Homework 6-Regression with PCA

Mark Pearl

19/02/2020

9.1 Regression with PCA

Using the same crime data set uscrime.txt as in Question 8.2, apply Principal Component Analysis and then create a regression lm_model using the first few principal components. Specify your new lm_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)!)

```
uscrime_data <- read.table('C:/Users/mjpearl/Desktop/omsa/ISYE-6501-OAN/hw6/data/uscrime.txt',header = T
head(uscrime_data)
```

```
##      M So   Ed  Po1  Po2    LF   M.F Pop   NW    U1  U2 Wealth Ineq
## 1 15.1  1  9.1  5.8  5.6 0.510  95.0  33 30.1 0.108 4.1   3940 26.1
## 2 14.3  0 11.3 10.3  9.5 0.583 101.2  13 10.2 0.096 3.6   5570 19.4
## 3 14.2  1  8.9  4.5  4.4 0.533  96.9  18 21.9 0.094 3.3   3180 25.0
## 4 13.6  0 12.1 14.9 14.1 0.577  99.4 157  8.0 0.102 3.9   6730 16.7
## 5 14.1  0 12.1 10.9 10.1 0.591  98.5  18  3.0 0.091 2.0   5780 17.4
## 6 12.1  0 11.0 11.8 11.5 0.547  96.4  25  4.4 0.084 2.9   6890 12.6
##       Prob    Time Crime
## 1 0.084602 26.2011   791
## 2 0.029599 25.2999  1635
## 3 0.083401 24.3006   578
## 4 0.015801 29.9012  1969
## 5 0.041399 21.2998  1234
## 6 0.034201 20.9995   682
```

9.1 Regression for PCA for US Crime Data

The following plots will conduct exploratory analysis on the data to get a sense of the data's distribution for each variable.

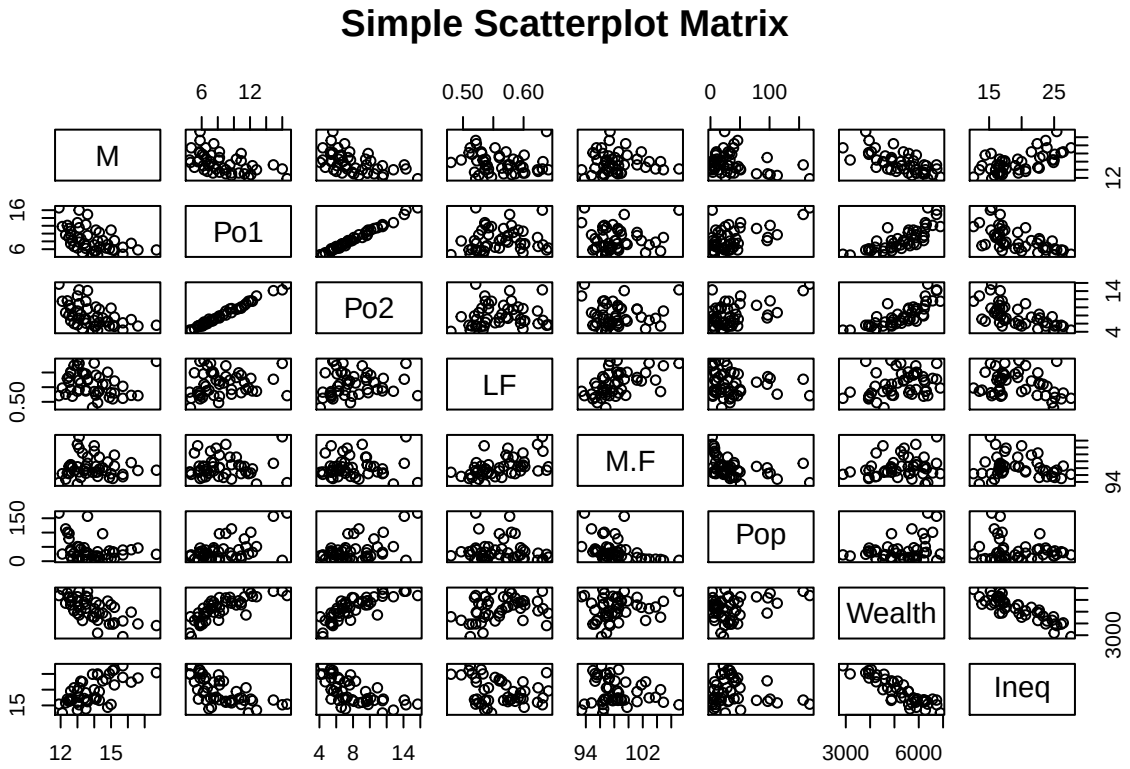
```
summary(uscrime_data)
```

```
##        M               So              Ed             Po1
##  Min.   :11.90   Min.   :0.0000   Min.   : 8.70   Min.   : 4.50
##  1st Qu.:13.00   1st Qu.:0.0000   1st Qu.: 9.75   1st Qu.: 6.25
##  Median :13.60   Median :0.0000   Median :10.80   Median : 7.80
##  Mean   :13.86   Mean   :0.3404   Mean   :10.56   Mean   : 8.50
##  3rd Qu.:14.60   3rd Qu.:1.0000   3rd Qu.:11.45   3rd Qu.:10.45
##  Max.   :17.70   Max.   :1.0000   Max.   :12.20   Max.   :16.60
##       Po2               LF              M.F              Pop
##  Min.   : 4.100   Min.   :0.4800   Min.   : 93.40   Min.   :  3.00
##  1st Qu.: 5.850   1st Qu.:0.5305   1st Qu.: 96.45   1st Qu.: 10.00
##  Median : 7.300   Median :0.5600   Median : 97.70   Median : 25.00
```

```
##  Mean   : 8.023   Mean   :0.5612   Mean   : 98.30   Mean   : 36.62
##  3rd Qu.: 9.700   3rd Qu.:0.5930   3rd Qu.: 99.20   3rd Qu.: 41.50
##  Max.   :15.700   Max.   :0.6410   Max.   :107.10   Max.   :168.00
##        NW               U1               U2            Wealth
##  Min.   : 0.20   Min.   :0.07000   Min.   :2.000   Min.   :2880
##  1st Qu.: 2.40   1st Qu.:0.08050   1st Qu.:2.750   1st Qu.:4595
##  Median : 7.60   Median :0.09200   Median :3.400   Median :5370
##  Mean   :10.11   Mean   :0.09547   Mean   :3.398   Mean   :5254
##  3rd Qu.:13.25   3rd Qu.:0.10400   3rd Qu.:3.850   3rd Qu.:5915
##  Max.   :42.30   Max.   :0.14200   Max.   :5.800   Max.   :6890
##       Ineq             Prob             Time           Crime
##  Min.   :12.60   Min.   :0.00690   Min.   :12.20   Min.   : 342.0
##  1st Qu.:16.55   1st Qu.:0.03270   1st Qu.:21.60   1st Qu.: 658.5
##  Median :17.60   Median :0.04210   Median :25.80   Median : 831.0
##  Mean   :19.40   Mean   :0.04709   Mean   :26.60   Mean   : 905.1
##  3rd Qu.:22.75   3rd Qu.:0.05445   3rd Qu.:30.45   3rd Qu.:1057.5
##  Max.   :27.60   Max.   :0.11980   Max.   :44.00   Max.   :1993.0
```

There's potentially a few variables in this dataset such as Population which could require scaling or normalization. In addition, based on our last findings from a previous homework, it might be beneficial to remove outlier values towards the upper quartile.

```
## 75% of the sample size for the uscrime dataset
pairs(~M+Po1+Po2+LF+M.F+Pop+Wealth+Ineq,data=uscrime_data,
   main="Simple Scatterplot Matrix")
```



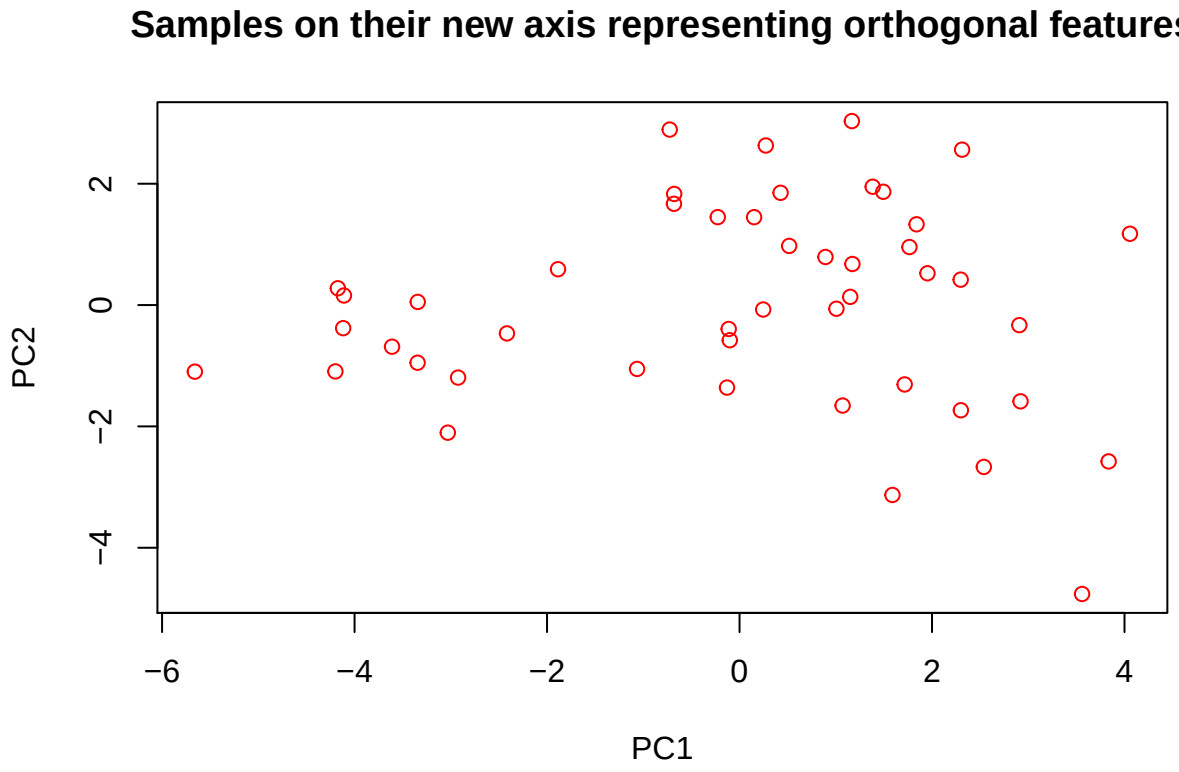
As we can see, there seems to be a positive correlation between Po1 and Po2 and a negative correlation between Wealth and Inequality. There are several approaches we can use to deal with these features. One approach through feature engineering would be to conduct PCA (Principal Component Analysis) on the correlated features to produce a net new feature which alleviates the co-linearity.

PCA will calculate the eigenvector corresponding to the largest eigenvalue of the covariance matrix. These PCA features will help us explain the greatest proportion of the variability in the dataset.

```
#Conduct PCA on the training dataset
pca <- prcomp(uscrime_data[-16],  scale=TRUE)

# create coloring label
class.color <- c(rep(2,100),rep(3,100))

plot(pca$x, col = class.color, main = 'Samples on their new axis representing orthogonal features')
```



Based on our result we can see that with the orthogonal representation it significantly reduces the multicolineairty of these features, which will make up a majority of the variance for the dataset. Let's determine how much variance is explained by our principal component features.

```
# calculate the variance explained by the PCs in percent
variance.total <- sum(pca$sdev^2)
variance.explained <- pca$sdev^2 / variance.total * 100
print(variance.explained)
```

```
##  [1] 40.1263510 18.6789802 13.3662956  7.7480520  6.3886598  3.6879593
##  [7]  2.1454579  2.0493418  1.5677019  1.3325395  1.1712360  0.8546007
```

```
## [13]  0.4622779  0.3897851  0.0307611
```

From our findings we can see that over 50% of the variance can be explained by the first 5 PCA features from the result. Let's use these to now construct a new lm_model to use the first 5 features and see how this impacts our performance results.

```
#number of PCs we want to test = k
k  = 5

#we now combine PCs 1:k with the crime data from our original data set
pca_crimedata <- cbind(pca$x[,1:k],uscrime_data[,16])

lm_model <- lm(V6~., data = as.data.frame(pca_crimedata))
summary(lm_model)
```

```
##
## Call:
## lm(formula = V6 ~ ., data = as.data.frame(pca_crimedata))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -420.79 -185.01   12.21  146.24  447.86
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   905.09      35.59  25.428  < 2e-16 ***
## PC1            65.22      14.67   4.447 6.51e-05 ***
## PC2           -70.08      21.49  -3.261  0.00224 **
## PC3            25.19      25.41   0.992  0.32725
## PC4            69.45      33.37   2.081  0.04374 *
## PC5          -229.04      36.75  -6.232 2.02e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 244 on 41 degrees of freedom
## Multiple R-squared:  0.6452, Adjusted R-squared:  0.6019
## F-statistic: 14.91 on 5 and 41 DF,  p-value: 2.446e-08
```

We can see compared to last week's results that we get a lower adjusted R2 value of 0.62. However since the difference is insignificant we can conclude that the model performs just as well with a reduced feature set. In production setting this can be very useful, espiecially for reducing training time!

```
#now we will run the predict function on our test dataset to determine the performance of the model
test_data <- data.frame(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, In

pred_df <- data.frame(predict(pca, test_data))
pred <- predict(lm_model, pred_df)
```

We can conclude our model produces nearly the same accuracy at a fraction of the cost as the observed value is very close with that we determined in exercise 8.2!