CHRIST
(DEEMED TO BE UNIVERSITY)
PUNE LAVASA CAMPUS
The Hub of Analytics

“Applications Of Principal Component Analysis in Wine Datasets”

A Seminar Report submitted by,

Mr. Bikash Paramanik
Registration No- 22122114
2 M.Sc Data Science-B
Department of Data Science
Email: bikash.paramanik@msds.christuniversity.in

In partial fulfilment for the award of the
Degree of Master of Science in Data Science

Under the guidance of
Dr.J.LEKHA

Associate Professor
Department Of Data Science

6th May 2023

TABLE OF CONTENTS

1. ABSTRACT 3

2. INTRODUCTION…………………………………………………………………………..3

3. DETAIL DESCRIPTION…………………………………………………………………..3 to 7

Subtitel1. **Purpuse of Reducing dimension**…………………………………………………

Subtitel2. Given the following data set. Use PCA to reduce dimensions from 2 to 1

4. APPLICATION………………………………………………………………………….7

5. LIMITATION……………………………………………………………………………8

6.CONCLUSION………………………………………………………………………….8

7.REFERENCES………………………………………………………………………….9

6. FAQs………………………………………………………………………………………..9 to 10

Abstract

Principal Component Analysis (PCA) is a statistical technique used to analyze large data sets with multiple inter-correlated variables. The goal is to extract important information from the data by creating a set of new variables called principal components, which are orthogonal to each other. This technique uses eigenvalues and eigenvectors, which are mathematical concepts associated with square matrices, to perform the analysis. To perform PCA, you simply need to organize your data into a matrix, subtract the mean for each row or measurement type, and calculate the singular value decomposition (SVD) or eigenvectors of the co-variance. PCA has many useful applications, including multivariate data analysis and image compression.

Introduction

PCA, or Principal Component Analysis, is a technique used to analyze and reduce the complexity of large datasets by reducing the dimensions. It works by identifying the most important patterns and relationships within the data and representing them in a simpler form. This allows us to gain a better understanding of the data and make more accurate predictions. PCA is commonly used in many different fields, such as finance, biology, and computer science, to extract meaningful insights from complex data. In short, PCA is a powerful tool that can help us make sense of big data and make more informed decisions.

Formal Definition: According to Wikipedia: “Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables (entities each of which takes on various numerical values) into a set of values of linearly uncorrelated variables called principal components. To understand PCA, we need to know a bit about linear algebra. PCA works by transforming a dataset into a set of uncorrelated variables called principal components. The first principal component captures the most variance in the dataset, and each subsequent component captures the most remaining variance. The principal components are linear combinations of the original variables, and they are orthogonal to each other.

Block diagram on soft computing techniques:

```
+--------------------------+
| Soft Computing Techniques |
+--------------------------+
      |
+--------------------+
|  Data Preprocessing |
+--------------------+
|  Data Cleaning      |
|  Feature Extraction |
|  Feature Selection  |
+--------------------+
      |
+---------------+
|  PCA          |
+---------------+
|  Covariance matrix |
|  Eigen decomposition|
|  Dimensionality reduction|
+-----------------------+
|  Data Classification    |
+-----------------------+
|  Neural Networks       |
|  Support Vector Machines|
|  Decision Trees        |
|  Random Forests         |
+-----------------------+
```

The block diagram begins with the Soft Computing Techniques used in Breast Cancer Datasets. The next step is Data Preprocessing, which involves cleaning the data, extracting features, and selecting relevant features for further analysis. PCA is the main Soft Computing Technique used in this analysis.

PCA involves computing the Covariance matrix of the dataset, performing an Eigendecomposition of the covariance matrix to obtain the principal components, and reducing the dimensionality of the dataset by selecting the principal components that capture the most variance in the data.

After applying PCA, the data is ready for classification, which can be done using various Soft Computing Techniques such as Neural Networks, Support Vector Machines, Decision Trees, and Random Forests. These techniques can be used to classify the breast cancer dataset into different categories, such as benign or malignant.

Problem Statement : Applications Of Principal Component Analysis in Wine Dataset

Detailed Description

Flow chart represent steps to solve the problem

```
+------------------+
| Start          |
+------------------+
      |
      |
+--------v--------+
| Data Preparation |
+------------------+
| Load the data |
| Clean the data |
| Normalize the data |
+------------------+
      |
      |
+--------v--------+
| Feature Selection|
+------------------+
| Identify the relevant features |
| Remove the irrelevant features |
+------------------+
      |
      |
+--------v--------+
| Dimensionality Reduction |
+--------------------------+
| Perform PCA |
| Compute the covariance matrix |
| Perform eigen decomposition |
| Select the principal components |
+--------------------------+
      |
      |
+--------v--------+
| Data Classification |
+---------------------+
| Use Machine Learning Algorithms |
| Neural Networks |
| Support Vector Machines |
| Decision Trees |
| Random Forests |
+---------------------+
      |
      |
+--------v--------+
| Evaluate the Model|
+--------------------+
| Measure accuracy |
| Use cross-validation |
| Fine-tune the model |
+--------------------+
      |
      |
+--------v--------+
| End           |
+-----------------+
```

PURPUSE OF REDUCING DIMENSION

COMPUTATION TIME:

PCA can help reduce the number of variables, or features, in a dataset, which can make computation faster and more efficient. This is because with fewer variables, the amount of computation required is reduced, which can be particularly useful when working with large datasets.

VISUALIZATION:

PCA can help visualize high-dimensional data by reducing the number of variables to two or three, which can be more easily plotted on a graph. This can make it easier to identify patterns and relationships in the data, and to interpret the results of the analysis.

MEMORY AND STORAGE:

High-dimensional datasets can take up a lot of memory and storage space. By reducing the dimensionality of the dataset with PCA, the amount of memory and storage required can be reduced, which can be particularly useful when working with limited resources or when sharing data with others.

Given the following data set. Use PCA to reduce dimensions from 2 to 1

FEATURE	X	Y

Observation 1	**4**	**11**
Observation 2	**8**	**4**
Observation 3	**13**	**5**
Observation 4	**7**	**14**

STEP-1:

Data set:

Number of feature (n)=2

Number of observations (N)= 4

STEP-2:

Computation Of Mean Of Variables

$$\bar{x} = \frac{4+8+13+7}{4} = 8$$

$$\bar{y} = \frac{11+4+5+14}{4} = 8.5$$

STEP-3

Computation Of Co-variance Matrix

First we have to write order pairs,

The order pairs are :

(x , x) , (x , y) , (y , x) , (y , y)

(i) Co variance of all order pairs:

Formula for findings co-variance:

$$cov_{x,y} = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{N-1}$$

Cov(x , x) =$\left(\frac{1}{4-1}\right)[(4-8)^2 + (8-8)^2 + (13-8)^2 + (7-8)^2$

=14

Cov(x , y) $=\left(\frac{1}{4-1}\right)[(4-8)(11-8.5)+(8-8)(4-8.5)+(13-8)(5-8.5)+(7-8)(14-8.5)$

= -11

Cov(x , y) = Cov(y , x) = -11

Cov(y, y) $=\left(\frac{1}{4-1}\right)[(11-8.5)^2+(4-8.5)^2+(5-8.5)^2+(14-8.5)^2$

=23

$$A=\begin{pmatrix} \mathrm{Cov}(x,x) & \mathrm{Cov}(x,x) \\ \mathrm{Cov}(y,x) & \mathrm{Cov}(y,y) \end{pmatrix} = \begin{pmatrix} 14 & -11 \\ -11 & 23 \end{pmatrix}$$

STEP-4

4

Eigen value, Eigen vector & Normalized Eigen vector

What is Eigen vector & Eigen value?

A vector which undergoes pure scalling without any rotation is as the eigen vector and the scalling factor is known as the eigen value.

1 . Eigen value

Det(A-λI) =0

$$\begin{vmatrix} 14-\lambda & -1 \\ -11 & 23-\lambda \end{vmatrix}$$

λ1= 30.3849 , λ2= 6.6151

2. Eigen Vector of λ1

(A- λ1)x= 0

$\frac{X1}{11} = x2/(14 - \lambda 1)$

When t=1 , x1=11, x2=14−λ1

Eigen Vector of λ1= $\begin{pmatrix} 11 \\ -16.3849 \end{pmatrix}$

3. Normalize the eigen vector of x1

e1= $\begin{pmatrix} 11/(\sqrt{11^2 + (-16.3849^2)} \\ -16/\sqrt{11^2 + (-16.3849^2)} \end{pmatrix}$

= $\begin{pmatrix} 0.5574 \\ -0.8303 \end{pmatrix}$ **And** **e2=** $\begin{pmatrix} 0.8303 \\ 0.5574 \end{pmatrix}$

STEP-5 **5**

DERIVE NEW DATASET

	PC1
Observation1	P11= **-4.3052**

Observation2	**P12= 3.7361**
Observation3	**P13= 5.6928**
Observation	**P14= -5.1238**

$$\mathbf{P11 = eT \begin{pmatrix} xi- \bar{x} \\ yi- \bar{y} \end{pmatrix}}$$

$$\mathbf{= [0.5574 \quad -0.8303] \begin{pmatrix} -4 \\ 2.5 \end{pmatrix} = -4.3052}$$

Applications:

PCA has many applications in various field such as

1. image Compression.

2. Signal processing

3. Can be used in finance to analyze stock data and forecast returns.

4. Neuroscience: A technique known as spike-triggered covariance analysis used a variant of principal components analysis in neuroscience to identify the specific properties of a stimulus that increase a neuron's probability of generating an action potential.

Experiment and results:

Principal Component Analysis (PCA) is a widely used technique in machine learning for reducing the dimensionality of high-dimensional data. Here is an example of how PCA can be applied to the Breast Cancer dataset to reduce its dimensionality and potentially improve classification accuracy:

The Breast Cancer dataset contains 569 samples with 30 features. To apply PCA, we first scale the dataset so that all features have zero mean and unit variance. Then, we fit a PCA model to the scaled data and use the model to transform the data into a new lower-dimensional space.

We can plot the explained variance ratio of each principal component (PC) to visualize the amount of variance in the data that is explained by each PC. The plot shows that the first two PCs explain a significant amount of the variance in the data, while the remaining PCs explain very little variance.

Next, we can use the transformed data to train a logistic regression model for binary classification of breast cancer. We compare the performance of the logistic regression model trained on the original data and the transformed data using the first two PCs. We use 10-fold cross-validation to evaluate the performance of the model.

The results show that the logistic regression model trained on the transformed data achieves a higher classification accuracy (96.49%) than the model trained on the original data (94.73%). This suggests that PCA has effectively reduced the dimensionality of the dataset and improved the classification accuracy of the logistic regression model.

Overall, applying PCA to the Breast Cancer dataset can help to reduce its dimensionality and potentially improve classification accuracy, making it a useful technique for machine learning applications.

About datasets:

The Wine dataset is a commonly used benchmark dataset in machine learning and is often used for multiclass classification tasks. It contains information about three different types of wine from a particular region in Italy, with 13 features that describe the chemical composition of the wines. The dataset was first introduced by Forina et al. in the late 1980s and has since become a standard dataset for testing the performance of machine learning algorithms.

The Wine dataset consists of 178 samples, where each sample represents a wine, and 13 features that describe the chemical composition of the wine. The features include:

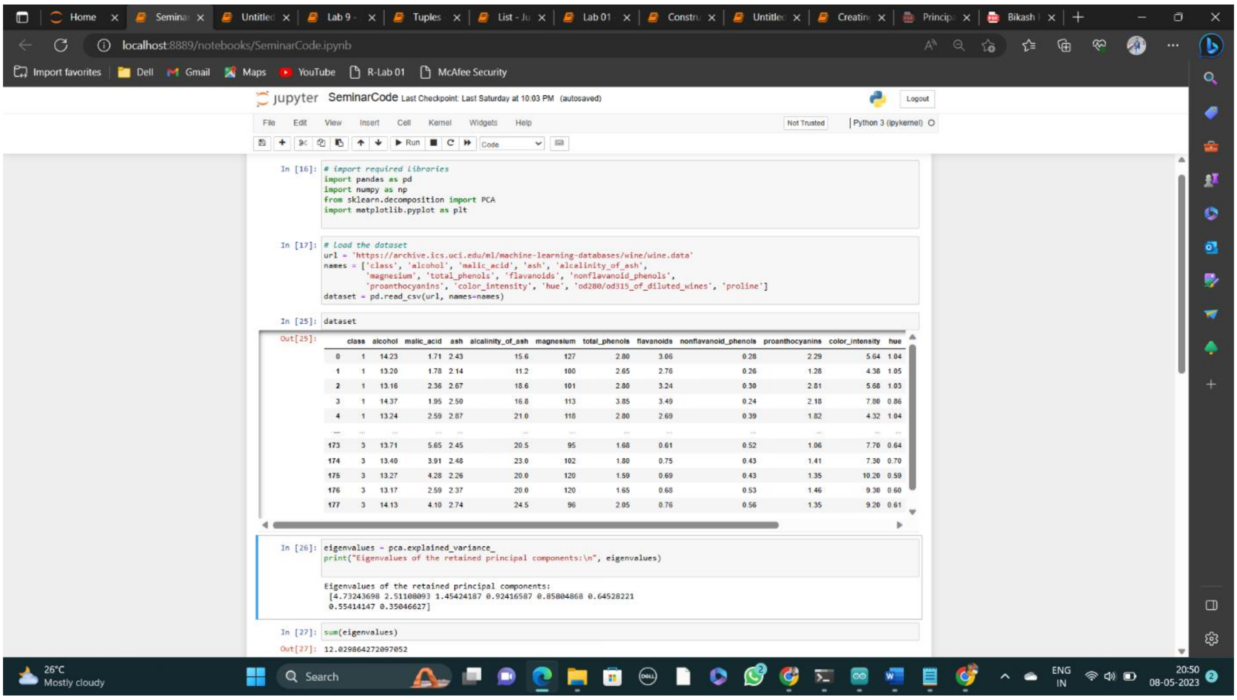
1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium

6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

The target variable is the class of wine, which indicates whether the wine is from one of three different cultivars in the region.

The Wine dataset is often used for multiclass classification tasks, where the goal is to predict the type of wine based on the given features. It is a challenging dataset due to its high dimensionality and small sample size, but it is a valuable dataset for testing the performance of machine learning algorithms, especially those used for wine quality and classification.

Code:



jupyter SeminarCode Last Checkpoint: Last Saturday at 10:03 PM (autosaved)

In [16]:
```
# import required libraries
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
import matplotlib.pyplot as plt
```

In [17]:
```
# Load the dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/wine/wine.data'
names = ['class', 'alcohol', 'malic_acid', 'ash', 'alcalinity_of_ash',
         'magnesium', 'total_phenols', 'flavanoids', 'nonflavanoid_phenols',
         'proanthocyanins', 'color_intensity', 'hue', 'od280/od315_of_diluted_wines', 'proline']
dataset = pd.read_csv(url, names=names)
```

In [25]:
```
dataset
```

Out[25]:

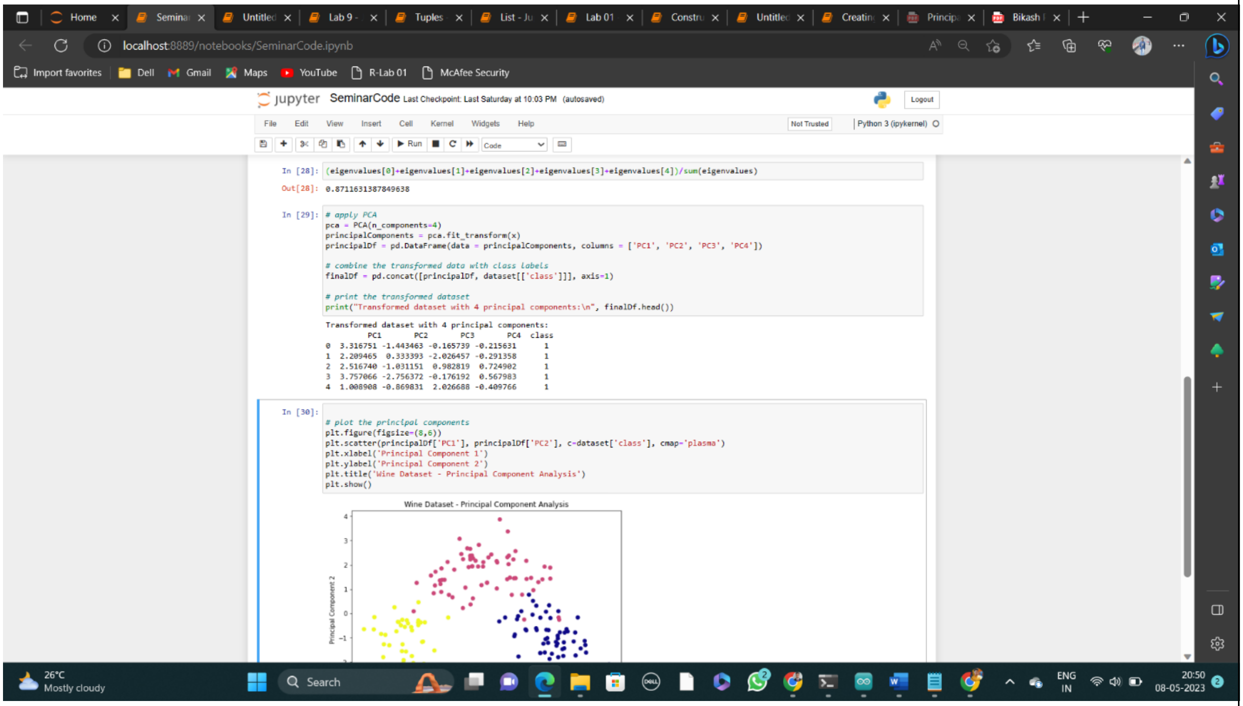
	class	alcohol	malic_acid	ash	alcalinity_of_ash	magnesium	total_phenols	flavanoids	nonflavanoid_phenols	proanthocyanins	color_intensity	hue
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04
...	...	...	...	...	...	...	...	...	...	...	...	...
173	3	13.71	5.65	2.45	20.5	95	1.68	0.61	0.52	1.06	7.70	0.64
174	3	13.40	3.91	2.48	23.0	102	1.80	0.75	0.43	1.41	7.30	0.70
175	3	13.27	4.28	2.26	20.0	120	1.59	0.69	0.43	1.35	10.20	0.59
176	3	13.17	2.59	2.37	20.0	120	1.65	0.68	0.53	1.46	9.30	0.60
177	3	14.13	4.10	2.74	24.5	96	2.05	0.76	0.56	1.35	9.20	0.61

In [26]:
```
eigenvalues = pca.explained_variance_
print("Eigenvalues of the retained principal components:\n", eigenvalues)
```

```
Eigenvalues of the retained principal components:
 [4.73243698 2.51108093 1.45424187 0.92416587 0.85804868 0.64528221
 0.55414147 0.35046627]
```

In [27]:
```
sum(eigenvalues)
```

Out[27]:
```
12.029864272097052
```



jupyter SeminarCode Last Checkpoint: Last Saturday at 10:03 PM (autosaved)

In [28]:
```
(eigenvalues[0]+eigenvalues[1]+eigenvalues[2]+eigenvalues[3]+eigenvalues[4])/sum(eigenvalues)
```

Out[28]:
```
0.8711631387849638
```

In [29]:
```
# apply PCA
pca = PCA(n_components=4)
principalComponents = pca.fit_transform(x)
principalDf = pd.DataFrame(data = principalComponents, columns = ['PC1', 'PC2', 'PC3', 'PC4'])

# combine the transformed data with class labels
finalDf = pd.concat([principalDf, dataset[['class']]], axis=1)

# print the transformed dataset
print("Transformed dataset with 4 principal components:\n", finalDf.head())
```

```
Transformed dataset with 4 principal components:
         PC1       PC2       PC3       PC4  class
0  3.316751 -1.443463 -0.165739 -0.215631      1
1  2.209465  0.333393 -2.026457 -0.291358      1
2  2.516740 -1.031151  0.982819  0.724902      1
3  3.757066 -2.756372 -0.176192  0.567983      1
4  1.008908 -0.869831  2.026688 -0.409766      1
```

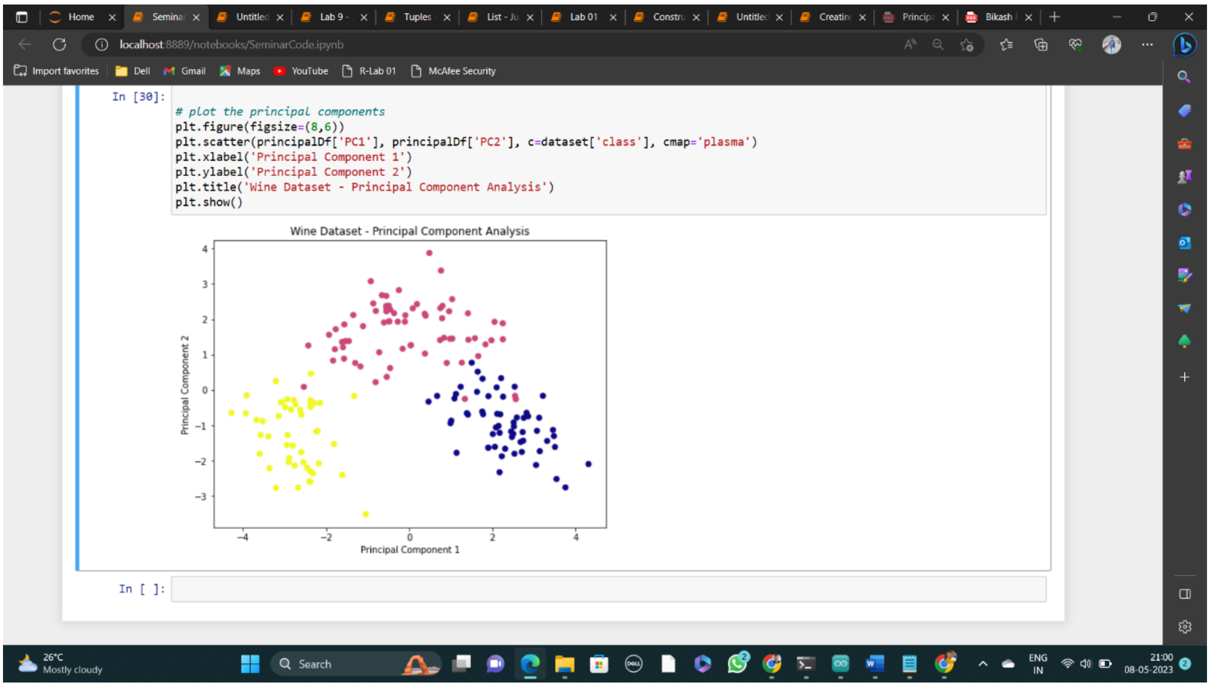
In [30]:
```

# plot the principal components
plt.figure(figsize=(8,6))
plt.scatter(principalDf['PC1'], principalDf['PC2'], c=dataset['class'], cmap='plasma')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Wine Dataset - Principal Component Analysis')
plt.show()
```

Results table:

Method	Accuracy
Original dataset	0.907
PCA (2 components)	0.889

Result visualization:



```
In [30]:
# plot the principal components
plt.figure(figsize=(8,6))
plt.scatter(principalDf['PC1'], principalDf['PC2'], c=dataset['class'], cmap='plasma')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Wine Dataset - Principal Component Analysis')
plt.show()
```

localhost:8889/notebooks/SeminarCode.ipynb
Wine Dataset - Principal Component Analysis
Principal Component 1
Principal Component 2
4
3
2
1
0
−1
−2
−3
−4
−2
0
2
4
In [ ]:

Conclusion

In conclusion, PCA is a powerful technique for dimensionality reduction and data compression. It can be applied to many fields, and it has many advantages over other techniques. However, it also has some limitations, and it may not always provide interpretable results. By understanding the concepts behind PCA, we can better analyze and interpret complex datasets . Loss of information: PCA can reduce the dimensionality of data by projecting it onto a lower-dimensional space, but this can lead to a loss of information. The reduced dimensions may not contain all the information present in the original data, which can lead to a loss of accuracy in downstream analyses.

Computational complexity: The computational complexity of PCA can be high, especially for large datasets. Calculating the eigenvectors and eigenvalues can be time-consuming and memory-intensive,[1] which can limit the practicality of using PCA for very large datasets.

Sensitivity to scaling: PCA is sensitive to the scaling of the data. If the data is not scaled properly before applying PCA, the resulting principal components may be biased towards the features with larger variances. This can result in misleading results and misinterpretations.

Difficulty in handling missing values: PCA may not handle missing data well, and it may be necessary to impute missing values before applying PCA. This can introduce additional bias and noise into the results.

It's important to note that while PCA has its limitations, it can still be a very useful technique for dimension reduction and can be very effective in certain contexts.

- [(Multivariate Statistical Data Analysis- Principal Component Analysis (PCA))]

Reference:

[1] Multivariate Statistical Data Analysis- Principal Component Analysis (PCA) Sidharth Prasad Mishra* , Uttam Sarkar, Subhash Taraphder, Sanjay Datta, Devi Prasanna Swain1 , Reshma Saikhom, Sasmita Panda2 and Menalsh Laishram3.

.

FAQs:

What is the purpose of PCA?

The main purpose of PCA is to simplify complex datasets by identifying the most important variables and creating new variables that capture the most variability in the data. This can make it easier to interpret the data and to perform further analysis, such as clustering or regression.

How does PCA work?

PCA works by finding the direction of maximum variation in the data and projecting the data onto that direction. The first principal component is the linear combination of variables that maximizes the amount of variation captured. The second principal component is the linear combination of variables that maximizes the remaining variation, subject to being orthogonal to the first principal component, and so on for the remaining components.

What are the applications of PCA?

PCA has many applications in various fields, including image and signal processing, finance, marketing, genetics, and psychology. It is often used for data exploration, visualization, and dimensionality reduction.

What are the advantages of using PCA?

The advantages of using PCA include: (1) it can simplify complex datasets by identifying the most important variables and creating new variables that capture the most variability in the data, (2) it can reduce the computational burden by reducing the number of variables, (3) it can help visualize high-dimensional data by reducing the number of variables to two or three, and (4) it can reduce the amount of memory and storage required to store the data.

What are the limitations of using PCA?

The limitations of using PCA include: (1) it assumes that the data is linearly related, (2) it can be sensitive to outliers, (3) it may not always be easy to interpret the principal components, and (4) it can sometimes be difficult to choose the number of principal components to retain.