# Week 4

Jinhong Du

## 目录

# glmnet

## train by cv

```
library(glmnet)
```

```
## Loading required package: Matrix

## Loading required package: foreach

## Loaded glmnet 2.0-13
```

```
PET <- read.table('PET.txt', header = TRUE)
index <- sample(1:nrow(PET), 20)
PET.train <- PET[index,]
PET.test <- PET[-index,]
pet.rr <- cv.glmnet(x = as.matrix(PET.train[,1:268]),
                    y = PET.train[,269],
                    nfolds = 5,
                    alpha = 0,
                    type.measure = "deviance")
```

```
pet.lasso <- cv.glmnet(x = as.matrix(PET.train[,1:268]),
                    y = PET.train[,269],
                    nfolds = 5,
                    alpha = 1,
                    type.measure = "deviance")

# plot.cv.glmnet(pet.rr)
# coef.cv.glmnet(pet.rr)
# plot.cv.glmnet(pet.lasso)
# coef.cv.glmnet(pet.lasso)
```

### predict

```
pre.rr <- predict.cv.glmnet(pet.rr,
                            newx = as.matrix(PET.test[,1:268]))
sum((pre.rr - PET.test[,269])^2)/length(pre.rr)
```

```
## [1] 18.23962
```

```
pre.lasso <- predict.cv.glmnet(pet.lasso,
                               newx = as.matrix(PET.test[,1:268]))
sum((pre.lasso - PET.test[,269])^2)/length(pre.lasso)
```

```
## [1] 0.5298461
```

### train and plot

```
pet.lasso <- glmnet(x = as.matrix(PET.train[,1:268]),
                    y = PET.train[,269],
                    alpha = 1,
                    family = "gaussian")
# plot(pet.lasso, xvar = "norm")
```

## lars

### train

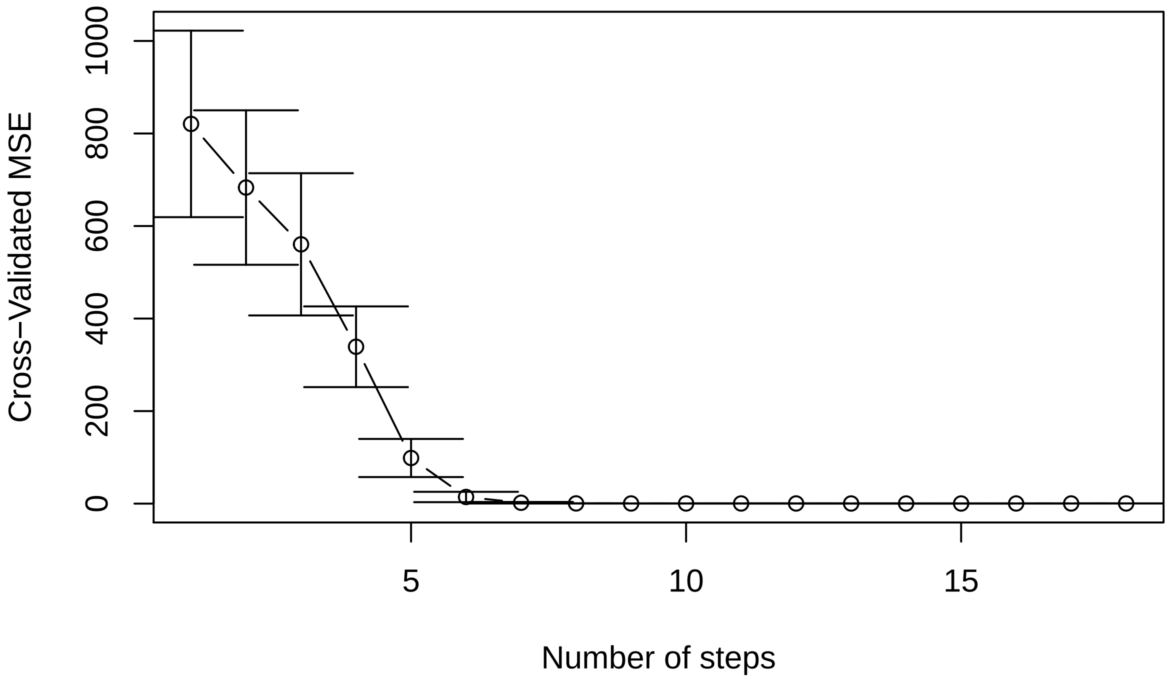
```
library(lars)
```

```
## Loaded lars 1.2
```

```
pet.lar <- lars(x = as.matrix(PET.train[,1:268]),
                y = PET.train[,269],
                type = "lar")
# pet.lar
# plot(pet.lar, breaks = FALSE)
```

### train by cv

```
pet.lar.cv <- cv.lars(x = as.matrix(PET.train[,1:268]),
                      y = PET.train[,269],
                      type = "lar")
```



### predict

```
pre.lar <- predict.lars(pet.lar,
                        type = "fit",
                        s = 5,
                        mode = "step",
                        newx = as.matrix(PET.test[,1:268]))
sum((pre.lar$fit - PET.test[,269])^2)/length(pre.lar$fit)
```

```
## [1] 373.827
```