Homework 6: Principal Component Analysis

2025-02-18

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)!)

```
# Load the crime dataset
crime_df <- read.delim("uscrime.txt", header = TRUE)

# Perform PCA on the first 15 predictor variables (scaling included)
output <- prcomp(crime_df[, 1:15], scale. = TRUE, center = TRUE, retx = TRUE)

# Summary of PCA
summary(output)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7
## Standard deviation     2.4534 1.6739 1.4160 1.07806 0.97893 0.74377 0.56729
## Proportion of Variance 0.4013 0.1868 0.1337 0.07748 0.06389 0.03688 0.02145
## Cumulative Proportion  0.4013 0.5880 0.7217 0.79920 0.86308 0.89996 0.92142
##                           PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     0.55444 0.48493 0.44708 0.41915 0.35804 0.26333 0.2418
## Proportion of Variance 0.02049 0.01568 0.01333 0.01171 0.00855 0.00462 0.0039
## Cumulative Proportion  0.94191 0.95759 0.97091 0.98263 0.99117 0.99579 0.9997
##                           PC15
## Standard deviation     0.06793
## Proportion of Variance 0.00031
## Cumulative Proportion  1.00000
```

```
# Display the first 5 principal component loadings (rounded to 2 decimals)
round(output$rotation, 2)[, 1:5]
```

```
##          PC1   PC2   PC3   PC4   PC5
## M      -0.30  0.06  0.17 -0.02 -0.36
## So     -0.33 -0.16  0.02  0.29 -0.12
## Ed      0.34  0.21  0.07  0.08 -0.02
## Po1     0.31 -0.27  0.05  0.33 -0.24
## Po2     0.31 -0.26  0.05  0.35 -0.20
## LF      0.18  0.32  0.27 -0.14 -0.39
## M.F     0.12  0.39 -0.20  0.01 -0.58
## Pop     0.11 -0.47  0.08 -0.03 -0.08
## NW     -0.29 -0.23  0.08  0.24 -0.36
## U1      0.04  0.01 -0.66 -0.18 -0.13
```

```
## U2      0.02 -0.28 -0.58 -0.07 -0.13
## Wealth  0.38 -0.08  0.01  0.12  0.01
## Ineq   -0.37 -0.03  0.00 -0.08 -0.22
## Prob   -0.26  0.16 -0.12  0.49  0.17
## Time   -0.02 -0.38  0.22 -0.54 -0.15
```

```r
# Extract the first 5 principal components
PCA_data <- output$x[, 1:5]

# Combine PCA-transformed data with the response variable (Crime)
crime_PC <- cbind(PCA_data, Crime = crime_df[, 16])

# Build regression model using the first 5 principal components
PCA_output <- lm(Crime ~ ., data = as.data.frame(crime_PC))

# Display summary of PCA regression model
summary(PCA_output)
```

```
##
## Call:
## lm(formula = Crime ~ ., data = as.data.frame(crime_PC))
##
## 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
```

```r
# Extract intercept and coefficients from PCA regression model
intercept_PCA <- PCA_output$coefficients[1]
coeffs_PCA <- PCA_output$coefficients[2:6]

# Convert PCA coefficients back to original variable space
a_vals <- coeffs_PCA %*% t(output$rotation[, 1:5])
coeffs_real <- a_vals / output$scale
intercept_real <- intercept_PCA - sum(a_vals * output$center / output$scale)

# Print original space coefficients
print(coeffs_real)
```

```
##             M       So      Ed      Po1      Po2       LF      M.F      Pop
## [1,] 48.37374 79.01922 17.8312 39.48484 39.85892 1886.946 36.69366 1.546583
##            NW       U1       U2     Wealth     Ineq      Prob     Time
```

```
## [1,] 9.537384 159.0115 38.29933 0.03724014 5.540321 -1523.521 3.838779
```

```
print(intercept_real)
```

```
## (Intercept)
##   -5933.837
```

```
# Predict crime for a new test point
testpts <- 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 = 0.040, Time = 39.0
)

# Compute prediction using original space coefficients
prediction <- intercept_real + as.matrix(testpts) %*% t(coeffs_real)
print(prediction)
```

```
##          [,1]
## [1,] 1388.926
```

```
# Compute predictions for the full dataset
preds <- intercept_real + as.matrix(crime_df[, 1:15]) %*% t(coeffs_real)

# Calculate residual sum of squares (RSS)
rss <- sum((preds - crime_df[, 16])^2)

# Calculate total sum of squares (TSS)
tss <- sum((crime_df[, 16] - mean(crime_df[, 16]))^2)

# Compute R-squared value
rsq <- 1 - rss / tss

# Print RSS, TSS, and R-squared values
print(rss)
```

```
## [1] 2441394
```

```
print(tss)
```

```
## [1] 6880928
```

```
print(rsq)
```

```
## [1] 0.6451941
```

Building the Models

We started by running a PCA on the 15 predictor variables to reduce the complexity of the data. By focusing on the first 5 principal components, we were able to capture around 86% of the total variance in the data. After using these components to build a regression model, we found that **64.5%** of the variation in crime rates could be explained by the model, which is decent but suggests there's room for improvement.

Key Insights from the PCA-based Regression Model:

1. **PC1 has a big impact on crime rates**
   - PC1 had a coefficient of **65.22**, and with a very small p-value (**6.51e-05**), it's clear this component is important. It seems to capture a mix of factors like wealth and income inequality, which are

strongly linked to crime rates.
    - **In short:** PC1 has a strong effect when it comes to predicting crime.
2. **PC2 has an opposite effect, reducing crime rates**
    - The coefficient for **PC2** is **-70.08**, with a p-value of **0.00224**, suggesting that this component is negatively correlated with crime. It likely represents other factors like education or law enforcement efforts.
    - **Takeaway:** PC2 seems to lower crime rates, which might indicate the impact of better education or stronger policing.
3. **PC5 also has a significant negative effect on crime**
    - PC5's coefficient is **-229.04**, with a p-value of **2.02e-07**, meaning it's a strong predictor of lower crime rates. This could reflect certain economic or demographic factors that help reduce crime.
    - **What this means:** PC5 strongly suggests that factors like higher income or better social policies can contribute to fewer crimes.

We used the model to predict the crime rate for a new city, and the result was **1388.93**, vs my output of **question 8.2: 1544.59**, which gives us a good estimate of the crime level for a city with these characteristics.

Takeaways from the PCA Model

The PCA-based model simplifies things by using five principal components, which helps reduce overfitting. But it doesn't explain as much of the variation in crime rates as the full model did ($\mathbf{R^2 = 0.8031}$). That being said, it's still a good model for predicting crime rates in new cities.

What stood out:

- **PC1** and **PC2** were the most influential in the model. Both were statistically significant and had strong effects on crime rates.
- **PC5** also played a big role, suggesting that some economic or demographic factors are key to reducing crime.
- Other components like **PC3** and **PC4** didn't seem to have as much of an impact.

Conclusion

The PCA-based model did a great job of simplifying the data and pinpointing key factors that influence crime, like income inequality and law enforcement presence. While it doesn't capture every nuance of crime rates, it provides useful predictions. For instance, the model predicted a crime rate of **1,388.93** for a new city, compared to the earlier estimate of **1,544.59**. This shows how the refined model still gives valuable insights, though we know crime is influenced by a range of factors that may not be fully captured here.