NTNU
Norwegian University of Science and Technology

TMA4267 Linear Statistical Models V2014 (24)

Shrinkage [ISLR6.2]
Dimension reduction [ISLR6.3]
Principal component analysis [ISLR10.2]

Mette Langaas

To be lectured: March 25, 2014
wiki.math.ntnu.no/emner/tma4267/2014v/start/

Lecture 24: Model selection (LS, subset)
Shrinkage SSE + penalty → ridge, → lasso

Shrinkage methods [ISLR 6.2, cont.]

$$\min_{\beta} \left(SSE + \lambda \sum_{j=1}^{p} \beta_j^2 \right) \rightarrow \hat{\beta}^R \quad \text{Ridge}$$

$$\min_{\beta} \left(SSE + \lambda \sum_{j=1}^{p} |\beta_j| \right) \rightarrow \hat{\beta}^L \quad \text{Lasso}$$

Simple exemple

$n = p$, no intercept in model $\quad Y_i = \beta_1 X_{1i} + \beta_2 X_{2i} + \dots + \beta_p X_{pi} + \varepsilon_i$

$\mathbb{X}$ is diagonal : $\underset{n \times p}{\mathbb{X}} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ 0 & 0 & & \vdots \\ \vdots & \vdots & & \\ 0 & 0 & & 1 \end{bmatrix}$

1) LS

$$SSE = \sum_{i=1}^{n} (y_i - \beta_1 x_{i1} - \dots - \beta_p x_{ip})^2$$

but $x_{ji} = 1$ for $i = j$ and 0 else.

$$SSE = \sum_{j=1}^{p} (y_j - \beta_j)^2 \Rightarrow \hat{\beta}_j = y_j \qquad \left(\text{check } (X^T X)^{-1} X^T Y\right)$$

LS solution

Remark: this will be a perfect fit and all residuals = 0.

2) Ridge.

$$\min_{\beta} \left(\sum_{j=1}^{p} (y_j - \beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2 \right)$$

can be shown that

$$\hat{\beta}_j^R = \frac{y_j}{1+\lambda}$$

$(X^T X + \lambda I)^{-1} \underbrace{X^T Y}_{\text{choose one } y}$

$X^T X + \lambda I$: $\text{diag}(1+\lambda)$

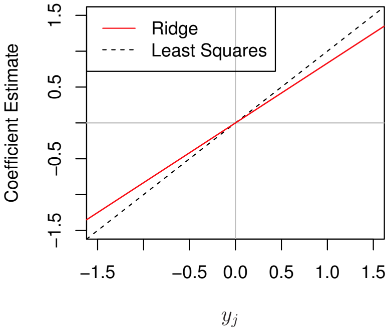
3) Lasso

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

gives

$$\hat{\beta}_j^L = \begin{cases} y_j - \frac{\lambda}{2} & y_j > \frac{\lambda}{2} \\ y_j + \frac{\lambda}{2} & y_j < -\frac{\lambda}{2} \\ 0 & |y_j| \leq \frac{\lambda}{2} \end{cases}$$

Simple example: Ridge and lasso



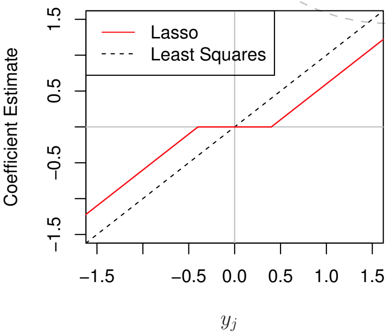


Figure 6.10 from An Introduction to Statistical Learning (2013)

For more complex situations:

* RR shrinks (more or less) every dimension with the same factor.

* Lasso ———— " ———— all coefficients towards zero with similar amount, and sufficiently small coefficients all the way to zero.

Selecting the tuning parameter λ

Method: cross validation [more in TMA4300 Comp. stat and ch5 of ISLR]

[NB: AIC, BIC, C_p & R^2_{adj} all are based on d = #parameters, and can not be used.]

1) Divide the data set into K equal parts ($K = 5, 10$)

2) Part $2, 3, \ldots, K$ is used to fit the model (RR & lasso) for a grid of λ-values $\rightarrow$ gives $\hat{\beta}$ for each λ.

Part 1 is used to calculate SSE for each $(\hat{\beta}, \lambda)$ pair

$\Rightarrow$ Get SSE for the λ-grid: $SSE_1(\lambda)$

3) Now part $(1, 3, 4, .., K)$ is used to fit the model for the λ-grid $\rightarrow$ $(\hat{\beta}, \lambda)$ pairs.

Part 2 is used to calculate SSE: $SSE_2(\lambda)$

4) Repeat: do the same for part $3, 4, .., K$ left out.

$$SSE_3(\lambda), \ \dots, \ SSE_K(\lambda)$$

5) Sum the SSE for each part $1, .., K$ left out for each value of λ. Plot.

Choose the λ with the minimum $\sum_{l=1}^{K} \frac{SSE_l(\lambda)}{K}$.

Credit: choose λ ridge

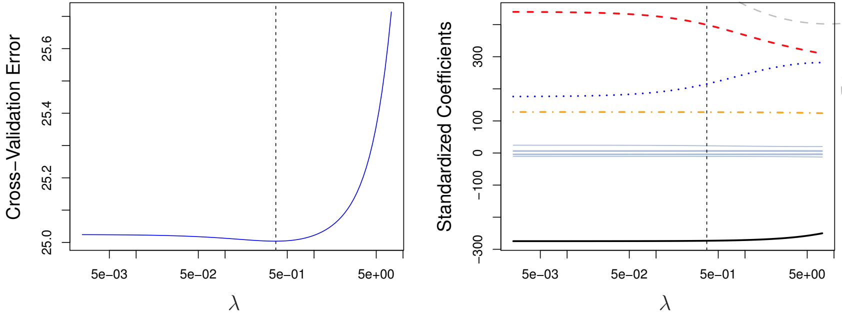


Figure 6.12 from An Introduction to Statistical Learning (2013)

Simulated data (B): choose λ lasso

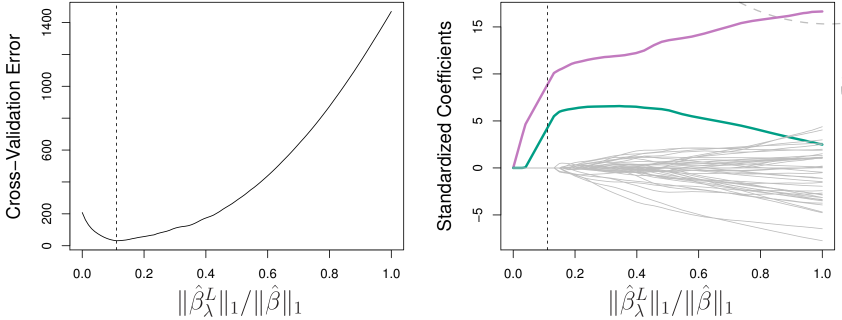


Figure 6.13 from An Introduction to Statistical Learning (2013)

R: acid rain with ridge & lasso

y = pH in lake

X1 SO_4, X2 NO_3, X3 Ca, X4 Al, X5 org, X6 area, X7 Telen Trondel. $p = 7$

$n = 26$

Previously: best subset & Cp gave model with x1, x2 and X3 to be the best.

The one-sd-rule: Due to the principle of parsimony ISLR recommends first finding $\min_{\lambda} MSE(\lambda) = \lambda_{min}$, then choosing

$$\sum_{k=1}^{K} \frac{SSE_k(\lambda)}{K}$$

$\lambda_{min} + sd(\lambda_{min}) = \lambda_{selected}$, where $sd(\lambda_{min})$ is found from the cross validation.

This is the default choice in R: cv.glmnet.

Ridge : all 7 - of cause

Lasso : x1, x2, x3, x4 with one-sd-rule.

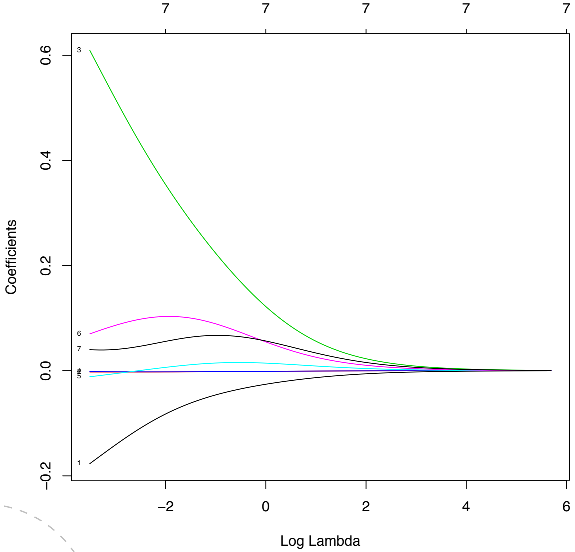
Acid rain

```
ds=read.table("http://www.math.ntnu.no/~mettela/TMA4267/
Data/acidrain.txt",header=TRUE)

# 2. Shrinkage with Ridge and lasso
#First we will fit a ridge-regression model.
This is achieved by calling ‘glmnet‘ with ‘alpha=0‘
There is also a ‘cv.glmnet‘ function which will do
the cross-validation for us.

library(glmnet)
x=model.matrix(y~.-1,data=ds)
y=ds$y
```

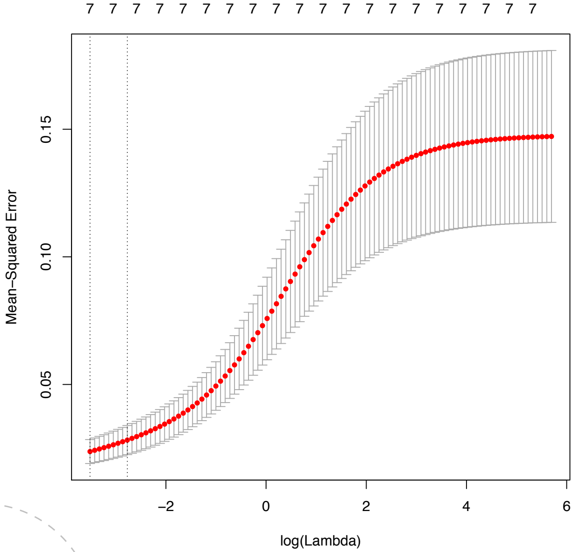
Acid rain: ridge

```
fit.ridge=glmnet(x,y,alpha=0)

plot(fit.ridge,xvar="lambda",label=TRUE)

cv.ridge=cv.glmnet(x,y,alpha=0)
cv.ridge$lambda.min
[1] 0.02976545
which.min(cv.ridge$cvm) #length 100, range: 297-0.0297
[1] 100
cv.ridge$lambda.1se
# use 1sd error rule default, unless foldid=FALSE
[1] 0.06265342
plot(cv.ridge)
coef(cv.ridge)
8 x 1 sparse Matrix of class "dgCMatrix"
                       1
(Intercept)  5.590860491
x1          -0.125713299
x2          -0.002482710
x3           0.476292879
x4          -0.002107013
x5          -0.002678078
x6           0.092472787
x7           0.042759054
```

Acid: choose λ ridge



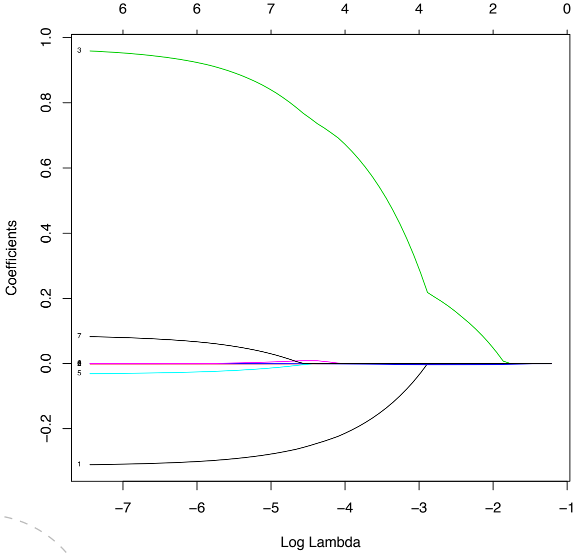
Acid: choose λ ridge



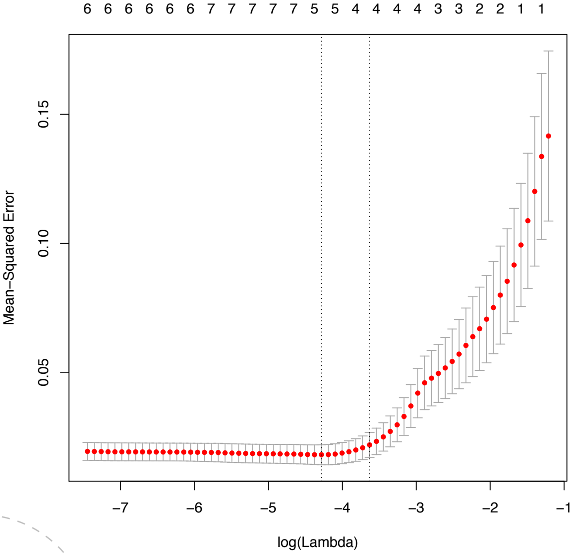
Acid rain: lasso

```
#Now we fit a lasso model; for this we use the default ‘alpha=1‘
fit.lasso=glmnet(x,y)
plot(fit.lasso,xvar="lambda",label=TRUE)
# lambda from 0.297 to 0.0005843, 68 values
cv.lasso=cv.glmnet(x,y)
which.min(cv.lasso$cvm) #50
plot(cv.lasso)
coef(cv.lasso)
8 x 1 sparse Matrix of class "dgCMatrix"
                      1
(Intercept)  5.653750217
x1          -0.204444353
x2          -0.001333015
x3           0.651533857
x4          -0.001845269
x5           .
x6           .
x7           .
```

Acid: choose λ lasso



Acid: choose λ lasso



Dimension reduction method [6.3, 10.2]

1) Transform the predictors
2) fit LS to transformed variables

1) $Z_1, Z_2, \dots, Z_M$ $\quad M<p$
linear combinations of original variables

$$Z_m = \sum_{j=1}^{p} \phi_{jm} X_j$$

where $\phi_{1m}, \phi_{2m}, \dots, \phi_{pm}$, $m=1,\dots,M$, are constants.
↖ how to find these?

2) the MLR model then becomes:

$$(*)\ Y_i = \theta_0 + \sum_{m=1}^{M} \theta_m \cdot Z_{im} + \varepsilon_i, \quad i=1,\dots,n$$

and is fitted using LS.
$(\theta_0, \theta_1, \dots, \theta_M)$ are the regression coefficients.
There are $M<p$ regression coefficients, which is the reason for "dimension reduction".

Comparing (*) to a MLR in the orignal variables

$$\sum_{m=1}^{M} \theta_m z_{im} = \sum_{m=1}^{M} \theta_m \sum_{j=1}^{p} \phi_{jm} \cdot x_{ij}$$

$$= \sum_{j=1}^{p} \underbrace{\sum_{m=1}^{M} \theta_m \phi_{jm}}_{\beta_j} x_{ij} = \sum_{j=1}^{p} \beta_j x_{ij}$$

Let $\sum_{i=1}^{M} \theta_m \phi_{jm} = \beta_j$ (**)

m=1

We see that this is a MLR in the orignal variables, but with the constraint (**).

If $M=p$ and z_m's are chosen to be linearly independent, the (*) just an ordinary MLR in the orignal variables.

Next : look at 1) using the method of principal component analysis (PCA).