrix(Salary~.,Hitters2)[,-1]
y=Hitters2$Salary
library(glmnet)

ridge.mod=glmnet(x,y,alpha=0,lambda=grid)
dim(coef(ridge.mod))

ridge.mod$lambda[50]
coef(ridge.mod)[,50]

ridge.mod$lambda[60]
coef(ridge.mod)[,60]
```

Numerical Examples: Sample R Code, Cont.

- Function `predict()`:
  - Obtain rige regression coefficients for a new λ
  - Obtain predictions for a test set

```
predict(ridge.mod, s=705, type="coefficients")[1:20,]

predict(ridge.mod, s=50, type="coefficients")[1:20,]
```

Ridge regression and Cross Validadion

- We can also do cross validation with Ridge regression
- Look up cv.glmnet to see how to specify choice of λ's and folds

```
set.seed(1)
#split data 75% train and 25% test for cross-validation
train=sample(c(TRUE,TRUE,TRUE,FALSE),nrow(Hitters2),rep=TRUE)
test=(!train)
cv.out=cv.glmnet(x[train,],y[train],type.measure="mse",alpha=0,lambda=grid)
plot(cv.out)
bestlam=cv.out$lambda.min
ridge.mod=glmnet(x[train,],y[train],alpha=0,lambda=bestlam)
ridge.pred=predict(ridge.mod,newx=x[test,])
mean((ridge.pred-y[test])^2)
```

LASSO and Cross Validadion

- We can also do cross validation with Lasso regression

```
set.seed(1)
train=sample(c(TRUE,TRUE,TRUE,FALSE),nrow(Hitters2),rep=TRUE)
test=(!train)
cv.out=cv.glmnet(x[train,],y[train],type.measure="mse",alpha=1,lambda=grid)
plot(cv.out)
bestlam=cv.out$lambda.min
las.mod=glmnet(x[train,],y[train],alpha=1,lambda=bestlam)
las.pred=predict(ridge.mod,newx=x[test,])
mean((las.pred-y[test])^2)
```

Subset Selection vs Ridge vs Lasso

- Subset Selection is computationally more expensive (not practical for large p).
- Ridge and Lasso regression can be computed very efficiently (almost as efficiently as doing unconstrained linear regression).
- Lasso produces simpler, more interpretable models that involves only a subset of predictors.
- Lasso is better at detecting and removing irrelevant predictors.
- Not clear which model leads to better prediction accuracy.

Summary of Linear Model Selection

- When p is large, we need automated ways to come up with good candidate models.
- Subset Selection: best-subset, forward stepwise regression.
- Shrinkage Methods: Ridge Regression, LASSO, Elastic Net
- Using R to do subset selection, ridge regression and LASSO.
- Use Cross-Validation on training data to find best parameter (model selection), then use test data to measure performance (model assessment)