Chair of Decision Sciences and Systems
TUM School of Computation, Information and Technology
Technical University of Munich

Business Analytics & Machine Learning
Homework sheet 09: Principle Component Analysis – Solution

Prof. Dr. Martin Bichler, Prof. Dr. Jalal Etesami
Julius Durmann, Markus Ewert, Johannes Knörr, Yutong Chao

Exercise H9.1 *PCA Calculation*

You are given the following dataset:

i	x_1	x_2	x_3
0	2	3	4
1	-2	0	2
2	0	1	2
3	4	2	0
4	1	1	1

Table 1 Dataset

You are further given the covariance matrix Σ_x:

$$\Sigma_x = \begin{bmatrix} 5 & 2 & -1 \\ 2 & 1.3 & 0.6 \\ -1 & 0.6 & 2.2 \end{bmatrix}$$

You compute the characteristic polynomial of Σ_x to be:

$$f(\lambda) = \lambda^3 - 8.5\lambda^2 + 15\lambda$$

a) What percentage of the variance does each principal component explain? How many principal components are necessary to explain all of the variance in the data?

b) Compute the principal component that explains most of the variance and scale it to be a unit vector. Using this principal component, calculate the one-dimensional projection of the data (as coordinates along the principal component).

c) A colleague plans to run a linear regression model using the three features x_1, x_2, and x_3. Discuss what problem occurs and how this problem can be mitigated.

d) Now another datapoint $(4, 2.6, 1.2)$ is added to the dataset. Do the eigenvalues, the principal components, and the ratio of explained variance of each component change? Discuss your reasoning.

Solution

a) We compute the eigenvalues as the roots of the characteristic polynomial $f(\lambda) = 0$.

Obviously $\lambda_1 = 0$. We solve a quadratic equation for the other two eigenvalues:

$$\lambda^2 - 8.5\lambda + 15 = 0 \Rightarrow \lambda_2 = 6, \lambda_3 = 2.5$$

The first eigenvalue is zero and thus explains none of the variance in the data. The second eigenvector explains $\frac{6}{6+2.5} = 70.59\%$ of the data and the third eigenvector explains $\frac{2.5}{6+2.5} = 29.41\%$ of the data. We thus only need two principal components to explain all of the variance.

b) We first compute the eigenvalues 6, 2.5, and 0.

We need to consider $\lambda = 6$. We solve the equation

$$(\Sigma_x - \lambda I_3)v = 0 \Leftrightarrow \begin{bmatrix} 5-6 & 2 & -1 \\ 2 & 1.3-6 & 0.6 \\ -1 & 0.6 & 2.2-6 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \Leftrightarrow \begin{cases} -v_1 + 2v_2 - v_3 = 0 \\ 2v_1 - 4.7v_2 + 0.6v_3 = 0 \\ -v_1 + 0.6v_2 - 3.8v_3 = 0 \end{cases}$$

We solve this linear equation system and yield a solution $v = r \begin{bmatrix} 5 \\ 2 \\ -1 \end{bmatrix}$ where r is a scalar. After normalization, we obtain a unit eigenvector $v = \begin{bmatrix} 0.913 \\ 0.365 \\ -0.183 \end{bmatrix}$.

The zero-mean dataset looks as follows ($\overline{x}_1 = 1, \overline{x}_2 = 1.4, \overline{x}_3 = 1.8$):

$$X = \begin{bmatrix} 1 & 1.6 & 2.2 \\ -3 & -1.4 & 0.2 \\ -1 & -0.4 & 0.2 \\ 3 & 0.6 & -1.8 \\ 0 & -0.4 & -0.8 \end{bmatrix}$$

In order to calculate the projected data, we solve the following equation:

$$Z = Xv = \begin{bmatrix} 1 & 1.6 & 2.2 \\ -3 & -1.4 & 0.2 \\ -1 & -0.4 & 0.2 \\ 3 & 0.6 & -1.8 \\ 0 & -0.4 & -0.8 \end{bmatrix} \begin{bmatrix} 0.913 \\ 0.365 \\ -0.183 \end{bmatrix} = \begin{bmatrix} 1.095 \\ -3.286 \\ -1.095 \\ 3.286 \\ 0 \end{bmatrix}$$

c) The second feature is a linear combination of the first and third features. We thus have a case of perfect multicollinearity, violating the Gauss-Markov properties. This is also indicated by the principal components since one component has no explanatory power.

One remedy would be removing the second feature, or conducting a principal component regression only using the first two principal components.

(*Alternative problem: overfitting since the number of features is almost as large as the number of observations*)

d) Applying the same normalization to the new data point as for the other data point yields

$$p = \begin{bmatrix} 4-1 \\ 2.6-1.4 \\ 1.2-1.8 \end{bmatrix} = \begin{bmatrix} 3 \\ 1.2 \\ -0.6 \end{bmatrix} = 3.286v.$$

Therefore, the new data point lies exactly along the calculated principal component. As a result, the eigenvectors / principal components will not change, but the eigenvalues will change, putting more emphasis on the calculated principal component and increasing its ratio of explained variance.

Exercise H9.2 *PCA implementation in Python*

In this exercise, we will conduct PCAs using primitive Python functions, trying to mimic the PCA computation algorithm. We will use the built-in *iris* dataset.
from sklearn.datasets import load_iris
iris = load_iris()
data = iris.data
Note: you may use functions from the numpy *library throughout this exercise.*

a) Check the structure of the dataset. Which attributes are numerical? Compute a correlation matrix for them.

b) Construct a new matrix consisting only of numerical attributes with mean values subtracted.

c) Calculate covariance matrix, eigenvectors and eigenvalues.

d) Compute the PCA scores by multiplying the transposed eigenvectors matrix and the transposed zero-mean matrix. Check the first six scores.

Solution

See *iris_solution.py*.

Exercise H9.3 *PCA in sklearn:* `sklearn.decomposition.PCA`

The code from the previous exercise is cumbersome and long. In this exercise, we will use sklearn.decomposition.PCA to get the same result with less effort.

a) Use sklearn.decomposition.PCA in order to compute PCA's for the *iris* dataset.

b) How much standard deviation does each component have? How much variance is explained by the first component?

c) Check the computed eigenvectors. Are they equal to what was computed in the previous exercise?

d) Check the first six PCA scores for the dataset. Compare them with the scores computed in the previous exercise.

e) Plot the projection of the dataset points on a space consisting of two principal components only.

Solution

See *iris_solution.py*.