isye6501_hw6

Chengqi

2019/9/27

Question 9.1

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression model using the first few principal components. Specify your new model in terms of the original variables (not the principal components), and compare its quality to that of your solution to Question 8.2. You can use the R function prcomp for PCA. (Note that to first scale the data, you can include scale. = TRUE to scale as part of the PCA function. Don't forget that, to make a prediction for the new city, you'll need to unscale the coefficients (i.e., do the scaling calculation in reverse)!)

First we have to load data.

```
#load data
data <- read.table('C:\\Users\\huangchengqi\\Desktop\\MS SCE\\19Fall\\I
SYE6501\\hw6\\data 9.1\\uscrime.txt',header=TRUE)
str(data)
```

```
## 'data.frame':    47 obs. of  16 variables:
##  $ M     : num  15.1 14.3 14.2 13.6 14.1 12.1 12.7 13.1 15.7 14 ...
##  $ So    : int  1 0 1 0 0 0 1 1 1 0 ...
##  $ Ed    : num  9.1 11.3 8.9 12.1 12.1 11 11.1 10.9 9 11.8 ...
##  $ Po1   : num  5.8 10.3 4.5 14.9 10.9 11.8 8.2 11.5 6.5 7.1 ...
##  $ Po2   : num  5.6 9.5 4.4 14.1 10.1 11.5 7.9 10.9 6.2 6.8 ...
##  $ LF    : num  0.51 0.583 0.533 0.577 0.591 0.547 0.519 0.542 0.553
 0.632 ...
##  $ M.F   : num  95 101.2 96.9 99.4 98.5 ...
##  $ Pop   : int  33 13 18 157 18 25 4 50 39 7 ...
##  $ NW    : num  30.1 10.2 21.9 8 3 4.4 13.9 17.9 28.6 1.5 ...
##  $ U1    : num  0.108 0.096 0.094 0.102 0.091 0.084 0.097 0.079 0.08
1 0.1 ...
##  $ U2    : num  4.1 3.6 3.3 3.9 2 2.9 3.8 3.5 2.8 2.4 ...
##  $ Wealth: int  3940 5570 3180 6730 5780 6890 6200 4720 4210 5260 ...
##  $ Ineq  : num  26.1 19.4 25 16.7 17.4 12.6 16.8 20.6 23.9 17.4 ...
##  $ Prob  : num  0.0846 0.0296 0.0834 0.0158 0.0414 ...
##  $ Time  : num  26.2 25.3 24.3 29.9 21.3 ...
##  $ Crime : int  791 1635 578 1969 1234 682 963 1555 856 705 ...
```

Now we do PCA to the original data. Since there are 15 predictors, we will have 15 principal components. Meanwhile, we need to scale the original data.

```
#do PCA
PCA_Model<-prcomp(data[,1:15],scale=TRUE)
summary(PCA_Model)

## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996
##                            PC7     PC8     PC9    PC10    PC11    PC
12
## Standard deviation     0.56729 0.55444 0.48493 0.44708 0.41915 0.358
04
## Proportion of Variance 0.02145 0.02049 0.01568 0.01333 0.01171 0.008
55
## Cumulative Proportion  0.92142 0.94191 0.95759 0.97091 0.98263 0.991
17
##                           PC13   PC14    PC15
## Standard deviation     0.26333 0.2418 0.06793
## Proportion of Variance 0.00462 0.0039 0.00031
## Cumulative Proportion  0.99579 0.9997 1.00000

#first we choose 10 pricipal components to build an original model.
data2 <- PCA_Model$x[,1:10]
pca_crime <- data.frame(cbind(data2,data[,16]))
model_1 <- lm(V11~.,pca_crime)
model_1$coefficients

## (Intercept)         PC1         PC2         PC3         PC4
PC5
##   905.08511    65.21593   -70.08312    25.19408    69.44603  -229.04
282
##         PC6         PC7         PC8         PC9        PC10
##   -60.21329   117.25590    28.71656   -37.17564    56.31771

summary(model_1)

##
## Call:
## lm(formula = V11 ~ ., data = pca_crime)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -428.85 -146.39    9.56  148.94  424.17
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.14  25.753  < 2e-16 ***
```

```
## PC1           65.22      14.48   4.504 6.77e-05 ***
## PC2          -70.08      21.22  -3.302  0.00217 **
## PC3           25.19      25.09   1.004  0.32198
## PC4           69.45      32.95   2.107  0.04211 *
## PC5         -229.04      36.29  -6.312 2.67e-07 ***
## PC6          -60.21      47.76  -1.261  0.21553
## PC7          117.26      62.62   1.872  0.06928 .
## PC8           28.72      64.07   0.448  0.65670
## PC9          -37.18      73.26  -0.507  0.61492
## PC10          56.32      79.46   0.709  0.48303
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 240.9 on 36 degrees of freedom
## Multiple R-squared:  0.6963, Adjusted R-squared:  0.6119
## F-statistic: 8.253 on 10 and 36 DF,  p-value: 9.127e-07
```

We then use stepwise method to remove predictors from model_1 in order to find a model with lowest AICc.

Apply the Stepwise Method:

```
#stepwise method
library(MuMIn)
library(MASS)
model_1 <- lm(V11~.,pca_crime)
stepAICc(model_1,direction='both', steps=1000)
```

```
## Start:  AIC=669.57
## V11 ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7 + PC8 + PC9 + PC10
##
##         Df Sum of Sq     RSS    AIC
## - PC8    1     11661 2101486 523.28
## - PC9    1     14950 2104775 523.35
## - PC10   1     29162 2118988 523.67
## - PC3    1     58541 2148366 524.31
## <none>               2089825 525.01
## - PC6    1     92261 2182087 525.05
## - PC7    1    203535 2293360 527.38
## - PC4    1    257832 2347657 528.48
## - PC2    1    633037 2722862 535.45
## - PC1    1   1177568 3267394 544.02
## - PC5    1   2312556 4402381 558.03
##
## Step:  AIC=666.2
## V11 ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7 + PC9 + PC10
##
##         Df Sum of Sq     RSS    AIC
## - PC9    1     14950 2116436 521.61
## - PC10   1     29162 2130648 521.92
## - PC3    1     58541 2160027 522.57
```

```
## <none>              2101486 523.28
## - PC6   1     92261 2193748 523.30
## + PC8   1     11661 2089825 525.01
## - PC7   1    203535 2305021 525.62
## - PC4   1    257832 2359318 526.72
## - PC2   1    633037 2734523 533.65
## - PC1   1   1177568 3279055 542.19
## - PC5   1   2312556 4414042 556.16
##
## Step:  AIC=663.1
## V11 ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7 + PC10
##
##        Df Sum of Sq     RSS    AIC
## - PC10  1     29162 2145598 520.25
## - PC3   1     58541 2174976 520.89
## <none>              2116436 521.61
## - PC6   1     92261 2208697 521.61
## + PC9   1     14950 2101486 523.28
## + PC8   1     11661 2104775 523.35
## - PC7   1    203535 2319971 523.93
## - PC4   1    257832 2374268 525.01
## - PC2   1    633037 2749473 531.91
## - PC1   1   1177568 3294004 540.40
## - PC5   1   2312556 4428992 554.32
##
## Step:  AIC=660.5
## V11 ~ PC1 + PC2 + PC3 + PC4 + PC5 + PC6 + PC7
##
##        Df Sum of Sq     RSS    AIC
## - PC3   1     58541 2204139 519.52
## - PC6   1     92261 2237859 520.23
## <none>              2145598 520.25
## + PC10  1     29162 2116436 521.61
## + PC9   1     14950 2130648 521.92
## + PC8   1     11661 2133937 522.00
## - PC7   1    203535 2349133 522.51
## - PC4   1    257832 2403430 523.59
## - PC2   1    633037 2778635 530.40
## - PC1   1   1177568 3323166 538.82
## - PC5   1   2312556 4458154 552.62
##
## Step:  AIC=658.69
## V11 ~ PC1 + PC2 + PC4 + PC5 + PC6 + PC7
##
##        Df Sum of Sq     RSS    AIC
## - PC6   1     92261 2296400 519.45
## <none>              2204139 519.52
## + PC3   1     58541 2145598 520.25
## + PC10  1     29162 2174976 520.89
## + PC9   1     14950 2189189 521.20
```

```
## + PC8    1     11661 2192478 521.27
## - PC7    1    203535 2407673 521.67
## - PC4    1    257832 2461970 522.72
## - PC2    1    633037 2837176 529.38
## - PC1    1   1177568 3381707 537.64
## - PC5    1   2312556 4516694 551.24
##
## Step:  AIC=657.7
## V11 ~ PC1 + PC2 + PC4 + PC5 + PC7
##
##        Df Sum of Sq     RSS    AIC
## <none>              2296400 519.45
## + PC6   1     92261 2204139 519.52
## + PC3   1     58541 2237859 520.23
## + PC10  1     29162 2267238 520.84
## + PC9   1     14950 2281450 521.14
## + PC8   1     11661 2284739 521.21
## - PC7   1    203535 2499935 521.44
## - PC4   1    257832 2554232 522.45
## - PC2   1    633037 2929437 528.89
## - PC1   1   1177568 3473968 536.90
## - PC5   1   2312556 4608956 550.19

##
## Call:
## lm(formula = V11 ~ PC1 + PC2 + PC4 + PC5 + PC7, data = pca_crime)
##
## Coefficients:
## (Intercept)          PC1          PC2          PC4          PC5
##      905.09        65.22       -70.08        69.45      -229.04
##         PC7
##      117.26
```

The result shows it is better to use pc1, pc2, pc4, pc5, pc7 as predictors to build the linear regression model.

```
#use the 5 mentioned predictors
data3 <- PCA_Model$x[,c(1,2,4,5,7)]
pca_crime_2 <- data.frame(cbind(data3,data[,16]))
model_2 <- lm(V6~.,pca_crime_2)
summary(model_2)

##
## Call:
## lm(formula = V6 ~ ., data = pca_crime_2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -493.75 -143.08  -12.83  132.37  424.51
##
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      34.52  26.218  < 2e-16 ***
## PC1            65.22      14.22   4.585 4.21e-05 ***
## PC2           -70.08      20.85  -3.362  0.00169 **
## PC4            69.45      32.37   2.146  0.03788 *
## PC5          -229.04      35.65  -6.426 1.07e-07 ***
## PC7           117.26      61.51   1.906  0.06364 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 236.7 on 41 degrees of freedom
## Multiple R-squared:  0.6663, Adjusted R-squared:  0.6256
## F-statistic: 16.37 on 5 and 41 DF,  p-value: 7.29e-09
```

We need to have the coefficients of the original predictors.

```
#calculate a0 and ai
a0 <- model_2$coefficients[1]
ai <- PCA_Model$rotation[,c(1,2,4,5,7)] %*% model_2$coefficients[2:6]
a0

## (Intercept)
##    905.0851

ai

##               [,1]
## M       38.0325597
## So      60.4971075
## Ed      -9.8341983
## Po1    125.4631441
## Po2    121.2746143
## LF      50.7331948
## M.F    131.4762193
## Pop     67.5469488
## NW      59.5559717
## U1      -0.8783094
## U2      38.0946966
## Wealth  18.2562394
## Ineq    66.0621812
## Prob   -97.5257989
## Time   -30.2544918
```

However, this coefficients are calculated using the scaled data. In order to do prediction to a given city, we need to unscale the coefficients. While scaling, we subtracted the mean and divides by the standard deviation, for each variable. We then have this equation: ai * (x - mean)/sd+a0 = original_ai * x+original_a0. That means: (1) original_ai = ai/sd (2) original_a0 = a0-ai*mean/sd,

```
#do the scale calculation in reverse
originalai <- ai/sapply(data[,1:15],sd)
```

```
originala0 <- a0 - sum(ai*sapply(data[,1:15],mean)/sapply(data[,1:15],s
d))
```

Now we have a proper regression model. We can use this model to predict number of crimes in a given city. We use the data given in question 8.2, which is: M = 14.0 So = 0 Ed = 10.0 Po1 = 12.0 Po2 = 15.5 LF = 0.640 M.F = 94.0 Pop = 150 NW = 1.1 U1 = 0.120 U2 = 3.6 Wealth = 3200 Ineq = 20.1 Prob = 0.04 Time = 39.0

```
#apply pca to predict crimes using the data given in question 8.2
test_data_set <-data.frame(M = 14,So = 0, Ed = 10.0, Po1 = 12.0, Po2 =
15.5,LF = 0.64, M.F = 94.0, Pop = 150, NW = 1.1, U1 = 0.12, U2 = 3.6, W
ealth = 3200, Ineq = 20.1, Prob = 0.040, Time = 39.0)
result <- as.matrix(test_data_set) %*% as.matrix(originalai)+originala0

result

##          [,1]
## [1,] 1357.472
```

The result shows the number of crimes predicted is about 1357, which is reasonable.