Principal Component Analysis

Principal Component Analysis is an unsupervised learning algorithm that is used for the dimensionality reduction in [machine learning](https://www.javatpoint.com/machine-learning). It is a statistical process that converts the observations of correlated features into a set of linearly uncorrelated features with the help of orthogonal transformation. These new transformed features are called the **Principal Components**. It is one of the popular tools that is used for exploratory data analysis and predictive modeling. It is a technique to draw strong patterns from the given dataset by reducing the variances.

PCA generally tries to find the lower-dimensional surface to project the high-dimensional data.

PCA works by considering the variance of each attribute because the high attribute shows the good split between the classes, and hence it reduces the dimensionality. Some real-world applications of PCA are ***image processing, movie recommendation system, optimizing the power allocation in various communication channels.*** It is a feature extraction technique, so it contains the important variables and drops the least important variable.

The PCA algorithm is based on some mathematical concepts such as:

Variance and Covariance

Eigen values and Eigen factors

Some common terms used in PCA algorithm:

* **Dimensionality:** It is the number of features or variables present in the given dataset. More easily, it is the number of columns present in the dataset.
* **Correlation:** It signifies that how strongly two variables are related to each other. Such as if one changes, the other variable also gets changed. The correlation value ranges from -1 to +1. Here, -1 occurs if variables are inversely proportional to each other, and +1 indicates that variables are directly proportional to each other.
* **Orthogonal:** It defines that variables are not correlated to each other, and hence the correlation between the pair of variables is zero.
* **Eigenvectors:** If there is a square matrix M, and a nonzero vector v is given. Then v will be an eigenvector if Av is the scalar multiple of v.
* **Covariance Matrix:** A matrix containing the covariance between the pair of variables is called the Covariance Matrix.

Principal Components in PCA

As described above, the transformed new features or the output of PCA are the Principal Components. The number of these PCs are either equal to or less than the original features present in the dataset. Some properties of these principal components are given below:

* The principal component must be the linear combination of the original features.
* These components are orthogonal, i.e., the correlation between a pair of variables is zero.
* The importance of each component decreases when going from 1 to n, it means the 1 PC has the most importance, and n PC will have the least importance.

Steps for PCA algorithm

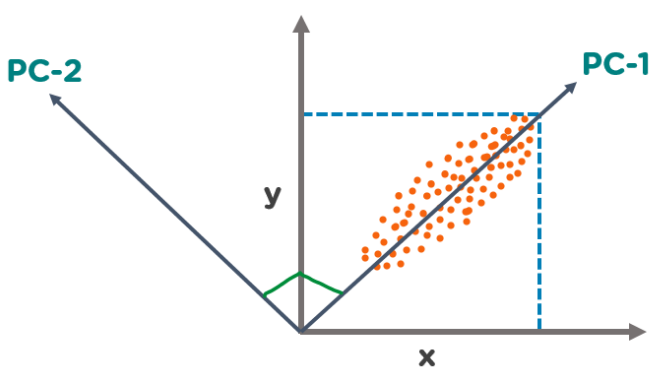
1. **Getting the dataset**

Firstly, we need to take the input dataset and divide it into two subparts X and Y, where X is the training set, and Y is the validation set.

1. **Representing data into a structure** Now we will represent our dataset into a structure. Such as we will represent the two-dimensional matrix of independent variable X. Here each row corresponds to the data items, and the column corresponds to the Features. The number of columns is the dimensions of the dataset.
2. **Standardizing the data** In this step, we will standardize our dataset. Such as in a particular column, the features with high variance are more important compared to the features with lower variance. If the importance of features is independent of the variance of the feature, then we will divide each data item in a column with the standard deviation of the column. Here we will name the matrix as Z.
3. **Calculating the Covariance of Z** To calculate the covariance of Z, we will take the matrix Z, and will transpose it. After transposing, we will multiply it by Z. The output matrix will be the Covariance matrix of Z.
4. **Calculating the EigenValues and EigenVectors** Now we need to calculate the eigenvalues and eigenvectors for the resultant covariance matrix Z. Eigenvectors or the covariance matrix are the directions of the axes with high information. And the coefficients of these eigenvectors are defined as the eigenvalues.
5. **Sorting the Eigen Vectors**In this step, we will take all the eigenvalues and will sort them in decreasing order, which means from largest to smallest. And simultaneously sort the eigenvectors accordingly in matrix P of eigenvalues. The resultant matrix will be named as P\*.
6. **Calculating the new features Or Principal Components**Here we will calculate the new features. To do this, we will multiply the P\* matrix to the Z. In the resultant matrix Z\*, each observation is the linear combination of original features. Each column of the Z\* matrices are independent of each other.
7. **Remove less or unimportant features from the new dataset.**  
   The new feature set has occurred, so we will decide here what to keep and what to remove. It means, we will only keep the relevant or important features in the new dataset, and unimportant features will be removed.

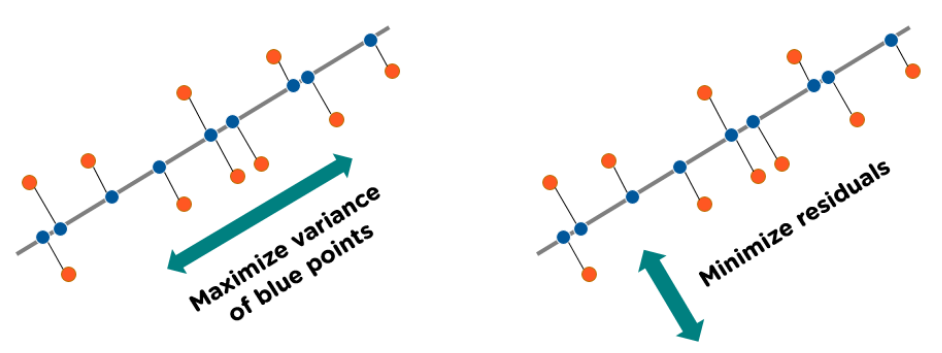
Applications of Principal Component Analysis

* PCA is mainly used as the dimensionality reduction technique in various AI applications such **as computer vision, image compression, etc.**
* It can also be used for finding hidden patterns if data has high dimensions. Some fields where PCA is used are Finance, data mining, Psychology, etc.
* PCA is used to visualize multidimensional data.
* It is used to reduce the number of dimensions in healthcare data.
* PCA can help resize an image.
* It can be used in finance to analyze stock data and forecast returns.
* PCA helps to find patterns in the high-dimensional datasets.



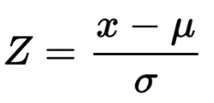
In the above figure, we have several points plotted on a 2-D plane. There are two principal components. PC1 is the primary principal component that explains the maximum variance in the data. PC2 is another principal component that is orthogonal to PC1.

**How does Principal Component Analysis Work?**



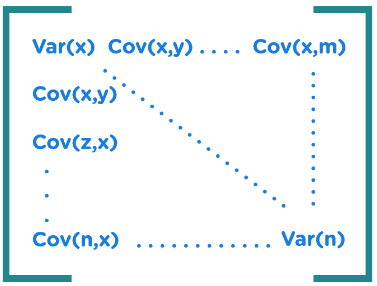
### 1. Normalize the data

Standardize the data before performing PCA. This will ensure that each feature has a mean = 0 and variance = 1.



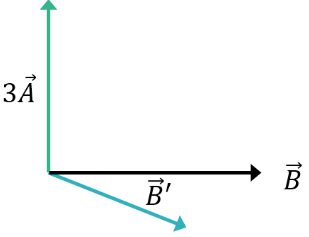
### 2. Build the covariance matrix

Construct a square matrix to express the correlation between two or more features in a multidimensional dataset.



### 3. Find the Eigenvectors and Eigenvalues

Calculate the eigenvectors/unit vectors and eigenvalues. Eigenvalues are scalars by which we multiply the eigenvector of the covariance matrix.



### 4. Sort the eigenvectors in highest to lowest order and select the number of principal components.

Now that you have understood How PCA in Machine Learning works, let’s perform a hands-on demo on PCA with Python.

## Principal Component Analysis Solved Example

Principal component analysis (PCA) is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components.

In this article, I will discuss how to find the principal components with a simple solved numerical example.

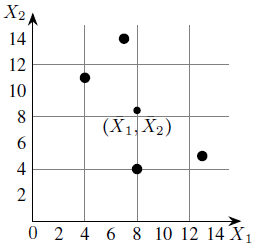
### Problem definition

Given the data in Table, reduce the dimension from 2 to 1 using the Principal Component Analysis (PCA) algorithm.

| **Feature** | **Example 1** | **Example 2** | **Example 3** | **Example 4** |
| --- | --- | --- | --- | --- |
| **X1** | 4 | 8 | 13 | 7 |
| **X2** | 11 | 4 | 5 | 14 |

**Step 1: Calculate Mean**

The figure shows the scatter plot of the given data points.

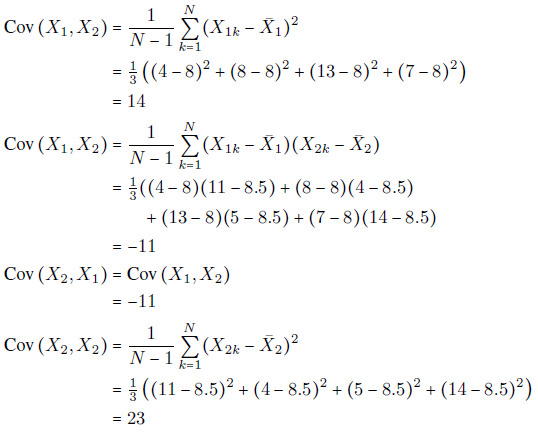


Calculate the mean of X1 and X2 as shown below.

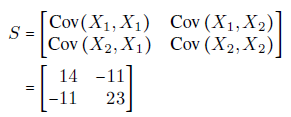
Mean of given data set

**Step 2: Calculation of the covariance matrix.**

The covariances are calculated as follows:

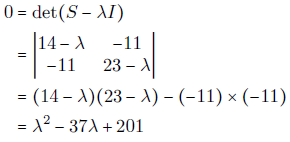


The covariance matrix is,

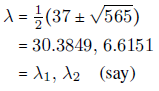


**Step 3: Eigenvalues of the covariance matrix**

The characteristic equation of the covariance matrix is,



Solving the characteristic equation we get,



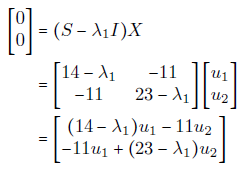
**Step 4: Computation of the eigenvectors**

To find the first principal components, we need only compute the eigenvector corresponding to the largest eigenvalue. In the present example, the largest eigenvalue is λ1 and so we compute the eigenvector corresponding to λ1.

The eigenvector corresponding to λ = λ1 is a vector

https://www.vtupulse.com/wp-content/uploads/2022/05/image-21.png

satisfying the following equation:



This is equivalent to the following two equations:

https://www.vtupulse.com/wp-content/uploads/2022/05/image-23.png

Using the theory of systems of linear equations, we note that these equations are not independent and solutions are given by,

https://www.vtupulse.com/wp-content/uploads/2022/05/image-24.png

that is,

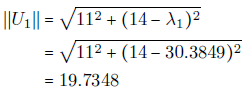
https://www.vtupulse.com/wp-content/uploads/2022/05/image-25.png

where t is any real number.

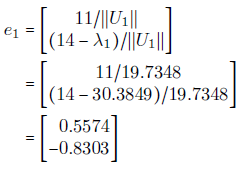
Taking **t = 1**, we get an eigenvector corresponding to λ1 as

https://www.vtupulse.com/wp-content/uploads/2022/05/image-26.png

To find a unit eigenvector, we compute the length of X1 which is given by,



Therefore, a unit eigenvector corresponding to λ1 is



By carrying out similar computations, the unit eigenvector e2 corresponding to the eigenvalue λ= λ2 can be shown to be,

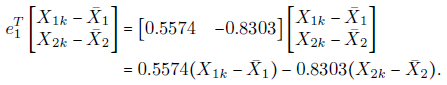
https://www.vtupulse.com/wp-content/uploads/2022/05/image-29.png

**Step 5: Computation of first principal components**

let,

https://www.vtupulse.com/wp-content/uploads/2022/05/image-30.png

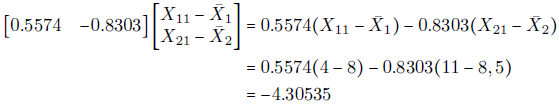
be the kth sample in the above Table (dataset). The first principal component of this example is given by (here “T” denotes the transpose of the matrix)



For example, the first principal component corresponding to the first example

https://www.vtupulse.com/wp-content/uploads/2022/05/image-32.png

is calculated as follows:



The results of the calculations are summarized in the below Table.

[**See also  K-Means and EM Algorithm in Python**](https://www.vtupulse.com/machine-learning/k-means-and-em-algorithm-in-python/)

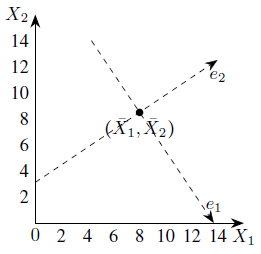
| **X1** | 4 | 8 | 13 | 7 |
| --- | --- | --- | --- | --- |
| **X2** | 11 | 4 | 5 | 14 |
| **First Principle Components** | -4.3052 | 3.7361 | 5.6928 | -5.1238 |

**Step 6: Geometrical meaning of first principal components**

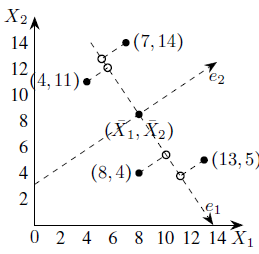
First, we shift the origin to the “center”

https://www.vtupulse.com/wp-content/uploads/2022/05/image-34.png

and then change the directions of coordinate axes to the directions of the eigenvectors e1 and e2.

The coordinate system for principal components

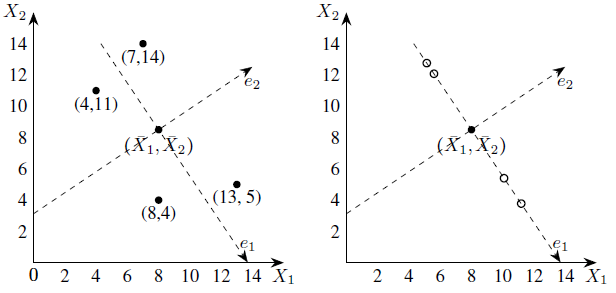
Next, we drop perpendiculars from the given data points to the e1-axis (see below Figure).



Projections of data points on the axis of the first principal component

The first principal components are the e1-coordinates of the feet of perpendiculars, that is, the projections on the e1-axis. The projections of the data points on the e1-axis may be taken as approximations of the given data points hence we may replace the given data set with these points.

Now, each of these approximations can be unambiguously specified by a single number, namely, the e1-coordinate of  
approximation. Thus the two-dimensional data set can be represented approximately by the following one-dimensional data set.

Geometrical representation of one-dimensional approximation to the data set

**Conclusion**

The principal component analysis is a widely used unsupervised learning method to perform dimensionality reduction. We hope that this article helped you understand what PCA is and the applications of PCA. You looked at the applications of PCA and how it works.

Do you have any questions related to this article on PCA in Machine Learning? If yes, then please feel free to put them in the comments sections. Our team will be happy to solve your queries. Finally, we performed a hands-on demonstration on classifying wine type by using the first two principal components.