Chapter 6 HW

Nehemya McCarter-Ribakoff

13 April 2017

Conceptual Questions

Exercise 1: We perform best subset, forward stepwise, and backward stepwise selection on a single data set. For each approach, we obtain p + 1 models, containing 0, 1, 2,…,p predictors. Explain your answers:

(a) Which of the three models with k predictors has the smallest training RSS?

A naive best subset selection approach will select the model with the smallest training RSS. Since RSS decrease monotonically w.r.t the number of predictors, this model will be the one with p + 1 predictors. This is why it is called naive, and the model's low training RSS will not hold up against test data. Best subset selection is also computationally expensive.

(b) Which of the three models with k predictors has the smallest test RSS?

This is the ultimate goal, so naturally, there is no straightforward answer to which of these models will have the smallest test RSS. The best we can do is estimate the test error directly or indirectly.

(c) True or False:

i. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by forward stepwise selection.

True

ii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)-variable model identified by backward stepwise selection.

True

iii. The predictors in the k-variable model identified by backward stepwise are a subset of the predictors in the (k + 1)-variable model identified by forward stepwise selection.

True

iv. The predictors in the k-variable model identified by forward stepwise are a subset of the predictors in the (k+1)-variable model identified by backward stepwise selection.

False. Backward stepwise selection is not possible when n < p because the whole model cannot be fit. Forward stepwise selection does not have this problem.

v. The predictors in the k-variable model identified by best subset are a subset of the predictors in the (k + 1)-variable model identified by best subset selection.

True

Exercise 2: For parts (a) through (c), indicate which of i. through iv. is correct. Justify your answer.

(a) The lasso, relative to least squares, is:

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Incorrect. Like ridge regression, the lasso decreases flexibility as λ increases. The bias-variance tradeoff described here is correct: the lasso will take a small increast in bias for a larger decrease in variance.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Incorrect. The lasso has less flexibility, and as mentioned above, the bias-variance tradeoff is reversed from the one mentioned here.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Correct. The lasso causes a decrease in flexibility, which means bias to the model will be somewhat higher, but this is acceptable if the bias increase is less than the consequent decrease in variance. Additionally, the lasso may drop some predictors entirely, resulting in a simpler model.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Incorrect. The lasso aims to decrease variance for a slight increase in bias. This has the relationship mixed up.

(b) Repeat (a) for ridge regression relative to least squares.

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Incorrect. Ridge regression uses a multiplicative term λ that causes a decrease in the model's flexibility as it increases. This gives the model a slight increase in bias for a much larger decrease in variance.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Incorrect, once again, ridge regression results in *less* flexibility. Moreover, less flexibility translates into less variance, since both refer to far a model will deviate from its pattern (e.g., linear) in order to more closely fit any given set of training data. Since variance is a measure of how much a model's shape changes with a given training set, ridge regression's decrease in flexibility means a decrease in variance.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Correct. Ridge regression causes a decrease in flexibility, which means bias to the model will be somewhat higher, but this is acceptable if the bias increase is less than the consequent decrease in variance.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Incorrect. Ridge regression aims to decrease variance for a slight increase in bias. This has the relationship mixed up.

(c) Repeat (a) for non-linear methods relative to least squares.

i. More flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Incorrect. A non-linear method will certainly have more flexibility than least squares, but this added flexibility will result in an increase in variance and a derease in bias.

ii. More flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Correct. As stated above, non-linear methods are not as rigid as least squares. Their flexibility will provide a lower bias, but higher variance. If the increase in variance is less than the decrease in bias, then it is a better approach than least squares.

iii. Less flexible and hence will give improved prediction accuracy when its increase in bias is less than its decrease in variance.

Incorrect. Non-linear models are more flexible.

iv. Less flexible and hence will give improved prediction accuracy when its increase in variance is less than its decrease in bias.

Incorrect. Non-linear models are more flexible, though the bias-variance tradeoff described here is an accurate for non-linear models.

Applied Questions

Exercise 9: In this exercise, we will predict the number of applications received using the other variables in the College data set.

(a) Split the data set into a training set and a test set.

```
library(ISLR)
library(caTools)
data(College)

set.seed(1)
apps.true = sample.split(College$Apps, 2/3)
set.train = subset(College, apps.true == TRUE)
set.test = subset(College, apps.true == FALSE)
```

(b) Fit a linear model using least squares on the training set, and report the test error obtained.

```
College.full = na.omit(College)
lm.fit = lm(Apps ~., set.train)
prediction = predict(lm.fit, set.test)
mse = mean((set.test$Apps - prediction)^2)
mse
```

```
## [1] 1689971
```

MSE: 1689971

(c) Fit a ridge regression model on the training set, with λ chosen by cross-validation. Report the test error obtained.

```
library(glmnet)
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loaded glmnet 2.0-5
```

```
College = na.omit(College)
X.train = model.matrix(Apps~., set.train)[,-1]
X.test = model.matrix(Apps~., set.test)[,-1]
Y.train = set.train$Apps
grid = 10 ^ seq(10, -2, length=100)

mod.ridge = cv.glmnet(X.train, Y.train, alpha = 0, lambda=grid)
best.lambda = mod.ridge$lambda.1se

ridge.pred = predict(mod.ridge, newx=X.test, s=best.lambda)
ridge.mse = mean((set.test$Apps - ridge.pred)^2)
ridge.mse
```

```
## [1] 3121469
```

MSE: 3121469

(d) Fit a lasso model on the training set, with λ chosen by crossvalidation. Report the test error obtained, along with the number of non-zero coefficient estimates.

```
X.train = model.matrix(Apps~., set.train)[,-1]
X.test = model.matrix(Apps~., set.test)[,-1]
Y.train = set.train$Apps
grid = 10 ^ seq(10, -2, length=100)

mod.ridge = cv.glmnet(X.train, Y.train, alpha = 0, lambda=grid)
best.lambda = mod.ridge$lambda.1se
ridge.pred = predict(mod.ridge, newx=X.test, s=best.lambda)
ridge.mse = mean((set.test$Apps - ridge.pred)^2)
ridge.mse
```

```
## [1] 3334977
```

MSE: 3121469

There appear to be no coefficients brought to zero.

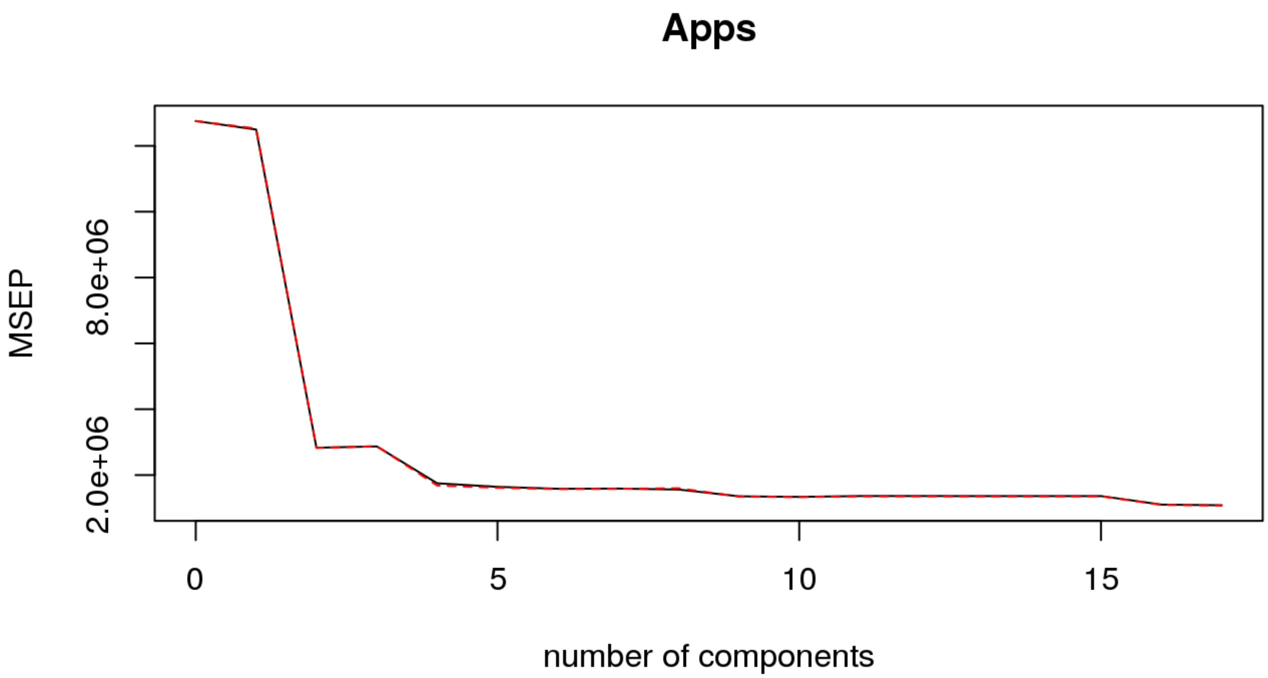
(e) Fit a PCR model on the training set, with M chosen by cross validation. Report the test error obtained, along with the value of M selected by cross-validation.

```
library(pls)
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```

```
pcr.fit = pcr(Apps~., data=set.train, scale=TRUE, validation ="CV")
validationplot(pcr.fit,val.type="MSEP")
```




```
pcr.pred = predict(pcr.fit, set.test, ncomp=9)
pcr.mse = mean((pcr.pred - set.test$Apps)^2)
summary(pcr.fit)
```

```
## Data:    X dimension: 518 17
##  Y dimension: 518 1
## Fit method: svdpc
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            3571     3535     1682     1694     1323     1281     1258
## adjCV         3571     3540     1680     1697     1297     1271     1255
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1261     1249     1164      1156      1167      1167      1166
## adjCV     1258     1265     1161      1154      1164      1165      1164
##        14 comps  15 comps  16 comps  17 comps
## CV         1166      1167      1050      1040
## adjCV      1164      1164      1046      1037
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       31.63    57.88    65.02    70.75    76.08    80.94    84.63
## Apps     3.19    78.23    78.42    88.07    88.08    88.23    88.32
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       87.82    90.88     93.28     95.40     97.13     98.19     98.95
## Apps    88.34    90.00     90.11     90.11     90.15     90.19     90.23
##       15 comps  16 comps  17 comps
## X        99.46     99.82    100.00
## Apps     90.23     92.25     92.58
```

```
pcr.mse
```

```
## [1] 3896197
```

Test MSE: 3896197, higher than our previous models M chosen by CV: 17 components

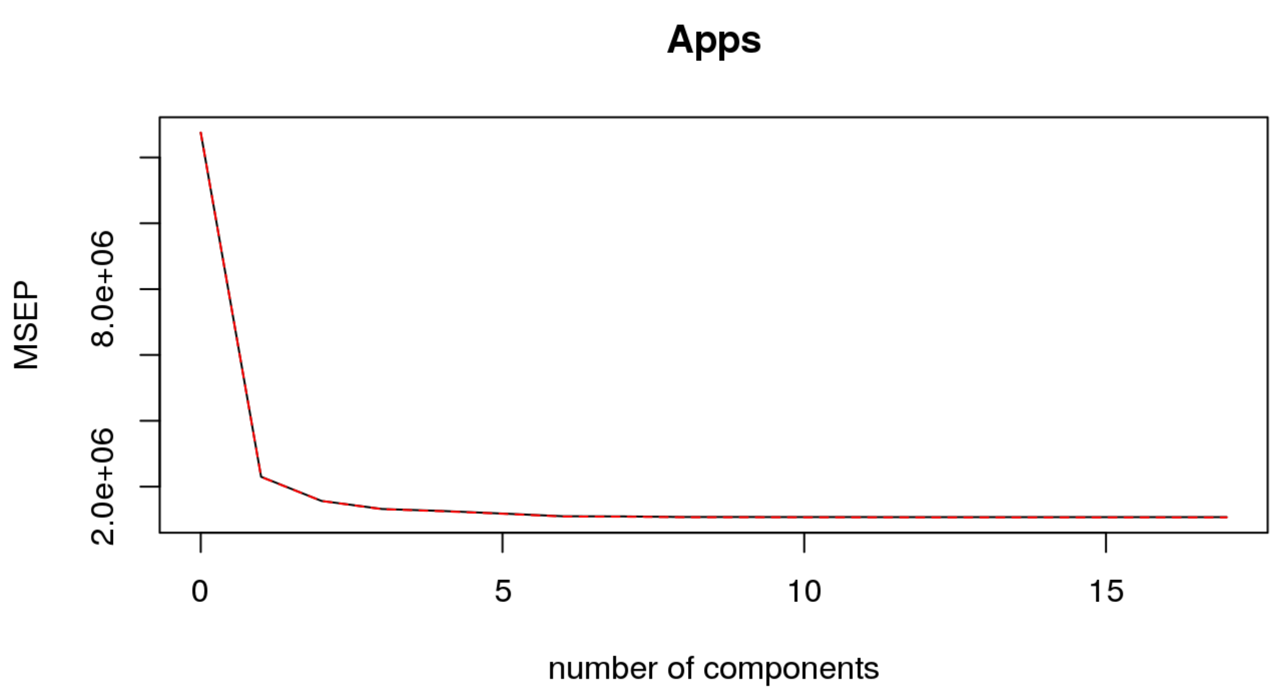
(f) Fit a PLS model on the training set, with M chosen by cross validation.

Report the test error obtained, along with the value of M selected by cross-validation.

```
pls.fit = plsr(Apps~., data=set.train, scale=TRUE, validation ="CV")
summary(pls.fit)
```

```
## Data:    X dimension: 518 17
##  Y dimension: 518 1
## Fit method: kernelpls
## Number of components considered: 17
##
## VALIDATION: RMSEP
## Cross-validated using 10 random segments.
##        (Intercept)  1 comps  2 comps  3 comps  4 comps  5 comps  6 comps
## CV            3571     1516     1251     1149     1123     1087     1049
## adjCV         3571     1515     1251     1147     1119     1085     1045
##        7 comps  8 comps  9 comps  10 comps  11 comps  12 comps  13 comps
## CV        1047     1038     1038      1037      1037      1036      1036
## adjCV     1043     1035     1034      1034      1033      1033      1032
##        14 comps  15 comps  16 comps  17 comps
## CV         1036      1036      1036      1036
## adjCV      1032      1032      1032      1032
##
## TRAINING: % variance explained
##       1 comps  2 comps  3 comps  4 comps  5 comps  6 comps  7 comps
## X       26.43    41.54    63.38    66.72    71.13    74.24    77.67
## Apps    82.49    88.29    90.30    91.08    91.75    92.43    92.50
##       8 comps  9 comps  10 comps  11 comps  12 comps  13 comps  14 comps
## X       80.60    82.89     85.41     87.93     91.41     93.25     94.58
## Apps    92.53    92.55     92.56     92.57     92.57     92.58     92.58
##       15 comps  16 comps  17 comps
## X        97.30     99.03    100.00
## Apps     92.58     92.58     92.58
```

```
validationplot(pls.fit,val.type="MSEP")
```



```
pls.pred = predict(pls.fit, set.test, ncomp=9)
pls.mse = mean((pls.pred - set.test$Apps)^2)
pls.mse
```

```
## [1] 1715153
```

Test error: 1715153, much lower than our previous estimate M chosen by CV: 9 components (9 through 17 are all equal)

(g) Comment on the results obtained. How accurately can we predict the number of college applications received? Is there much difference among the test errors resulting from these five approaches?

I am not sure how to interpret these test error estimates because they seem incredibly large. Our PLS and linear models perform with the lowest test errors, but they are so far from 0 I am not sure what to compare them to, or how to gague their prediction accuracy.

Teamwork report

Team member	Conceptual	Applied	Contribution %
Nehemya	Yes	Yes	100%
Total			100%