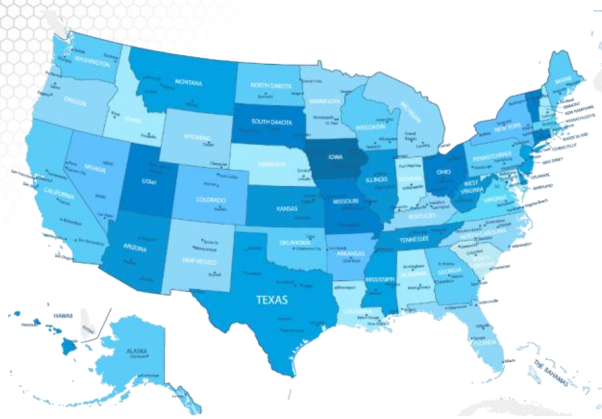
Regression Analysis
Model Selection

Nicoleta Serban, Ph.D.
Professor
Stewart School of Industrial and Systems Engineering

Regularized Regression:
Data Examples

About This Lesson

Ranking States by SAT Performance



SAT Mean Score by State – Year 1982
790 (South Carolina) – 1088 (Iowa)

- ***Which variables are associated with state average SAT scores?***
- ***After accounting for selection biases, how do the states rank?***
- ***Which states perform best for the amount of money they spend?***

Georgia Tech

Ridge Regression

```
library(MASS)
## Scale the predicting variables and the response variable
ltakers = log(takers)
predictors = cbind(ltakers, rank, income, years, public, expend)
predictors = scale(predictors)
sat.scaled = scale(sat)

## Apply ridge regression for a range of penalty constants
lambda = seq(0, 10, by=0.25)
out = lm.ridge(sat.scaled~predictors, lambda=lambda)
round(out$GCV, 5)

which(out$GCV == min(out$GCV))
 2.25
   10
round(out$coef[,10], 4)
predictorsltakers    predictorsrank  predictorsincome   predictorsyears  predictorspublic
          -0.4771            0.4195            0.0223            0.1796           -0.0028
 predictorsexpend
           0.1808
```

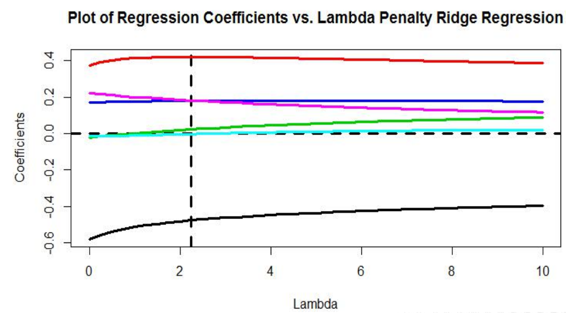
The ridge regression outputs estimates for each lambda in the considered range *(not shown)*

The lambda is selected to minimize the (generalized) CV score

Georgia Tech

Ridge Regression

```
plot(lambda, out$coef[1,], type = "l", col=1, lwd=3,
        xlab = "Lambda", ylab = "Coefficients",
        main = "Plot of Regression Coefficients vs. Lambda
Penalty Ridge Regression",
        ylim = c(min(out$coef), max(out$coef)))
for(i in 2:6)
    points(lambda, out$coef[i,], type = "l", col=i, lwd=3)
abline(h = 0, lty = 2, lwd = 3)
abline(v = 2.25, lty = 2, lwd=3)
```



Georgia Tech

LASSO Regression

```
library(lars)
object = lars(x=predictors, y=sat.scaled)
Object
```

Sequence of LASSO moves:

	ltakers	rank	years	expend	income	public
Var	1	2	4	6	3	5
Step	1	2	3	4	5	6

round(object$Cp,2)

0	1	2	3	4	5	6
349.91	103.40	46.89	35.64	3.10	5.09	7.00

The selected model according to Malow's Cp is at the fourth variable introduced in the model.

Georgia Tech

LASSO Regression: First Implementation

library(lars)
object = lars(x=predictors, y=sat.scaled)
Object

Sequence of LASSO moves:

	ltakers	rank	years	expend	income	public
Var	1	2	4	6	3	5
Step	1	2	3	4	5	6

round(object$Cp,2)

0	1	2	3	4	5	6
349.91	103.40	46.89	35.64	3.10	5.09	7.00

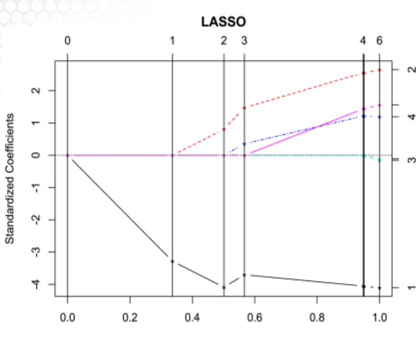
The selected model according to Malow's Cp is at the fourth variable introduced in the model.

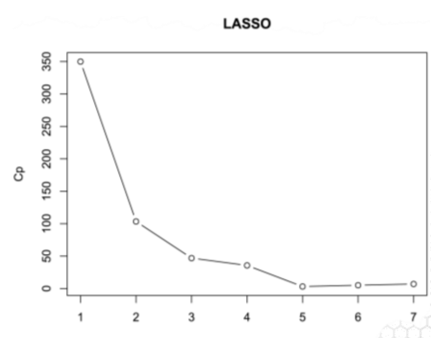
- From the order the predictors were added, i.e., log(*takers*), *rank*, *years*, *expend*, *income* and *public*, the first four are selected
- After LASSO variable selection, apply ordinary least squares (OLS) with the selected predicting variables

Georgia Tech

LASSO Regression: First Implementation

plot.lars(object)
plot.lars(object, xvar="df", plottype="Cp")

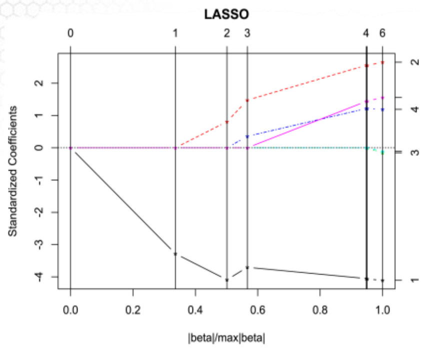


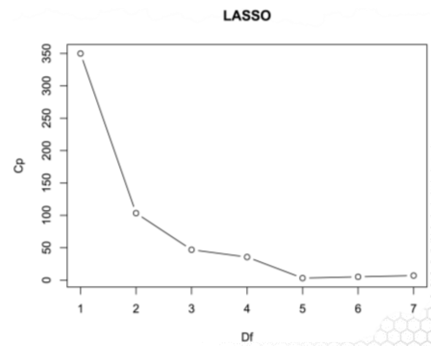


Georgia Tech

LASSO Regression: First Implementation

plot.lars(object)
plot.lars(object, xvar="df", plottype="Cp")





From the order the predictors were added, i.e., log(*takers*), *rank*, *years*, *expend*, *income* and *public*, the first four are selected

Georgia Tech

LASSO Regression: Second Implementation

```
library(glmnet) # alpha=1 lasso, alpha=0 ridge
Xpred= cbind(ltakers, rank, income, years, public, expend)

# Find the optimal lambda using 10-fold CV
satmodel.cv=cv.glmnet(Xpred, sat, alpha=1, nfolds=10)

## Fit lasso model with 100 values for lambda
satmodel = glmnet(Xpred, sat, alpha = 1, nlambda=100)

## Extract coefficients at optimal lambda
coef(satmodel, s=satmodel.cv$lambda.min)
```

```
(Intercept)  516.519096
ltakers      -37.244873
rank         3.420034
income       .
years        13.780563
public       .
expend       1.662338
```

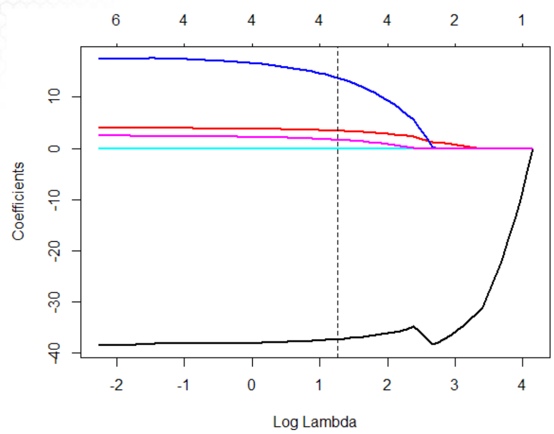
Because CV uses random assignments, expect slightly different coefficients each time it is run.

Using LASSO and the penalty selected using 10-fold CV, the selected predictors are: log(*takers*), *rank*, *years*, and *expend*

Georgia Tech

LASSO Regression: Second Implementation

```
## Plot coefficient paths
plot(satmodel,xvar="lambda", lwd=2)
abline(v=log(satmodel.cv$lambda.min), col='black', lty=2)
```



Elastic Net Regression

```
library(glmnet) # alpha=1 lasso, alpha=0 ridge
Xpred= cbind(ltakers, rank, income, years, public, expend)

# Find the optimal lambda using 10-fold CV
satmodel.cv=cv.glmnet(Xpred, sat, alpha=0.5, nfolds=10)

## Fit elastic net model with 100 values for lambda
satmodel = glmnet(Xpred, sat, alpha=0.5, nlambda = 100)

## Plot coefficient paths
coef(satmodel, s=satmodel.cv$lambda.min)
```

(Intercept)	386.70798566
ltakers	-32.76143283
rank	4.24246653
income	0.01727723
years	16.40041447
public	.
expend	1.83649835

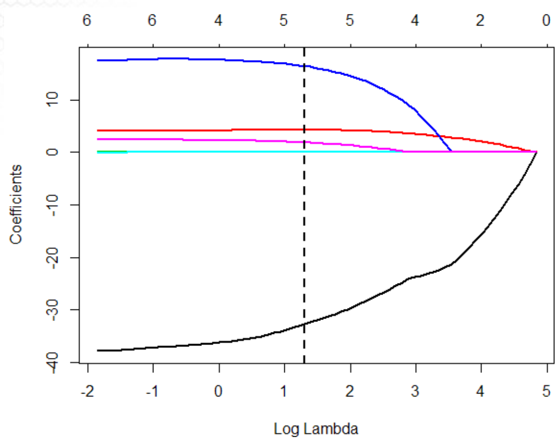
Because CV uses random assignments, expect slightly different coefficients each time it is run.

Using Elastic Net and the penalty selected using 10-fold CV, the selected predictors are: log(*takers*), *rank*, *income*, *years*, and *expend*

Elastic Net

Extract coefficients at optimal lambda

```
plot(satmodel, xvar="lambda", lwd=2)
abline(v=log(satmodel.cv$lambda.min), col='black', lty=2, lwd=2)
```



Overview of All Selection Approaches

	Log(Takers)	Rank	Income	Years	Public	Expend
Best Subset & Mallow's Cp		×		×	×	×
Stepwise & AIC	×	×		×		×
LASSO & Mallow's Cp	×	×		×		×
Lasso & 10-fold CV	×	×		×		×
Elastic Net & 10-fold CV	×	×	×	×		×

- *Rank*, *Years*, and *Expend* are selected by all approaches
- Best Subset alone selects *Public* and does not select *Takers*
- *Income* is selected only by Elastic Net

Predicting Bankruptcy

- Effective bankruptcy prediction is useful for investors and analysts, allowing for accurate evaluation of a firm's prospects.
- Roughly 40 years ago, Ed Altman showed that publicly available financial indicators can be used to distinguish between firms that are about to go bankrupt and those that are not.

Which financial indicators are associated with bankruptcy for telecommunications firms?

Acknowledgement: This example was provided by Dr. Jeffrey Simonoff from New York University and was inspired by the honors thesis of Jeffrey Lui.

Georgia Tech

LASSO Regression

```
library(glmnet)
X = cbind(WC.TA, RE.TA, EBIT.TA, S.TA, BVE.BVL)

## 10-fold CV to find the optimal lambda
bank5.cv = cv.glmnet(X, Bankrupt, family=c("binomial"), alpha=1, type="class", nfolds=10)

## Fit lasso model with 100 values for lambda
bank5 = glmnet(X, Bankrupt, family=c("binomial"), alpha=1, nlambda=100)

## Extract coefficients at optimal lambda
coef(bank5, s=bank5.cv$lambda.min)
```

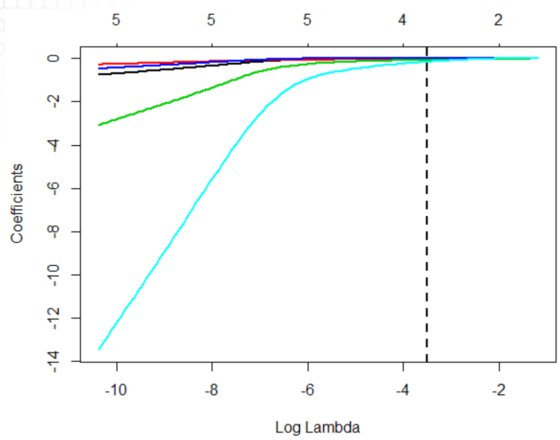
(Intercept)	-0.95995368
WC.TA	.
RE.TA	-0.02874387
EBIT.TA	-0.05757731
S.TA	.
BVE.BVL	-0.14135425

Using LASSO and the penalty selected using 10-fold CV, the selected predictors are: *RE.TA*, *EBIT.TA*, and *BE.BVL*

Georgia Tech

LASSO Regression

Plot coefficient paths
plot(bank5, xvar="lambda", lwd=2)
abline(v=log(bank5.cv$lambda.min), col='black', lty 2, lwd=2)



Georgia Tech

Elastic Net Regression

library(glmnet) **# alpha=1 lasso, alpha=0 ridge**
X = cbind(WC.TA,RE.TA,EBIT.TA,S.TA,BVE.BVL)

10-fold CV to find the optimal lambda
bank6.cv = cv.glmnet(X, Bankrupt, family=c("binomial"),alpha= 0.5, type="class", nfolds=10)

Fit elastic net model with 100 values for lambda
bank6 = glmnet(X, Bankrupt, family=c("binomial"),alpha=0.5,nlambda=100)

Extract coefficients at optimal lambda
coef(bank6, s=bank6.cv$lambda.min)

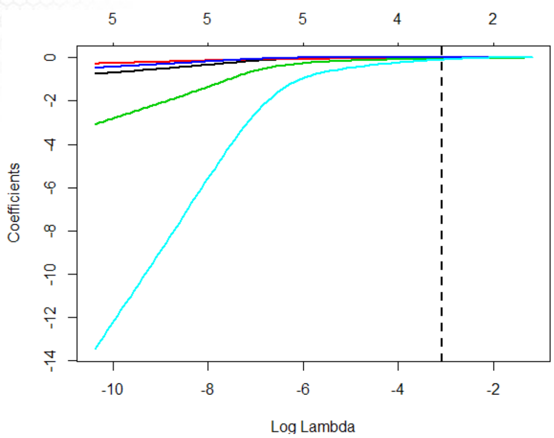
(Intercept)	-0.572208580
WC.TA	-0.006693551
RE.TA	-0.015677213
EBIT.TA	-0.050962740
S.TA	.
BVE.BVL	-0.108940580

Using Elastic Net and the penalty selected using 10-fold CV, the selected predictors are: *WC.TA*, *RE.TA*, *EBIT.TA*, and *BVE.BVL*

Georgia Tech

Elastic Net

Plot coefficient paths
plot(bank5, xvar="lambda", lwd=2)
abline(v=log(bank6.cv$lambda.min), col='black', lty=2, lwd=2)



Overview of All Selection Approaches

	WC.TA	RE.TA	EBIT.TA	S.TA	BVE.BVL
Best subset AIC		×	×		×
Stepwise & AIC		×	×		×
Lasso & 10-fold CV		×	×		×
Elastic Net & 10-fold CV	×	×	×		×

Summary

Georgia Tech