HW6

Richard Hardis, Alan Lo, Arjun Goyal, James Trawick

9/25/2019

# HW6: PCA

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)!)

Load in data and then perform principal components analysis on the predicting variables.

# install.packages('caTools')  
library(caTools)  
getwd()

## [1] "C:/Users/richa/Documents/GitHub/6501-hw/HW6"

setwd('C:/Users/richa/Documents/GitHub/6501-hw/HW6')  
crime\_df = read.table("uscrime.txt", header = TRUE)  
  
pred\_df = crime\_df[1:15]  
#pdf\_scaled = scale(pred\_df, scale = TRUE, center=TRUE)  
  
pca\_model = prcomp(pred\_df, center=TRUE, 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 PC12  
## Standard deviation 0.56729 0.55444 0.48493 0.44708 0.41915 0.35804  
## Proportion of Variance 0.02145 0.02049 0.01568 0.01333 0.01171 0.00855  
## Cumulative Proportion 0.92142 0.94191 0.95759 0.97091 0.98263 0.99117  
## 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

We will be using the first 9 principal components as these cumulatively explain above 95% of the variation in the dataset.

Now we make a new dataframe combining the data for the first 9 PCs and the target variable.

#use k = 9 PC's  
k=9  
pca\_crime = cbind(pca\_model$x[,1:k], crime\_df[,16])  
head(pca\_crime)

## PC1 PC2 PC3 PC4 PC5 PC6  
## [1,] -4.199284 -1.0938312 -1.11907395 0.67178115 0.05528338 0.3073383  
## [2,] 1.172663 0.6770136 -0.05244634 -0.08350709 -1.17319982 -0.5832373  
## [3,] -4.173725 0.2767750 -0.37107658 0.37793995 0.54134525 0.7187223  
## [4,] 3.834962 -2.5769060 0.22793998 0.38262331 -1.64474650 0.7294884  
## [5,] 1.839300 1.3309856 1.27882805 0.71814305 0.04159032 -0.3940902  
## [6,] 2.907234 -0.3305421 0.53288181 1.22140635 1.37436096 -0.6922513  
## PC7 PC8 PC9   
## [1,] -0.56640816 -0.007801727 0.22350995 791  
## [2,] 0.19561119 0.154566472 0.43677720 1635  
## [3,] 0.10330693 0.351138883 0.06299232 578  
## [4,] 0.26699499 -1.547460841 -0.37954181 1969  
## [5,] 0.07050766 -0.543237437 0.22463245 1234  
## [6,] 0.22648209 0.562323186 0.41772217 682

Generate a multiple linear regression model with the pca\_crime dataframe.

pca\_regression = lm(V10 ~ ., data=as.data.frame(pca\_crime))  
summary(pca\_regression)

##   
## Call:  
## lm(formula = V10 ~ ., data = as.data.frame(pca\_crime))  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -455.9 -132.5 21.5 139.9 393.0   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 905.09 34.91 25.928 < 2e-16 \*\*\*  
## PC1 65.22 14.38 4.535 5.88e-05 \*\*\*  
## PC2 -70.08 21.08 -3.325 0.00201 \*\*   
## PC3 25.19 24.92 1.011 0.31857   
## PC4 69.45 32.73 2.122 0.04061 \*   
## PC5 -229.04 36.04 -6.355 2.08e-07 \*\*\*  
## PC6 -60.21 47.44 -1.269 0.21228   
## PC7 117.26 62.20 1.885 0.06728 .   
## PC8 28.72 63.64 0.451 0.65446   
## PC9 -37.18 72.76 -0.511 0.61244   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 239.3 on 37 degrees of freedom  
## Multiple R-squared: 0.692, Adjusted R-squared: 0.6171   
## F-statistic: 9.239 on 9 and 37 DF, p-value: 3.588e-07

Now transform the data back into the original space.

b0 = pca\_regression$coefficients[1]  
b = pca\_regression$coefficients[2:(k+1)]  
  
# Matrix multiply the eigenvalue matrix by the coefficients  
a = pca\_model$rotation[,1:k] %\*% b  
  
# Unscale the a coefficients vector by multiplying by the respective standard deviations and adding the means  
a\_unscaled = (a \* pca\_model$scale) + pca\_model$center  
a0\_unscaled = b0 - sum((a\*pca\_model$scale)+ pca\_model$center)

We can see that the forumula for this equation in the original space has the following values:

print(a\_unscaled)

## [,1]  
## M 73.8901823  
## So 47.2109363  
## Ed 15.7791421  
## Po1 384.7699774  
## Po2 353.2470697  
## LF 1.7407456  
## M.F 505.9095140  
## Pop 1068.5646394  
## NW 598.9895267  
## U1 -0.3570662  
## U2 24.9722176  
## Wealth 42813.6394897  
## Ineq 229.5143060  
## Prob -2.6372247  
## Time 17.0957716

print(a0\_unscaled)

## (Intercept)   
## -45227.24

Using this model to predict the crime rate, we get the following output:

new\_city = 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,  
Ineq = 20.1,  
Prob = 0.04,  
Time = 39.0)  
  
new\_city\_pred\_df = data.frame(predict(pca\_model, new\_city))  
  
prediction = predict(pca\_regression, new\_city\_pred\_df)  
prediction

## 1   
## 1136.169

The prediction from last week’s multiple linear model including all predicting variables was 155.4349 offenses per 100,000 population in 1960. The PC model with 9 principal components predicted 1136.169 offenses per 100,000 population in 1960. There is a large discrepancy between the two model predictions suggesting that one or both models is not a good predictor of crime rates. A good approach here would be to dive into feature selection and find the best combination of features to predict the crime rate. This can be done using techniques such as stepwise regression which incorporates forward and backward feature selection techniques. This would be useful for assessing both the original model’s inputs and choosing which principal components add the most predictive power to the model instead of the approach taken in this homework where the PC’s were chosen based on how much cumulative variation they explained.