STAT 8330 FALL 2015 ASSIGNMENT 3

Peng Shao

September 19, 2015

Exercises 6.9.

Solution.

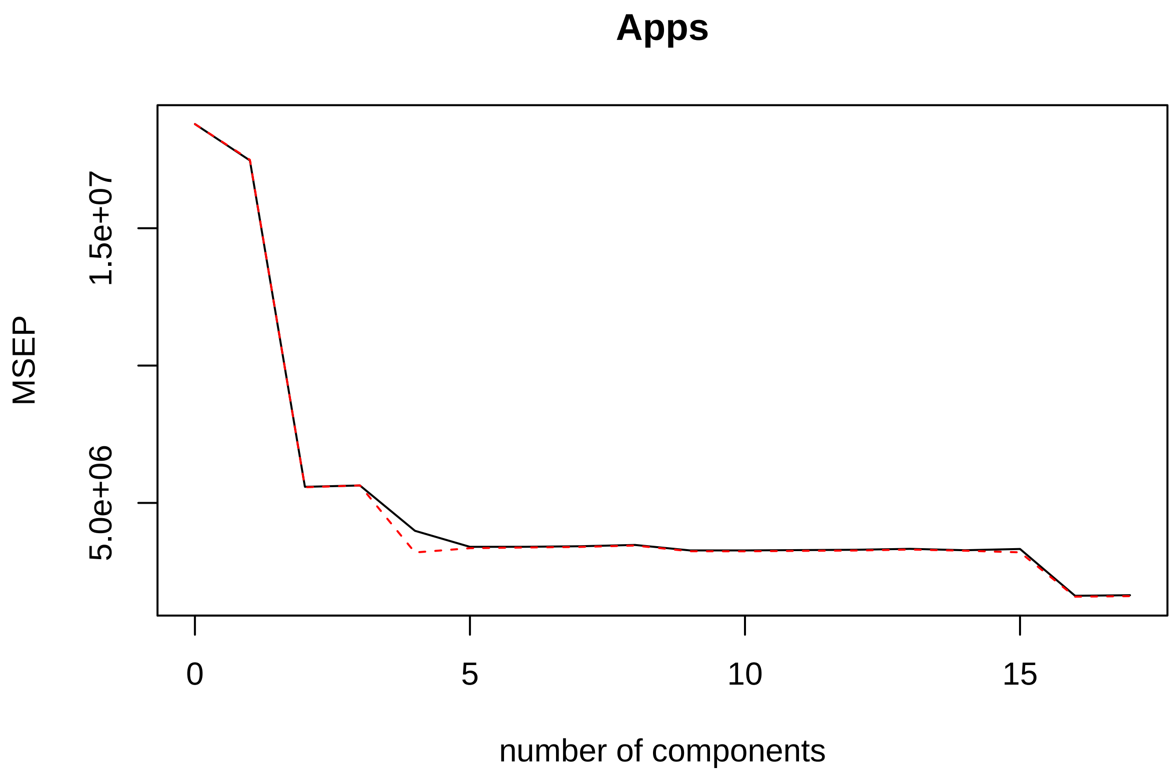
```
## [1] 777  18
```

```
## [1] 1108531
```

```
## [1] 1054527
```

```
## [1] 1039503
```

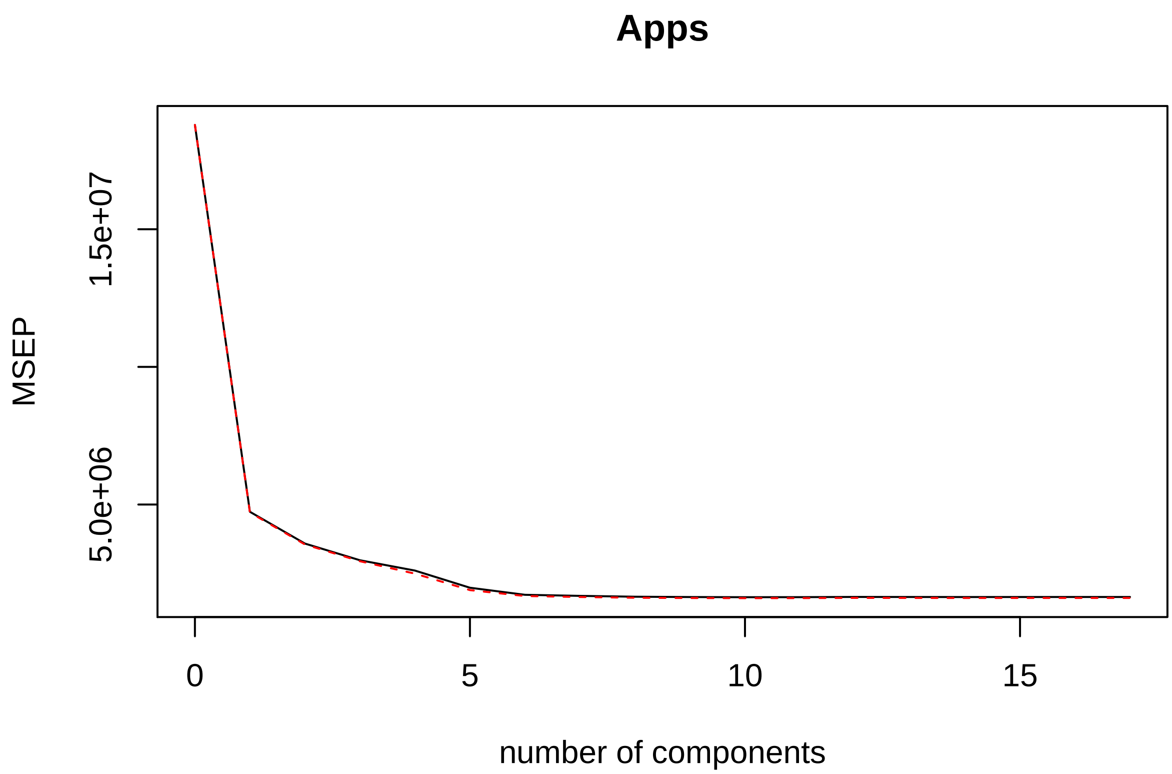
```
##   (Intercept)        Accept        Enroll     Top10perc   P.Undergrad
## -1.032331e+03  1.427185e+00 -4.564142e-02  2.982154e+01  1.816370e-02
##      Outstate    Room.Board           PhD      Terminal     S.F.Ratio
## -6.984539e-02  1.045862e-01 -3.206556e+00 -1.710039e+00  7.985690e+00
##   perc.alumni        Expend     Grad.Rate
## -2.249056e+00  7.052280e-02  3.836355e+00
```




```
## [1] 1325616
```

```
## Data:    X dimension: 777 17
##  Y dimension: 777 1
## Fit method: svdpc
## Number of components considered: 16
```

```
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      31.670    57.30    64.30    69.90    75.39    80.38    83.99
## Apps    2.316    73.06    73.07    82.08    84.08    84.11    84.32
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.40    90.50     92.91     95.01     96.81      97.9     98.75
## Apps    85.18    85.88     86.06     86.06     86.10      86.1     86.13
##       15 comps  16 comps
## X        99.36     99.84
## Apps     90.32     92.52
```




```
## [1] 1279922
```

```
## Data:    X dimension: 777 17
##  Y dimension: 777 1
## Fit method: svdpc
## Number of components considered: 10
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X      31.670    57.30    64.30    69.90    75.39    80.38    83.99
## Apps    2.316    73.06    73.07    82.08    84.08    84.11    84.32
##       8 comps  9 comps  10 comps
## X       87.40    90.50     92.91
## Apps    85.18    85.88     86.06
```

Appendix

```r
rm(list = ls())
library(ISLR)
library(ggplot2)
library(boot)
library(glmnet)
library(leaps)
library(pls)
library(MASS)
library(knitr)

# 9
set.seed(1)
attach(College)
# (a)
set.seed(1)
dim(College)
train.ind <- sample(1:nrow(College), nrow(College)/2)
test.ind <- -train.ind
College.train <- College[train.ind, ]
College.test <- College[test.ind, ]

# (b).
set.seed(1)
lm.fit_1 <- lm(Apps ~ ., data = College.train)
lm.pred <- predict(lm.fit_1, newdata = College.test)
mean((lm.pred - College.test$Apps)^2)

# (c).
set.seed(1)
ridge.cv.out <- cv.glmnet(x = model.matrix(Apps ~ .,College.train)[,-2],
                          y = College.train$Apps,
                          alpha = 0)
ridge.bestlam <- ridge.cv.out$lambda.min
ridge.mod <- glmnet(x = model.matrix(Apps ~ .,College.train)[,-2],
                    y = College.train$Apps,
                    alpha = 0,
                    lambda = ridge.bestlam)
ridge.pred <- predict(ridge.mod,
                      s = ridge.bestlam,
                      newx = model.matrix(Apps ~ ., College.test)[,-2])
mean((ridge.pred - College.test$Apps)^2)

# (d).
set.seed(1)
lasso.cv.out <- cv.glmnet(x = model.matrix(Apps ~ ., College.train)[,-2],
                          y = College.train$Apps,
                          alpha = 1)
lasso.bestlam <- lasso.cv.out$lambda.min
lasso.mod <- glmnet(x = model.matrix(Apps ~ ., College.train)[,-2],
                    y = College.train$Apps,
                    alpha = 1,
                    lambda = lasso.bestlam)
lasso.pred <- predict(lasso.mod,
```

```
                      s = lasso.bestlam,
                      newx = model.matrix(Apps ~ ., College.test)[,-2])
mean((lasso.pred - College.test$Apps)^2)
lasso.mod.full <- glmnet(x = model.matrix(Apps ~ ., College)[,-2],
                         y = College$Apps,
                         alpha = 1,
                         lambda = lasso.bestlam)
lasso.coef <- predict(lasso.mod.full,
                      type="coefficients",
                      s=lasso.bestlam)[1:18,]
lasso.coef <- lasso.coef[lasso.coef!=0]
lasso.coef


# (e)
set.seed(1)
pcr.fit <- pcr(Apps ~ .,
               data = College.train,
               scale = TRUE,
               validation = "CV")
validationplot(pcr.fit,
               val.type = "MSEP")
pcr.pred <- predict(pcr.fit,
                    model.matrix(Apps ~ ., College.test)[,-2],
                    ncomp = 16)
mean((pcr.pred - College.test$Apps)^2)
pcr.fit.full <- pcr(Apps ~ .,
                    data = College,
                    scale = TRUE,
                    ncomp = 16)
summary(pcr.fit.full)


# (f)
set.seed(1)
plsr.fit <- plsr(Apps ~ .,
                 data = College.train,
                 scale = TRUE,
                 validation = "CV")
validationplot(plsr.fit,
               val.type = "MSEP")
plsr.pred <- predict(plsr.fit,
                     model.matrix(Apps ~ ., College.test)[,-2],
                     ncomp = 10)
mean((plsr.pred - College.test$Apps)^2)
plsr.fit.full <- pcr(Apps ~ .,
                     data = College,
                     scale = TRUE,
                     ncomp = 10)
summary(plsr.fit.full)
detach(College)

# 2
```

```
attach(Boston)
```