MONASH University

MONASH BUSINESS SCHOOL

ETC3250: Regularisation and Shrinkage

Week 9, class 2

Professor Di Cook, Econometrics and Business Statistics

GROUP OF EIGHT AUSTRALIA

AACSB ACCREDITED EFMD EQUIS ACCREDITED ASSOCIATION OF MBAS ACCREDITED

Problem

- When the number of variables (p) is large, estimating a model is problematic.
- Particularly, when it is larger than the sample size ($p >> n$), the variance of an estimate could be ∞.
- Constraining, or shrinking the estimates, can substantially decrease the variance, while minimally affecting the bias.

Simple solutions

- Subset selection: Fit models to best subset
- Dimension reduction: Use combinations of variables, e.g. PCs, and feed these into your model

Shrinkage using Ridge Regression

- Modified least squares

$$\sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p} \beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p} \beta_j^2$$

- where λ is a tuning parameter
- Minimizing this quantity trades off error with small β's, at least forcing some of them to be small

Shrinkage using Lasso

- More recent alternative to ridge regression

$$\sum_{i=1}^{n}(y_i - b_0 - \sum_{j=1}^{p}\beta_j x_{ij})^2 + \lambda \sum_{j=1}^{p}|\beta_j|$$

- the change using an l_1 error, really forces some of the coefficients to be 0.

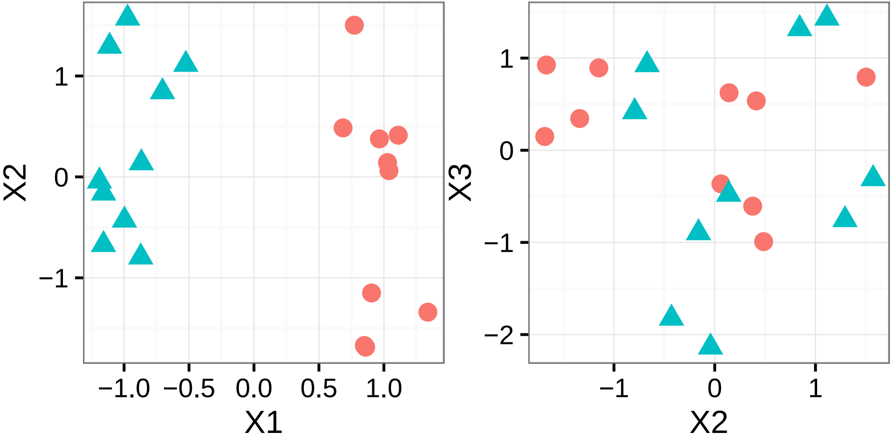
Simulation example

```
x<-matrix(rnorm(20*100),ncol=100)
x[1:10,1]<-x[1:10,1]+5
x<-scale(x)
x<-data.frame(x, cl=c(rep("A",10),rep("B",10)))
library(ggplot2)
qplot(X1,X2,data=x,colour=cl, size=I(3), shape=cl) +
  theme_bw() + theme(legend.position="None", aspect.ratio=1)
```

```
qplot(X2,X3,data=x,colour=cl, size=I(3), shape=cl) +
  theme_bw() + theme(legend.position="None", aspect.ratio=1)
```

```
# Generate test data
x.t<-matrix(rnorm(10*100),ncol=100)
x.t[1:5,1]<-x.t[1:5,1]+5
x.t<-scale(x.t)
x.t<-data.frame(x.t, cl=c(rep("A",5),rep("B",5)))
```

Simulation example



Fit LDA

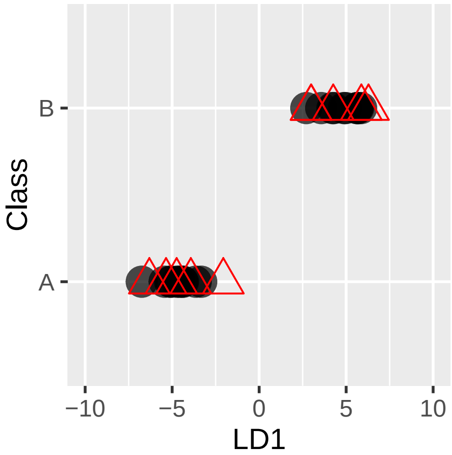
```
## Call:
## lda(cl ~ ., data = x[, c(1:2, 101)], prior = c(0.5, 0.5))
##
## Prior probabilities of groups:
##   A   B
## 0.5 0.5
##
## Group means:
##           X1         X2
## A  0.9555333 -0.2864578
## B -0.9555333  0.2864578
##
## Coefficients of linear discriminants:
##           LD1
## X1 -4.9308578
## X2  0.1098234
```

Predict LDA

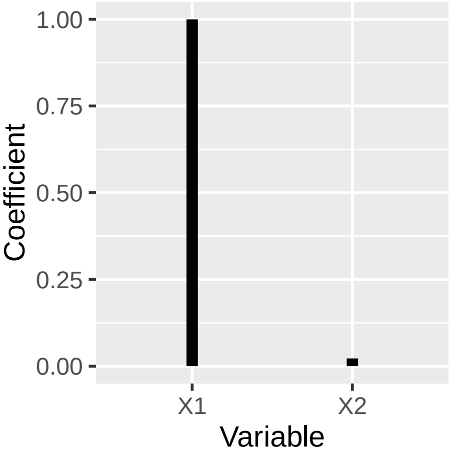
```
##
##       A  B
##    A 10  0
##    B  0 10
```

```
##
##      A B
##    A 5 0
##    B 0 5
```

Plot training data and test in discriminant space



Estimates



Increase the number of noise variables

- The next few slides repeat the results just shown for increasing number of variables
- None of the additional variables contribute to the separation between classes
- Additional variables are purely noise

Error

p=5

```
##
##       A  B
##    A 10  0
##    B  0 10
```

```
##
##      A B
##    A 5 0
##    B 0 5
```

Error

p=8

```
##
##        A  B
##     A 10  0
##     B  0 10
```

```
##
##       A B
##     A 5 0
##     B 0 5
```

Error

p=11

```
##
##      A  B
##   A 10  0
##   B  0 10

##
##     A B
##   A 5 0
##   B 0 5
```

Error

p=12

```
##
##       A  B
##    A 10  0
##    B  0 10

##
##      A B
##    A 5 0
##    B 0 5
```

Error

p=13

```
##
##       A  B
##    A 10  0
##    B  0 10

##
##      A B
##    A 4 1
##    B 0 5
```

Error

p=14

```
##
##       A  B
##    A 10  0
##    B  0 10
```

```
##
##      A B
##    A 4 1
##    B 0 5
```

Error

p=15

```
##
##       A  B
##    A 10  0
##    B  0 10
```

```
##
##      A B
##    A 4 1
##    B 1 4
```

Error

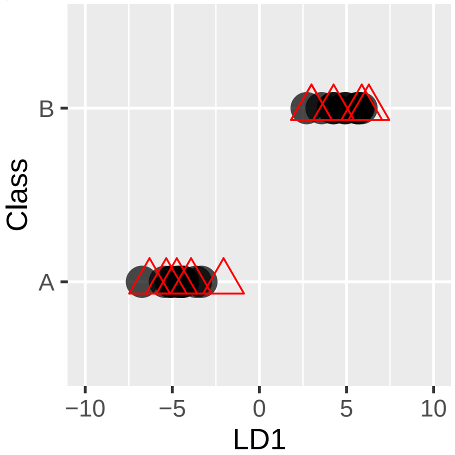
p=16

```
##
##      A  B
##   A 10  0
##   B  0 10
```

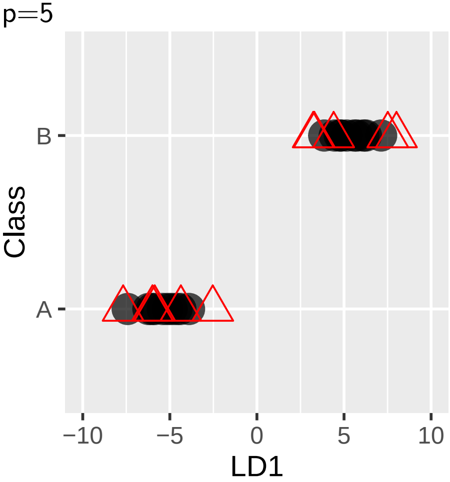
```
##
##     A B
##   A 4 1
##   B 0 5
```

Plot training data and test in discriminant space

p=2



Plot training data and test in discriminant space

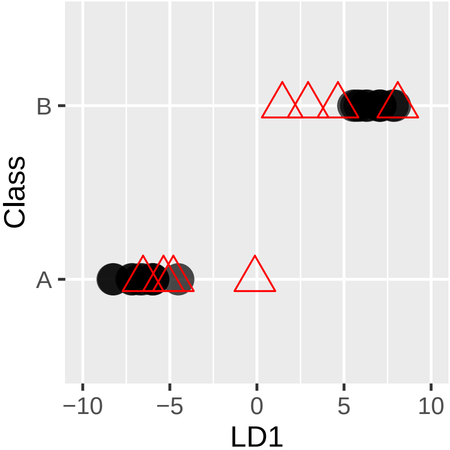


Plot training data and test in discriminant space

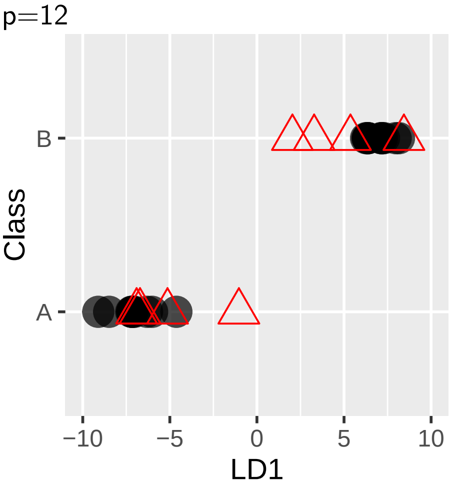
p=8

Plot training data and test in discriminant space

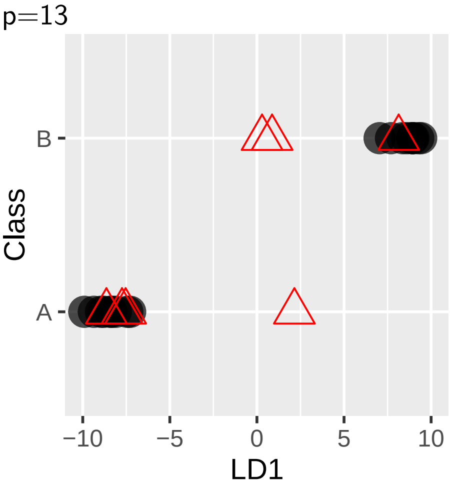
p=11



Plot training data and test in discriminant space

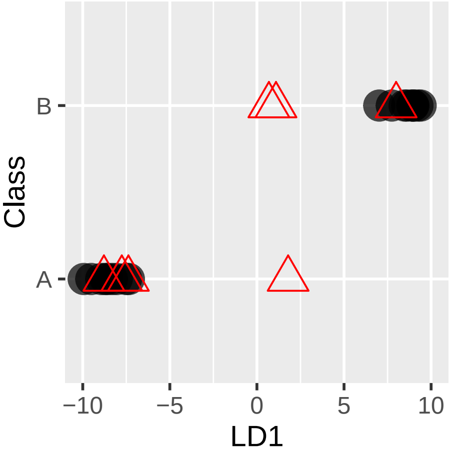


Plot training data and test in discriminant space



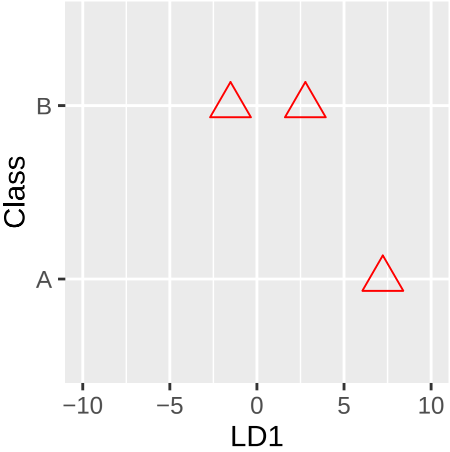
Plot training data and test in discriminant space

p=14

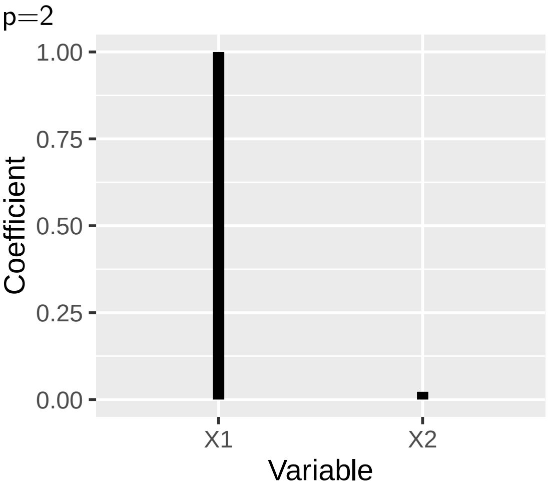


Plot training data and test in discriminant space

p=15

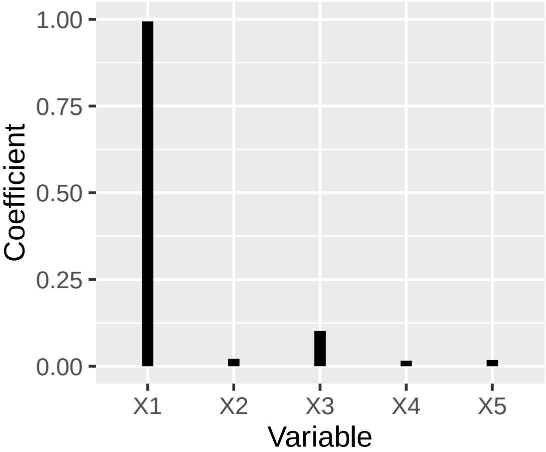


Estimates



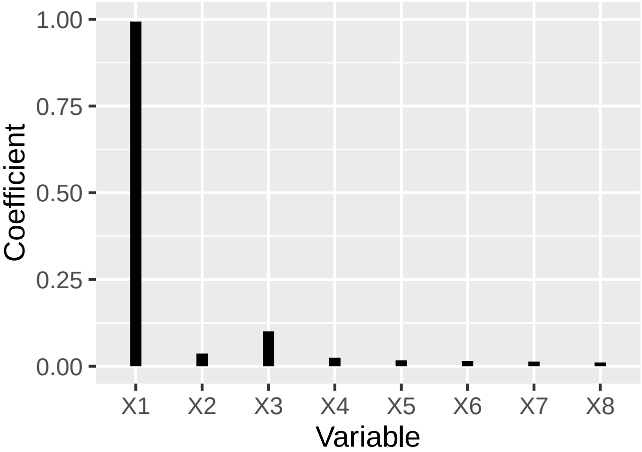
Estimates

p=5



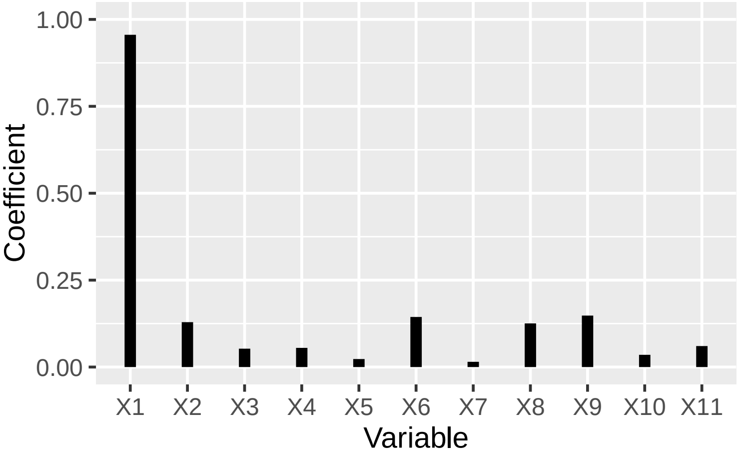
Estimates

p=8



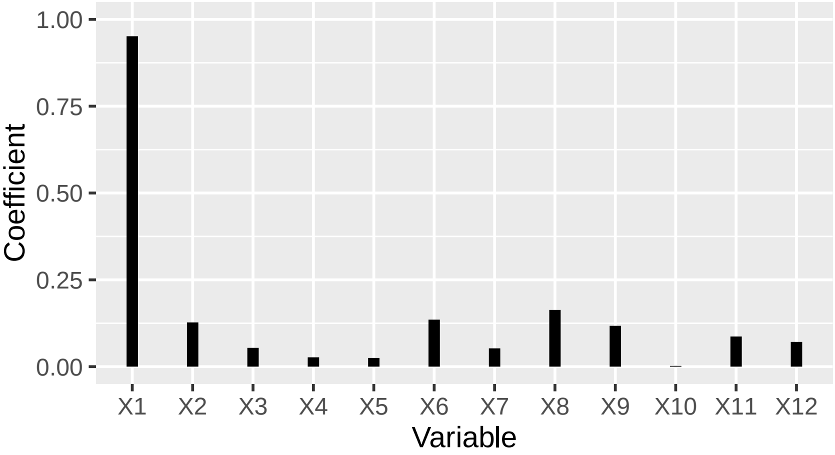
Estimates

p=11



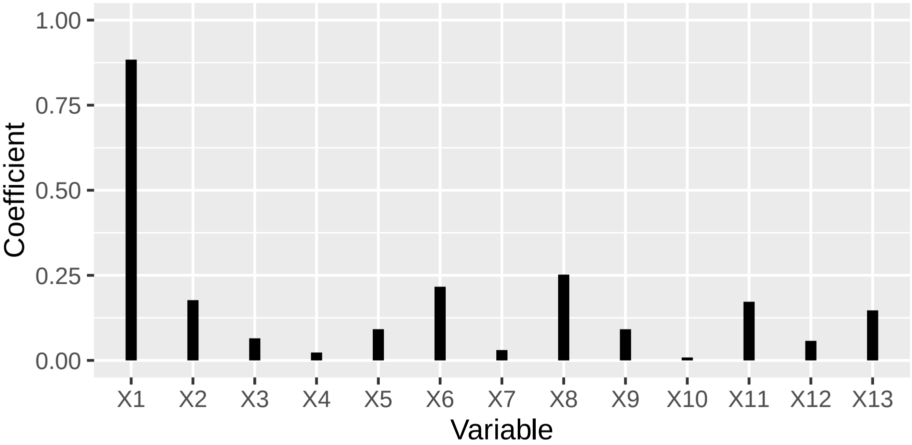
Estimates

p=12

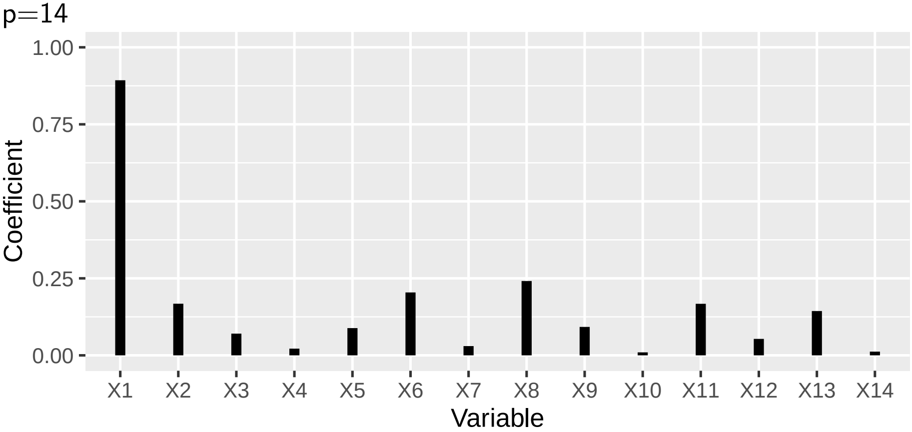


Estimates

p=13



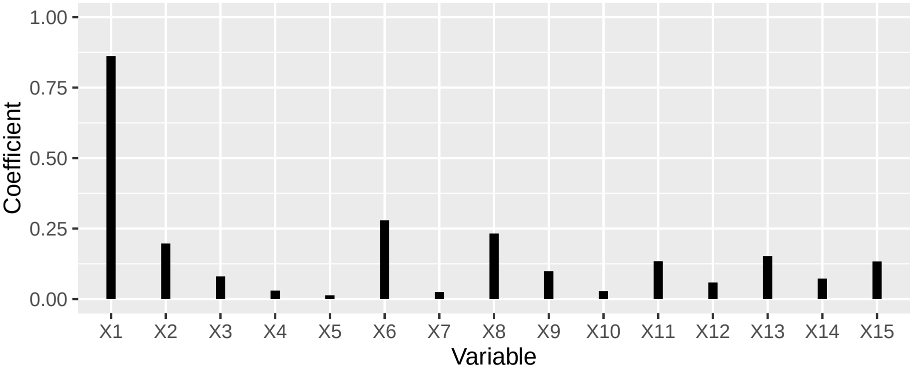
Estimates



p=14
Coefficient
1.00
0.75
0.50
0.25
0.00
X1
X2
X3
X4
X5
X6
X7
X8
X9
X10
X11
X12
X13
X14
Variable

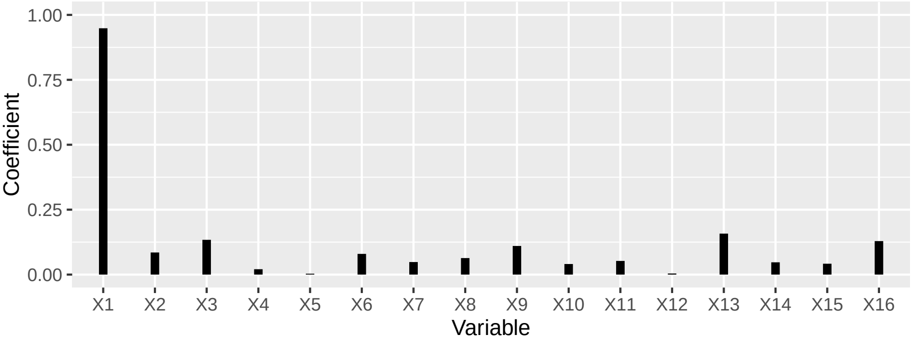
Estimates

p=15



Estimates

p=16



Penalized LDA

- Reference: `http://faculty.washington.edu/dwitten/Papers/JRSSBPenLDA.pdf`
- LDA does an eigendecomposition of $W^{-1}B$, which creates an estimation problem if $<< p$
- Instead compute regularised versions of W, B

$$maximize_{\beta_k} \beta_k^T \hat{\Sigma}_b^k \beta_k - \lambda_k \sum_{j=1}^{p} |\hat{\sigma_k} \beta_{kj}|$$

subject to

$$\beta_k^T \hat{\Sigma}_w \beta_k \leq 1,$$

where $\hat{\Sigma}_b^k = \frac{1}{n} X^T Y (Y^T Y)^{-1/2} P_k^{\perp} (Y^T Y)^{-1/2} Y^T X$, $\hat{\Sigma}_w$ is a positive definite estimate of Σ_w.

Penalized LDA

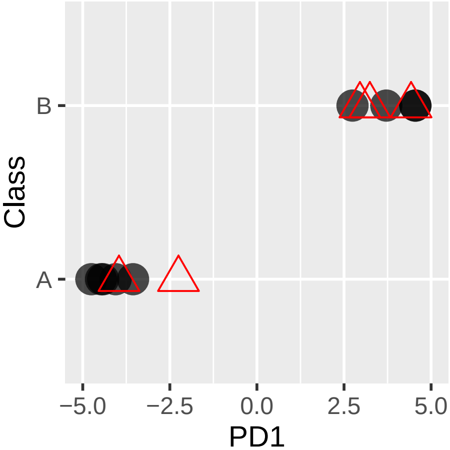
```
library(penalizedLDA)
cv.out<-PenalizedLDA.cv(as.matrix(x[,c(1:15)]), as.numeric(x$c
```

```
## Fold  1
## 12345Fold  2
## 12345Fold  3
## 12345Fold  4
## 12345Fold  5
## 12345Fold  6
## 12345
```

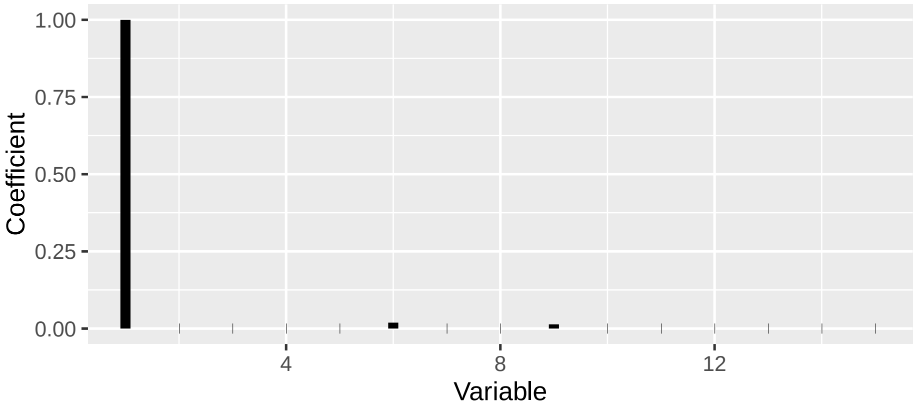
```
x.pda<-PenalizedLDA(as.matrix(x[,c(1:15)]), xte=as.matrix(x.t
table(x.t$cl, x.pda$ypred)
```

```
##
##      1 2
##    A 5 0
```

Plot training and test



Estimates



Share and share alike