**Experiment 4**

**Aim:**

To study and implement Principal Component Analysis (PCA) for dimensionality reduction

**Objectives:**

1. To transform given data using PCA.
2. To reconstruct the original data from principal components.
3. Observe effect of decreasing the number of principal components on the reconstructed data.

**Software platform:**

Matlab

**Theory:**

Principal Component Analysis (PCA) is a classic among the many methods of multivariate data analysis. Invented in 1901 by Karl Pearson the method is mostly used today as a tool in exploratory data analysis and dimension reduction, but also for making predictive models in machine learning.

PCA is a useful statistical technique that has found application in fields such as face recognition and image compression, and is a common technique for finding patterns in data of high dimension.

The main idea behind the principal component analysis is to represent multidimensional data with fewer numbers of variables retaining main features of the data. It is inevitable that by reducing dimensionality some features of the data will be lost. It is hoped that these lost features are comparable with the noise and they do not tell much about underlying population.

It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data.

PCA is a quantitative technique for describing populations with multiple types of measurements for each observation. PCA is a descriptive technique and, as such, has no hypotheses associated with the procedure. It is a technique that allows you to reduce values from several variables into a single variable (data reduction).

PCA is a method to study the structure of the data, with emphasis on determining the patterns of covariance among variables. Thus, PCA is the study of the structure of the variance-covariance matrix.

**Algorithm:**

1. Get two dimensional data { (x1, y1), (x2, y2), ……….(xn, yn)}.
2. Subtract the mean.
3. Calculate the covariance matrix of size. If n=2 matrix size is 2 x 2.
4. Calculate the Eigen values and Eigen vectors of the covariance matrix.
5. Choose the components & form a feature vector given as:

Feature Vector = (eig1, eig2).

For n=2, we get 2 Eigen vectors, each of size 2 given as:

1. Arrange the Eigen Vectors in descending order of Eigen values.
2. Determine *Row Feature Vector by* transposing EV*.*
3. Select the components (i.e. Eigen vectors) and Derive the new data set by,

*Final Data = Row Feature Vector x Mean Subtracted Data*

1. Reconstruct the data for the three cases: using both the components, first component only and second component only.

*Mean Subtracted Data = Row Feature Vector-1 x Final Data*

Reconstruct the data by adding respective mean to the respective rows of *Mean Subtracted Data*

1. Compare the Reconstructed data for the 3 cases.
2. Plot the original data (x vs y), mean subtracted data, transformed data (component1 vs component 2) and reconstructed data for the three cases.

**Conclusion:**

**FAQ’S:**

1. Give significance of PCA in machine learning.
2. What are advantages and disadvantages of PCA technique?
3. Using PCA, reduce the below 2D data to 1D with proper steps and explanation.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| x | 2.5 | 0.5 | 2.2 | 1.9 | 3.1 | 2.3 | 2.0 | 1.0 | 1.5 | 1.2 |
| y | 2.4 | 0.7 | 2.9 | 2.2 | 3.0 | 2.7 | 1.6 | 1.1 | 1.6 | 0.9 |