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

Business Analytics & Machine Learning
Tutorial sheet 9: Principle Component Analysis – Solution

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

Exercise T9.1 *Principle component analysis*

Given the following dataset $D = \{(-3,-1,-1),(0,-1,0),(-2,-1,2),(1,-1,3)\}$, compute its principal components by following the PCA algorithm introduced in class and generate the transformed data.
Each tuple of the set D represents an observation or row vector.

a) Calculate the zero-mean dataset X from the given dataset D. Note down the means.

b) Calculate the 3 x 3 covariance matrix Σ_X using the following formulas. What can you infer from it?
$var(x_j) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)^2$
$cov(x_j, x_k) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k)$
Reminder: Since the matrix X is centered, you can use the following formulas:
$var(x_j) = \frac{1}{N-1}\sum_{i=1}^{N} x_{ij}^2$
$cov(x_j, x_k) = \frac{1}{N-1}\sum_{i=1}^{N} x_{ij}x_{ik}$

c) Find the eigenvalues for the covariance matrix by solving the equation: $|\Sigma_X - \lambda I_3| = 0$. You can use the *Laplace expansion* to calculate the determinant.

d) Find the corresponding eigenvectors and order them by significance. How is the variance distributed among them?

Hint: Solving the equation $(\Sigma_X - \lambda I_3)v = 0$ gives you the corresponding eigenvectors.

e) Compute a one-dimensional PCA projection of the dataset.

f) Compute a two-dimensional PCA projection of the dataset.

Hint for e.) and f.): The general formula for projections is: $Z = X\Phi$

Solution

a) The matrix of our dataset D looks as follows:

$$D = \begin{bmatrix} -3 & -1 & -1 \\ 0 & -1 & 0 \\ -2 & -1 & 2 \\ 1 & -1 & 3 \end{bmatrix}$$

We calculate the mean values for each feature:
$\overline{d}_1 = \frac{1}{4} \times ((-3) + 0 + (-2) + 1) = \frac{1}{4} \times (-4) = -1$
$\overline{d}_2 = \frac{1}{4} \times ((-1) + (-1) + (-1) + (-1)) = \frac{1}{4} \times (-4) = -1$
$\overline{d}_3 = \frac{1}{4} \times ((-1) + 0 + 2 + 3) = \frac{1}{4} \times (4) = 1$

We transform our dataset to a zero means dataset by subtracting the means:
$x_{ij} = d_{ij} - \overline{d}_j$

$$X = \begin{bmatrix} -3-(-1) & -1-(-1) & -1-1 \\ 0-(-1) & -1-(-1) & 0-1 \\ -2-(-1) & -1-(-1) & 2-1 \\ 1-(-1) & -1-(-1) & 3-1 \end{bmatrix} = \begin{bmatrix} -2 & 0 & -2 \\ 1 & 0 & -1 \\ -1 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix}$$

b) The covariance matrix of the centered dataset is computed by determining the variances $var(x_j)$ for each feature and the covariance $cov(x_j, x_k)$ between features.

$$\Sigma_X = \begin{bmatrix} var(x_1) & cov(x_1,x_2) & cov(x_1,x_3) \\ cov(x_2,x_1) & var(x_2) & cov(x_2,x_3) \\ cov(x_3,x_1) & cov(x_3,x_2) & var(x_3) \end{bmatrix}$$

Given that the mean of each feature is now 0, we calculate:
$var(x_j) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)^2 = \frac{1}{N-1}\sum_{i=1}^{N} x_{ij}^2$
$var(x_1) = \frac{1}{4-1} \times ((-2)^2 + 1^2 + (-1)^2 + 2^2) = \frac{1}{3} \times (4+1+1+4) = \frac{10}{3}$

Similarly, $var(x_2) = 0$ and $var(x_3) = \frac{10}{3}$.

The covariance is calculated:
$cov(x_j, x_k) = \frac{1}{N-1}\sum_{i=1}^{N}(x_{ij} - \overline{x}_j)(x_{ik} - \overline{x}_k) = \frac{1}{N-1}\sum_{i=1}^{N} x_{ij}x_{ik}$
$cov(x_1, x_2) = \frac{1}{4-1} \times ((-2)\times 0 + 1\times 0 + (-1)\times 0 + 2\times 0) = \frac{1}{3}\times 0 = 0$
$cov(x_1, x_3) = \frac{1}{4-1} \times ((-2)\times(-2) + 1\times(-1) + (-1)\times 1 + 2\times 2) = \frac{1}{3}\times 6 = 2$
$cov(x_2, x_3) = \frac{1}{4-1} \times (0\times(-2) + 0\times(-1) + 0\times 1 + 0\times 2) = \frac{1}{3}\times 0 = 0$

We do not need to compute the other covariance values as we know that the covariance matrix is symmetric.

$$\Sigma_X = \begin{bmatrix} \frac{10}{3} & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & \frac{10}{3} \end{bmatrix}$$

From the covariance matrix, we can expect the variables x_1 and x_3 to increase together, they are positively correlated. We also notice that for x_2 the variance and the relative covariance values are 0. That is because it is a constant feature, i.e. its column has only one value, and therefore no variance. We can already assume that the characteristic polynomial of the covariance matrix will yield an eigenvalue of value 0.

c) To compute the eigenvalues of the covariance matrix Σ_x, we need to solve the characteristic equation $|\Sigma_x - \lambda I_3| = 0$.

First, we derive the characteristic polynomial of Σ_x: $\Sigma_x - \lambda I_3 = \begin{bmatrix} \frac{10}{3} & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & \frac{10}{3} \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$

$$\begin{bmatrix} \frac{10}{3} - \lambda & 0 & 2 \\ 0 & -\lambda & 0 \\ 2 & 0 & \frac{10}{3} - \lambda \end{bmatrix}$$

$|\Sigma_x - \lambda I_3| = (\frac{10}{3} - \lambda)(-1)^2(-\lambda(\frac{10}{3} - \lambda)) + 2(-1)^4(0 + 2\lambda) = -\lambda(\frac{10}{3} - \lambda)^2 + 4\lambda = -\lambda(\frac{100}{9} - \frac{20}{3}\lambda + \lambda^2) + 4\lambda = -\lambda^3 + \frac{20}{3}\lambda^2 - \frac{64}{9}\lambda$

Then we solve the characteristic equation for λ:

$-\lambda^3 + \frac{20}{3}\lambda^2 - \frac{64}{9}\lambda = 0 \Rightarrow -9\lambda^3 + 60\lambda^2 - 64\lambda = 0 \Rightarrow \lambda(-9\lambda^2 + 60\lambda - 64) = 0$

Hence $\lambda_1 = 0$. We solve the second degree equation for the other two eigenvalues:

$-9\lambda^2 + 60\lambda - 64 = 0 \Rightarrow \lambda_2 = \frac{4}{3}, \lambda_3 = \frac{16}{3}$

d) The corresponding eigenvectors are found by using these values of λ in the equation $(\Sigma_x - \lambda I_3)v = 0$.

We can already ignore $\lambda_1 = 0$, because variance along the corresponding eigenvector would be exactly 0, i.e. all data points fall in the exact same point. Consequently, no significant principal component will be computed. Nevertheless, for correctness:

- For $\lambda_1 = 0$:

  $(\Sigma_x - 0I_3)v = 0 \Rightarrow \begin{bmatrix} \frac{10}{3} & 0 & 2 \\ 0 & 0 & 0 \\ 2 & 0 & \frac{10}{3} \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \Rightarrow \begin{cases} \frac{10}{3}v_1 + 2v_3 = 0 \\ 2v_1 + \frac{10}{3}v_3 = 0 \end{cases} \Rightarrow v_1 = v_3 = 0$, while v_2 can take any value. Thus the eigenvectors of Σ_x corresponding to $\lambda_1 = 0$ are of the form $r\begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$, where r is a scalar.

- For $\lambda_2 = \frac{4}{3}$:

  $(\Sigma_x - \frac{4}{3}I_3)v = 0 \Rightarrow \begin{bmatrix} 2 & 0 & 2 \\ 0 & -\frac{4}{3} & 0 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \Rightarrow \begin{cases} 2v_1 + 2v_3 = 0 \\ -\frac{4}{3}v_2 = 0 \\ 2v_1 + 2v_3 = 0 \end{cases} \Rightarrow v_1 = -v_3 \text{and} v_2 = 0.$

  Thus the eigenvectors of Σ_x corresponding to $\lambda_2 = \frac{4}{3}$ are of the form $r\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$, where r is a scalar. After normalizing we get $\begin{bmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ -\frac{\sqrt{2}}{2} \end{bmatrix}$

- For $\lambda_3 = \frac{16}{3}$:

  $(\Sigma_x - \frac{16}{3}I_3)v = 0 \Rightarrow \begin{bmatrix} -2 & 0 & 2 \\ 0 & -\frac{16}{3} & 0 \\ 2 & 0 & -2 \end{bmatrix} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = 0 \Rightarrow \begin{cases} -2v_1 + 2v_3 = 0 \\ -\frac{16}{3}v_2 = 0 \\ 2v_1 - 2v_3 = 0 \end{cases} \Rightarrow v_1 = v_3 \text{and} v_2 = 0.$

  Thus the eigenvectors of Σ_x corresponding to $\lambda_3 = \frac{16}{3}$ are of the form $r\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$, where r is a scalar. After normalizing we get $\begin{bmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix}$, which is clearly orthogonal to the other eigenvectors.

  The eigenvectors are ordered in decreasing order by the corresponding eigenvalues. We have $\lambda_3 > \lambda_2 > \lambda_1$. Therefore the rotation matrix – or the principal component loadings – is:

$$\Phi = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \end{bmatrix}$$

The proportion of explained variance is calculated by the eigenvalues' ratio to their sum:

$\frac{\lambda_3}{\lambda_1+\lambda_2+\lambda_3} = \frac{\frac{16}{3}}{\frac{20}{3}} = 80\%$ of the variance explained by the first principal component

$\frac{\lambda_3+\lambda_2}{\lambda_1+\lambda_2+\lambda_3} = \frac{\frac{20}{3}}{\frac{20}{3}} = 100\%$ of the variance explained by the first two principal components

As expected, the last principal component does not contribute to describing the data.

e) For the 1D projection, we multiply the centered dataset with the first principal component:

$$Z = \begin{bmatrix} -2 & 0 & -2 \\ 1 & 0 & -1 \\ -1 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ 0 \\ \frac{\sqrt{2}}{2} \end{bmatrix} = \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 0 \\ 2\sqrt{2} \end{bmatrix}$$

f) For the 2D projection, we multiply the centered dataset with the first two principal components:

$$Z = \begin{bmatrix} -2 & 0 & -2 \\ 1 & 0 & -1 \\ -1 & 0 & 1 \\ 2 & 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ 0 & 0 \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix} = \begin{bmatrix} -2\sqrt{2} & 0 \\ 0 & \sqrt{2} \\ 0 & -\sqrt{2} \\ 2\sqrt{2} & 0 \end{bmatrix}$$

Exercise T9.2 *PCA Reconstruction*

Making use of the PCA projections computed in Exercise 10.1, restore the original dataset using the formula: $D \approx Z\Phi^T +$ means

a) Reverse the one-dimensional PCA projection to restore the original data. How would the data look when plotted into the original coordinate system?

b) What result do you expect when reconstructing the original data from the two-dimensional PCA projection? What is the information loss?

Solution

a) To restore the original dataset from the principal component scores, we multiply the new dataset with the transposed eigenvectors and add the original dimension means:

$D \approx Z\Phi^T +$ means

$$D \approx \begin{bmatrix} -2\sqrt{2} \\ 0 \\ 0 \\ 2\sqrt{2} \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{bmatrix} + \begin{bmatrix} -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} -3 & -1 & -1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ 1 & -1 & 3 \end{bmatrix}$$

When plotting this dataset, the data points would be aligned, as we considered the projection of the data on only one dimension, omitting the variance on the second one. We lose some information in this reconstruction, but we have conserved 80% of the original variance.

b) We do the same for the 2D PCA projection:

$$D \approx \begin{bmatrix} -2\sqrt{2} & 0 \\ 0 & \sqrt{2} \\ 0 & -\sqrt{2} \\ 2\sqrt{2} & 0 \end{bmatrix} \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix} + \begin{bmatrix} -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \\ -1 & -1 & 1 \end{bmatrix} = \begin{bmatrix} -3 & -1 & -1 \\ 0 & -1 & 0 \\ -2 & -1 & 2 \\ 1 & -1 & 3 \end{bmatrix} = D$$

The information loss in this reconstruction is zero. This is due to the fact that the first two principal components explain 100% of the variance of the data.

Exercise T9.3 *PCA Understanding*

You are given the following dataset:

$$D = \{(1,-4), (-2,2), (0,-2), (-1,-1), (2,-3)\}$$

a) Apply principal component analysis and determine the two principal component vectors

b) Draw the data points, the principal components, and the data set recovered from the 1D projection in a single plot.

c) Now another point (a,b) is added to the dataset. As a results, the principal components do not change. Determine a possible point (a,b).

d) We now adjust the original dataset in the following three ways

   1) Multiply both coordinates of the first point by some factor $k \in \mathbb{R}$.
   2) Multiply all coordinates of all points by the same factor $k \in \mathbb{R}$.
   3) Flip all coordinates of all points, e.g. $(1,-4) \rightarrow (-4,1)$.

   In each case, will the eigenvalues and/or principal components change? Discuss your reasoning.

Solution

a) The matrix of our dataset D looks as follows:

$$D = \begin{bmatrix} 1 & -4 \\ -2 & 2 \\ 0 & -2 \\ -1 & -1 \\ 2 & -3 \end{bmatrix}$$

With $\overline{d}_1 = 0$ and $\overline{d}_2 = -1.6$, we obtain the following zero-mean matrix:

$$X = \begin{bmatrix} 1 & -2.4 \\ -2 & 3.6 \\ 0 & -0.4 \\ -1 & 0.6 \\ 2 & -1.4 \end{bmatrix}$$

The covariance matrix of X will be

$$\Sigma_X = \begin{bmatrix} 2.5 & -3.25 \\ -3.25 & 5.3 \end{bmatrix}$$

To compute eigenvalues, we need to solve

$$|\Sigma_x - \lambda I_2| = 0 \Leftrightarrow (2.5-\lambda)(5.3-\lambda) - 3.25^2 = 0 \Leftrightarrow \lambda^2 - 7.8\lambda + 2.6875 = 0$$

Solving the quadratic equation yields $\lambda_1 = 7.44$ and $\lambda_2 = 0.36$.

To obtain the eigenvectors, we need to solve $(\Sigma_x - \lambda I_2)v = 0$. For λ_1 we obtain $r\begin{bmatrix} 1 \\ -1.52 \end{bmatrix}$, and for λ_2 we obtain $r\begin{bmatrix} 1 \\ 0.66 \end{bmatrix}$.

In each case, we choose r to normalize the eigenvectors. We obtain the following principal components:

$$v_1 = \begin{bmatrix} 0.55 \\ -0.83 \end{bmatrix}, v_2 = \begin{bmatrix} 0.83 \\ 0.55 \end{bmatrix}$$

b) To compute the 1D-dataset, we first multiply the zero-mean data with the first principal component. We then project the data back and add the means.

$$\tilde{D} = Xv_1v_1^T + \text{means} = \begin{bmatrix} 1 & -2.4 \\ -2 & 3.6 \\ 0 & -0.4 \\ -1 & 0.6 \\ 2 & -1.4 \end{bmatrix} \begin{bmatrix} 0.55 \\ -0.83 \end{bmatrix} \begin{bmatrix} 0.55 & -0.83 \end{bmatrix} + \begin{bmatrix} 0 & -1.6 \\ 0 & -1.6 \\ 0 & -1.6 \\ 0 & -1.6 \\ 0 & -1.6 \end{bmatrix} = \begin{bmatrix} 1.40 & -3.73 \\ -2.26 & 1.83 \\ 0.18 & -1.88 \\ -0.58 & -0.72 \\ 1.25 & -3.50 \end{bmatrix}$$

We can now plot the original data points (blue), the projected data points (orange) and the principal components (black).

c) The principal components will not change if the point (a, b) is perfectly explained by the first principal component. We can thus pick any point along the direction of the first eigenvector. For example, any of the projected points from b) satisfy this requirement, e.g. $(a, b) = (1.40, -3.73)$

d) 1) This will change both eigenvalues and eigenvectors. Scaling the coordinates of just one point changes the zero-mean and covariance matrix, resulting in different eigenvalues and eigenvectors.

2) The eigenvalues will change, but the eigenvectors will be identical. Multiplying each coordinate by some k will result in a covariance matrix $k^2\Sigma_X$. The eigenvalues will be $k^2\lambda_1$ and $k^2\lambda_2$. As eigenvectors, we thus obtain

$$k^2 r\begin{bmatrix} 1 \\ -1.52 \end{bmatrix}, k^2 r\begin{bmatrix} 1 \\ 0.66 \end{bmatrix}$$

Standardizing these vectors will lead to the same eigenvectors v_1 and v_2 regardless of the chosen k.

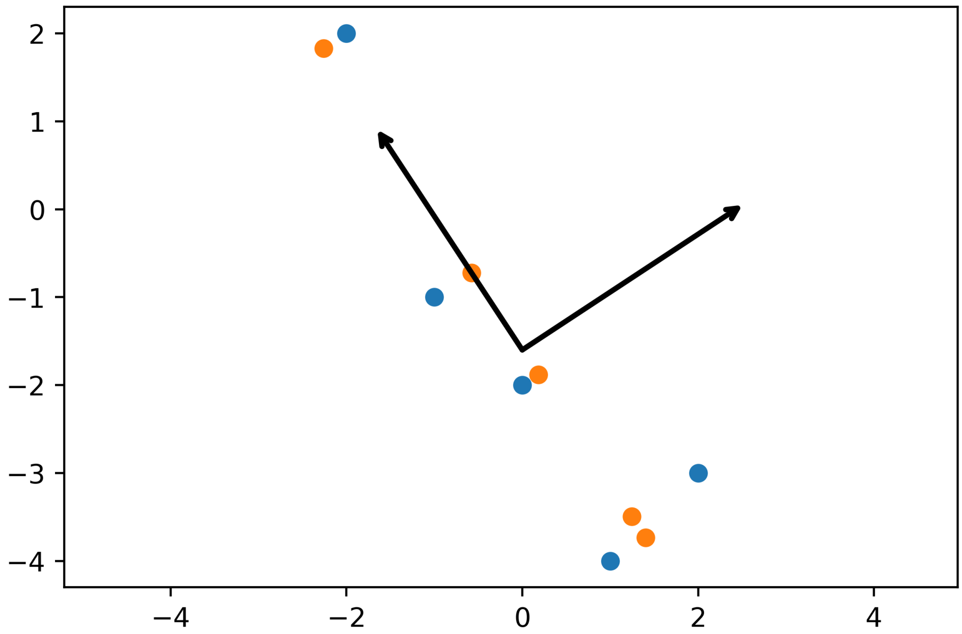

Figure 1 Original data, Eigenvectors, and 1D-Data

3) "Flipping" coordinates is equivalent to an isometric affine transformation / line reflection. The "structure" of the data thus does not change. As a result, the eigenvalues are identical, but the eigenvectors will also undergo the reflection, i.e. the new principal components will be

$$v_1 = \begin{bmatrix} -0.83 \\ 0.55 \end{bmatrix}, v_2 = \begin{bmatrix} 0.55 \\ 0.83 \end{bmatrix}$$

Exercise T9.4 *PCA for image compression*

The goal of this exercise is to use Principal Component Analysis for image compression in Python.

a) Load an image of the famous painting "The Starry Night" by Vincent Van Gogh and store red, green, and blue values as color matrices. You can access the painting here. Use the provided template *image_compression_PCA_template.py* to load the painting, store color values, and to plot it.

b) Apply a PCA to each color matrix to identify principal color vectors. Choose $n = 5$ as the number of components. Plot the resulting compressed image. The following code snippet may be helpful:

```
for i in range(image.shape[2]):
  # get channel image, normalize data to [0, 1] before applying PCA
  channel_data = img[:, :, i] / 255.0

  # perform pca
  pca = PCA(n_components=number_of_components)
  pca.fit(channel_data)
  compressed_channel_data = pca.inverse_transform(pca.transform(channel_data))
```

c) Apply PCA for $n = \{1, 2, 5, 10, 20\}$ (if your computer permits, you can increase n even further). Plot and save the compressed image in each iteration. Observe the size of the image files. Looking at the images, at what point do you clearly identify the painting?

d) Determine a reasonable number of clusters using the "elbow criterion". For this purpose, create a scree plot that plots the explained variance of each component against the number of components, e.g. for $n \in [1, 10]$. Does the elbow point correspond to your visual impression in d)?

e) Compare the image compression using PCA with the image compression using clustering from last week.

Solution

See "solution.py".