Dimensionality_Reduction_Technique_Principal_Component_Analysis

December 13, 2023

Jasmine Sellers[1]

[1]University of Washington undergraduate

Abstract: This paper provides a comprehensive overview of a dimensionality reduction technique called principal component analysis (PCA). The investigation begins with the mathematical underpinnings of PCA, discussing eigenvalues, eigenvectors, and data transformation into a lower-dimensional space. The work is structured with an introduction to the main steps of PCA supplemented with a toy example.

The paper then introduces a data visualization code utilizing PCA. This drafting code connects theory and practice to explore datasets, applying PCA to observe visualizations in reduced-dimensional spaces and gain insights into data trends and relationships.

Keywords: dimensionality reduction, principal component analysis (PCA), covariance, variance, eigenvector, eigenvalue, feature vector

Introduction Dimensionality reduction techniques are methods that reduce the dimensions in a dataset while retaining key characteristics of the data. These techniques are applied in domains such as machine learning, data analysis, and data visualization. It tackles several key issues including higher-dimensional data, noise, and complexity. Some dimensionality reduction techniques include neural networks, matrix factorization, and non-metric multidimensional scaling. This paper will concentrate on principal component analysis (PCA), a commonly used method which utilizes linear algebra in the computation of the covariance matrix and principal components.

Principal Component Analysis Principal component analysis (PCA) is a standard tool used in data analysis due to its non-parametric method for emphasizing the most significant information from high-dimensional datasets. The primary goal of PCA is to transform the data to lower dimensions while preserving its variability. This paper will use an example alongside the main formulas to explain principal component analysis.

I will use the following generic example to walk through the main steps of PCA where X, Y, Z represent variables and x_1, x_2, $\ldots$ represent values of the corresponding variables.

X	Y	Z
x_1	y_1	z_1
x_2	y_2	z_2
x_3	y_3	z_3

1. **Standardization**
   The standardization of data in PCA uses the following formulas:

- mean = $\frac{\text{sum of the terms}}{\text{total number of terms}}$
- standard deviation (SD) = $\sqrt{\sum_{i=1}^{n} \frac{(x_i - mean)^2}{n}}$
- z = $\frac{\text{value - mean}}{\text{standard deviation}}$

The standardization process calculates the mean of each column/variable which is used to calculate the standard deviation of each column/variable. Then the z-score of each value is computed and place its corresponding component. This can be illustrated as such:

X's corresponding mean	Y's corresponding mean	Z's corresponding mean
$\frac{x_1+x_2+x_3}{n}$	$\frac{y_1+y_2+y_3}{n}$	$\frac{z_1+z_2+z_3}{n}$

X's corresponding SD	Y's corresponding SD	Z's corresponding SD
$\sqrt{\sum_{i=1}^{n} \frac{(x_i-\bar{x})^2}{n}}$	$\sqrt{\sum_{i=1}^{n} \frac{(y_i-\bar{y})^2}{n}}$	$\sqrt{\sum_{i=1}^{n} \frac{(z_i-\bar{z})^2}{n}}$

Standardize data set calculated by taking each value and subtracting by the column's mean and dividing by the column's standard deviation.

Standardized values of X	Standardized values of Y	Standardized values of Z
$\frac{x_1-\bar{x}}{\text{SD of x}}$	$\frac{y_1-\bar{y}}{\text{SD of y}}$	$\frac{z_1-\bar{z}}{\text{SD of z}}$
$\frac{x_2-\bar{x}}{\text{SD of x}}$	$\frac{y_2-\bar{y}}{\text{SD of y}}$	$\frac{z_2-\bar{z}}{\text{SD of z}}$
$\frac{x_3-\bar{x}}{\text{SD of x}}$	$\frac{y_3-\bar{y}}{\text{SD of y}}$	$\frac{z_3-\bar{z}}{\text{SD of z}}$

2. **Covariance Matrix Computation**
   Covariance is a measure of the relationship between two variables and calculated using the following formula where x and y are two random variables:

- $Cov(X,Y) = \sum_{i=1}^{n} \frac{(x_i-\bar{x})(y_i-\bar{y})}{\text{n}}$ where n is the number of data points There are two important properties of covariance:

1. $Cov(X,X) = Var(X)$
2. $Cov(X,Y) = Cov(Y,X)$ The covariance matrix is defined as $C_x = \frac{1}{n}AA^T$ where A is an mxn matrix in which the rows of A represent the variables measured (e.g. X) and the columns represent individual observations (e.g. x_1, y_1, z_1). Continuing from the example, in this case, $A = \begin{pmatrix} x_1 & x_2 & x_3 \\ y_1 & y_2 & y_3 \\ z_1 & z_2 & z_3 \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ thus the covariance matrix, $C_x = \frac{1}{3}\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}\begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix} =$ $\begin{pmatrix} Var(X) & Cov(X,Y) & Cov(X,Z) \\ Cov(Y,X) & Var(Y) & Cov(Y,Z) \\ Cov(Z,X) & Cov(Z,Y) & Var(Z) \end{pmatrix}$

   - This is an example of a covariance calculation: $Cov(X,Y) = \frac{(x_1-\bar{x})(y_1-\bar{y})+(x_2-\bar{x})(y_2-\bar{y})+(x_3-\bar{x})(y_3-\bar{y})}{3}$ If the covariance is positive, then this indicates positive correlation between the two variables and vice versa.

3. **Eigenvector Decomposition: Determining the Principal Components**
   The next step in solving for the principal components of the data set is to find the eigenvalue and eigenvector of the covariance matrix, C_x. Eigenvector in linear algebra refers to a non-zero vector when multiplied by a square matrix, A, results in a scaled version of the vector. This can be represented as Av = λv in which v is the eigenvector of A and λ is the eigenvalue. The eigenvectors should then be ordered descending, based on their corresponding eigenvalues to organize the variables based on order of significance. The eigenvector with the largest eigenvalue is the principal component. Following this, the user can choose the n number of variables to focus on, transforming the data set to n dimensions. The chosen eigenvectors are called the feature vector. The final data set is formed by matrix multiplication of the standardized data set by the featured vector.

4. **Proportion of Total Variance**
   Recall that the eigenvalues are represented with λ, the proportion of total variance which we can represent with V can be represented as such: $V_i = \frac{\lambda_i}{\sum_{i=1}^{n} \lambda_j}$ Where n is the number of principal components and V_i is the proportion of total variance for the ith principal component.

5. **Interpretation of Principal Components and Proportion of Total Variance**
   As mentioned previously the principal components are those with the highest eigenvalues as it captures the most significant relationships between the data. A high principal component value indicates that the corresponding variables contribute significantly to the underlying pattern/trend of the data. The opposite holds if the principal component is a low value.
   The proportion of total variance characterizes the importance of that principal component in explaining patterns and relationships within the data. The higher the proportion for a specific principal component, the more variability that is captured by this principal component from the original dataset. In other words, a larger value indicates the corresponding component explains a larger portion of the variability in the data set. In practical application, this means the corresponding principal component has more significance in representing primary relationships in the data.

Python Script for Dimensionality Reduction with PCA

1. Principal Component Analysis (PCA) Code Example

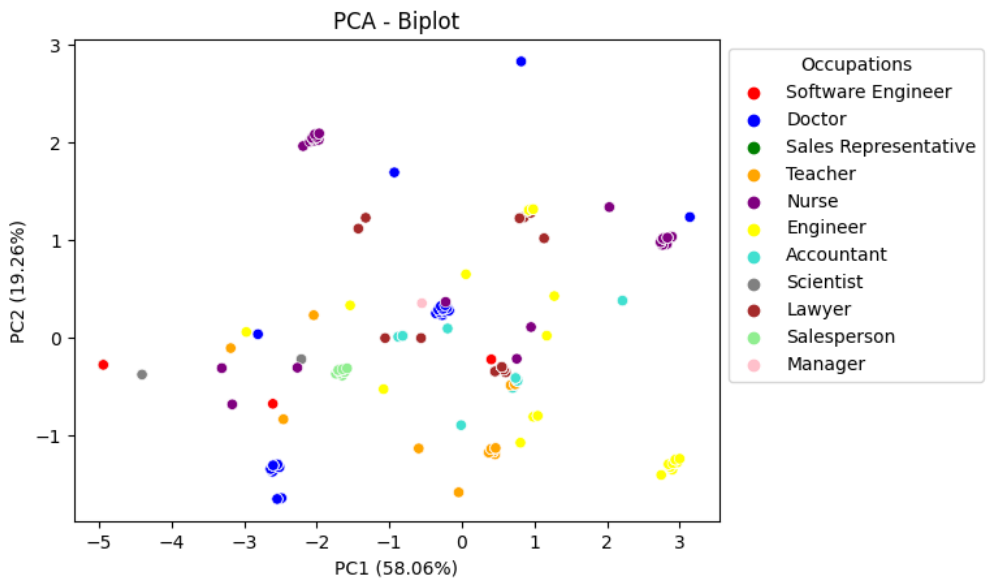
Suppose you want to interpret the following data set of six variables of interest (i.e., six dimensions). The following code will walk through an example of how to perform PCA to interpret the trends in the dataset focusing on six variables: age, sleep duration (hrs), quality of sleep (1-10, subjective rating), physical activity (min/day), stress level (1-10, subjective rating), and heart rate (bpm): https://www.kaggle.com/datasets/uom190346a/sleep-health-and-lifestyle-dataset/data. The following dataset can be downloaded, and the code can be run through GoogleColab to show the computations that can be done on a six-dimensional dataset. The challenge of observing trends in a data set of more than two to three variables becomes apparent. Thus, this example walks through how we can utilize Python to perform the calculations described above to give us the principal components and portion of total variance, which we can use for analysis. The main functions to note from the code below are np.cov()and np.linalg.eig()

[4]:
```python
import numpy as np
import pandas as pd
from google.colab import files
import matplotlib.pyplot as plt
import seaborn as sns

# Upload a file from your device
uploaded = files.upload()
# Get the file path from chosen csv file
file_path = list(uploaded.keys())[0]

# Getting sample data from the uploaded csv file
excel_data = file_path
columns_to_read = ['Age', 'Sleep Duration', 'Quality of Sleep',
'Physical Activity Level', 'Stress Level', 'Heart Rate']
myData = pd.read_csv(excel_data, usecols = columns_to_read)
myOccupations = pd.read_csv(excel_data, usecols = ['Occupation'])

# Step 1: Standardization
ex_variable_means = np.mean(myData, axis=0)
ex_variable_sd = np.std(myData, axis=0)
ex_standardized_data = (myData - ex_variable_means) / ex_variable_sd

# Step 2: Covariance Matrix Computation
ex_cov_matrix = np.cov(ex_standardized_data, rowvar=False)

# Step 3: Eigenvector Decomposition
ex_eigenvalues, ex_principal_components = np.linalg.eig(ex_cov_matrix)

# Step 4: Proportion of Total Variance
ex_proportion_of_total_variance = ex_eigenvalues / np.sum(ex_eigenvalues)

print("Covariance Matrix:")
print(ex_cov_matrix)
print("\nPrincipal Component:")
print(ex_principal_components)
print("\nEigenvalues: ")
print(ex_eigenvalues)
print("\nProportion of Total Variance:")
print(ex_proportion_of_total_variance)

# Stores the two largest eigenvalues from step 3
def two_largest_indices(arr):
    if len(arr) < 2:
        raise ValueError("Array must have at least two elements")
    sorted_indices = np.argsort(arr)
    largest_index = sorted_indices[-1]
    second_largest_index = sorted_indices[-2]
    return largest_index, second_largest_index

ex_indicies_of_largest = two_largest_indices(ex_eigenvalues)
```

```
# Get the corresponding principal components of the two largest eigenvalues
ex_two_principal_components = np.column_stack((ex_principal_components[:
↪,ex_indicies_of_largest[0]],
                                               ex_principal_components[:
↪,ex_indicies_of_largest[1]]))
print('\nTransformed Data')
print(ex_two_principal_components)

# Reduction of Dimensions (higher dimension -> 2D)
transformed_data = np.dot(ex_standardized_data, ex_two_principal_components)
np_occupations = myOccupations.to_numpy().flatten()

data_and_occupation = {'PC1': transformed_data[:,0],
                       'PC2': transformed_data[:,1],
                       'Occupation': np_occupations}

df_PC1_PC2_Occupation = pd.DataFrame(data_and_occupation)

# Visualization of the data
# creates color codes for the possible occupations in the dataset
occupation_palette = {'Software Engineer': 'red', 'Doctor': 'blue', 'Nurse':␣
↪'purple', 'Engineer': 'yellow',
                      'Teacher': 'orange', 'Sales Representative': 'green',␣
↪'Lawyer': 'brown',
                      'Accountant': 'turquoise','Scientist': 'gray', 'Manager':␣
↪'pink', 'Salesperson': 'lightgreen'}
# creates a scatter plot with x as the first principal components and y as the␣
↪second principal components.
sns.scatterplot(x = 'PC1',
                y = 'PC2',
                hue ='Occupation',
                data = df_PC1_PC2_Occupation,
                palette = occupation_palette)

plt.title('PCA - Biplot')
plt.xlabel('PC1 (' + str(round(ex_proportion_of_total_variance[0] * 100, 2)) +␣
↪'%)')
plt.ylabel('PC2 (' + str(round(ex_proportion_of_total_variance[1] * 100, 2)) +␣
↪'%)')
plt.legend(loc='upper left', bbox_to_anchor=(1,1), title='Occupations')

print('\n')
plt.show()
```

```
# sources: https://colab.research.google.com/notebooks/io.ipynb, https://
↪pythonbasics.org/read-csv-with-pandas/,  https://www.geeksforgeeks.org/
↪python-read-csv-using-pandas-read_csv/,  https://numpy.org/doc/stable/
↪reference/routines.linalg.html, https://www.tutorialspoint.com/
↪how-to-plot-a-graph-in-python, https://seaborn.pydata.org/generated/seaborn.
↪scatterplot.html
```

```
<IPython.core.display.HTML object>

Saving Sleep_health_and_lifestyle_dataset.csv to
Sleep_health_and_lifestyle_dataset (2).csv
Covariance Matrix:
[[ 1.00268097  0.34563351  0.47500394  0.17947259 -0.42347677 -0.22621103]
 [ 0.34563351  1.00268097  0.88558086  0.21292965 -0.81319735 -0.51783949]
 [ 0.47500394  0.88558086  1.00268097  0.1934136  -0.90116155 -0.66163381]
 [ 0.17947259  0.21292965  0.1934136   1.00268097 -0.03422598  0.1373382 ]
 [-0.42347677 -0.81319735 -0.90116155 -0.03422598  1.00268097  0.67182278]
 [-0.22621103 -0.51783949 -0.66163381  0.1373382   0.67182278  1.00268097]]

Principal Component:
[[ 0.2929922   0.32080985 -0.88702029 -0.06758719 -0.11525077 -0.08112241]
 [ 0.47875506  0.07451706  0.26325654  0.44265275 -0.61248009 -0.35336026]
 [ 0.51936269  0.03237107  0.09197523  0.11185927  0.04546088  0.84034272]
 [ 0.08751607  0.84322811  0.3560927  -0.37018261  0.10375667 -0.08188207]
 [-0.50177136  0.11927532 -0.04595293 -0.18341171 -0.74656532  0.37534995]
 [-0.39468721  0.40647278 -0.08073337  0.78504968  0.20345387  0.12160401]]

Eigenvalues:
[3.49302358 1.15856571 0.73627157 0.41417762 0.14670846 0.06733885]

Proportion of Total Variance:
[0.58061399 0.19257799 0.12238382 0.06884503 0.02438603 0.01119313]

Transformed Data
[[ 0.2929922   0.32080985]
 [ 0.47875506  0.07451706]
 [ 0.51936269  0.03237107]
 [ 0.08751607  0.84322811]
 [-0.50177136  0.11927532]
 [-0.39468721  0.40647278]]
```



The scatterplot above plots the values of the PC1 and PC2 against each other. PC1 represents the standardized data matrix multiplied by the corresponding eigenvector of the largest eigenvalue. PC2 represents the standardized data matrix multiplied by the corresponding eigenvector of the second-largest eigenvalue. One method in which the graph may be interpreted is by observing any separation or clustering of the scatterplot. Cluster points may have some relationship in the original high-dimensional data space. Outliers in the PCA plot can also indicate deviation from the general pattern of the data.
For instance, examining the visual representation above reveals intriguing clusters among the sampled individuals. When categorizing individuals by their respective occupations, distinctive groups emerge, encompassing nurses, doctors, teachers, salespersons, engineers, and accountants. While this observation may not serve as conclusive evidence for shared characteristics in variables such as age, sleep duration (in hours), sleep quality (rated on a subjective scale of 1-10), daily physical activity (in minutes), stress levels (rated on a subjective scale of 1-10), and heart rate (in beats per minute) among specific occupations, it does provide valuable insights into discernible patterns within this sample. Moreover, analyzing the proportion of total variance for each variable provides valuable insights into their respective contributions to the overall variability of the data. For example, the calculations conducted in step 4 reveal that age and sleep duration significantly contribute to the dataset's variability, accounting for 58.06% and 19.26%, respectively, according to covariance calculations. This observation becomes particularly relevant when assessing the consistency of variables within the sample, shedding light on which factors maintain consistant values and which exhibit greater variability.In conclusion, the exploration of principal component analysis not only reveals patterns and relationships within the data but also highlights variables that significantly influence its variance.

2. **Comprehensive PCA Analyzer: Adaptive Dimensionality Reduction Tool**

The following code allows for the user to upload their .csv file with the data to which principal component analysis will be performed on (without the data/sample IDs). The inputed higher dimension data will then be reduced to two dimensions using the first and second largest principal components. The code will output the principal components with their corresponding eigenvalues, proportion of total variance, and a visualization of the data.

```
[ ]: # Upload a file from your device
     uploaded = files.upload()
     # Get the file path from chosen csv file
     file_path = list(uploaded.keys())[0]

     # Getting sample data from the uploaded csv file
     excel_data = file_path
     inputData = pd.read_csv(excel_data)

     # Step 1: Standardization
     variable_means = np.mean(inputData, axis=0)
     variable_sd = np.std(inputData, axis=0)
     standardized_data = (inputData - variable_means) / variable_sd

     # Step 2: Covariance Matrix Computation
     cov_matrix = np.cov (standardized_data, rowvar=False)

     # Step 3: Eigenvector Decomposition
     eigenvalues, principal_components = np.linalg.eig(cov_matrix)

     # Proportion of Total Variance
     proportion_of_total_variance = eigenvalues / np.sum(eigenvalues)

     # Get the two largest numbers
     two_principal_components = np.column_stack((principal_components[:
      ↪,indicies_of_largest[0]], principal_components[:,indicies_of_largest[1]]))
     print('\nTransformed Data')
     print(two_principal_components)

     # Reduction of Dimensions (higher dimension -> 2D)
     transformed_data = np.dot(standardized_data, two_principal_components)

     # Visualization of the data
     plt.scatter(transformed_data[:,0], transformed_data[:,1])
     plt.title('PCA - Biplot')
     plt.xlabel('PC1 (' + str(round(proportion_of_total_variance[0] * 100, 2)) + '%)')
     plt.ylabel('PC2 (' + str(round(proportion_of_total_variance[1] * 100, 2)) + '%)')

     print('\n')
     plt.show()
```

```
<IPython.core.display.HTML object>
```

Discussion Notably, PCA is a dimensionality reduction technique; thus, the interpretation of the graphs commonly involves a combination of different statistical analyses. Other statistical analyses may include linear regression, cluster analysis, and time series analysis. Additionally, while PCA can be a powerful reduction technique, not all patterns will have clear and meaningful interpretations.

Conclusions Dimensionality reduction techniques allow for the reduction of dimensions in a dataset while retaining key characteristics of the data. As evidenced by the sample data set, it tackles several issues, including higher-dimensional data, noise, and complexity. Principal component analysis (PCA), the reduction technique focused upon in this paper, facilitates the exploration of high-dimensional datasets by identifying principal components and their contributions to variance. PCA not only simplifies data representation but also uncovers patterns and relationships of the dataset, thus making it a robust supplemental tool in data analysis.

References

1. Google. (n.d.). Google Colaboratory. Retrieved from https://colab.research.google.com/notebooks/io.ipynb

2. Hastie, T., Tibshirani, R., & Friedman, J. H. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)*. New York: Springer.

3. Kruskal, J.B. (1964). Nonmetric multidimensional scaling: A numerical method. *Psychometrika, 29*, 115–129. https://doi.org/10.1007/BF02289694

4. Jolliffe, I.T. (1986). *Principal Component Analysis*. Springer Series in Statistics. New York, NY: Springer. https://doi.org/10.1007/978-1-4757-1904-8_1

5. Jolliffe, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences, 374*(2065), 20150202. https://doi.org/10.1098/rsta.2015.0202

6. *Linear Algebra (numpy.linalg)*. (n.d.). NumPy v1.26 Manual. Retrieved from https://numpy.org/doc/stable/reference/routines.linalg.html

7. Read CSV with Pandas. (n.d.). Python Tutorial. Retrieved from https://pythonbasics.org/read-csv-with-pandas/

8. Shlens, J. (2014). A tutorial on principal component analysis. arXiv preprint arXiv:1404.1100.

9. Smith, L. I. (2002). A tutorial on principal components analysis.

10. Poole, D. (2015). Section 4.1 Introduction to Eigenvalues and Eigenvectors. In *Linear Algebra: A Modern Introduction* (pp. 254–260). Cengage Learning.