AIML

Dr. Nitin A. Shelke

Dimensionality Reduction Techniques

- PCA
- Kernel PCA
- LDA

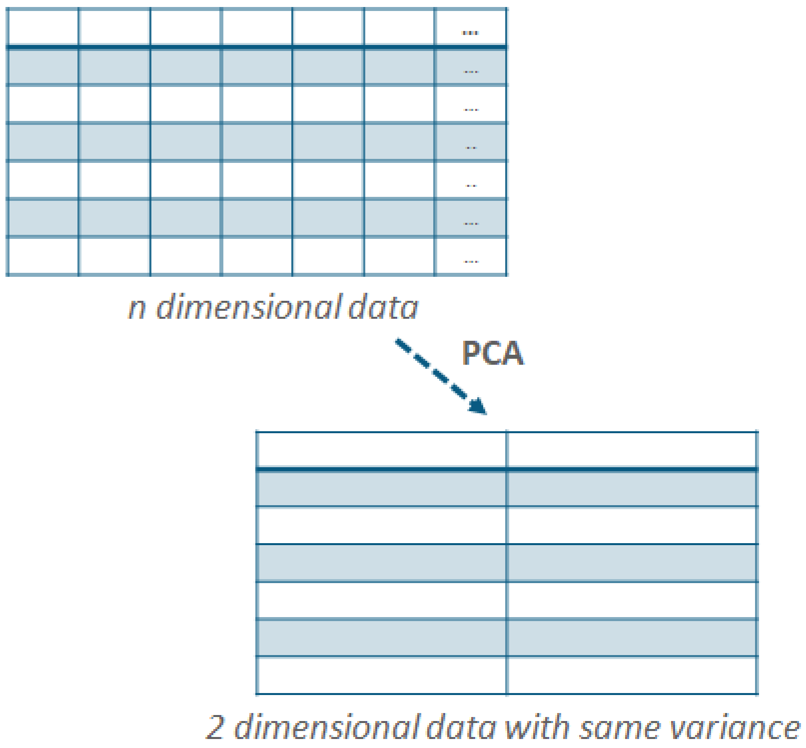
Need of Dimensionality Reduction

- Machine Learning in general works wonders when the dataset provided for training the machine is large and concise.
- However, using a large data set has its own pitfalls. The biggest pitfall is the curse of dimensionality.
- It turns out that in large dimensional datasets, there might be lots of inconsistencies in the features or lots of redundant features in the dataset, which will only increase the computation time and make data processing convoluted.
- To get rid of the curse of dimensionality, a process called dimensionality reduction was introduced.

PCA

- **Principal Component Analysis (PCA)** is a dimensionality reduction technique or algorithm. PCA is defined by a transformation of a high dimensional vector space into a low dimensional space.
- Moreover, PCA is an unsupervised statistical technique used to examine the interrelations among a set of variables.
- The main idea behind PCA is to identify the patterns in the data and correlations among various features in the data set. On finding a strong correlation between different variables, a final decision is made about reducing the dimensions of the data in such a way that the significant data is still retained.

Example


n dimensional data
PCA
2 dimensional data with same variance

Step By Step Computation Of PCA

- The below steps need to be followed to perform dimensionality reduction using PCA:
  - Standardization of the data
  - Computing the covariance matrix
  - Calculating the eigenvalues and eigenvectors
  - Computing the Principal Components (New set of variables)
  - Reducing the dimensions of the data set.

Note: PCA tries to compress as much information as possible in the first PC, the rest in the second, and so on…

Example PCA

Example:

X	Y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

Original dataset

Compute Covariance Matrix i.e. A

Cov(X,X)	Cov(X,Y)
Cov(Y,X)	Cov(Y,Y)

$$Cov(X,Y) = \frac{1}{N-1}\sum_{i=1}^{N}(X_i - \overline{X})(Y_i - \overline{Y})$$

Example PCA

Example:

X	Y	$X_i - \bar{X}$	$(Y_i - \bar{Y})$	$(X_i - \bar{X})(X_i - \bar{X})$	$(Y_i - \bar{Y})(Y_i - \bar{Y})$	$(X_i - \bar{X})(Y_i - \bar{Y})$
2.5	2.4	0.69	0.49	0.4761	0.2401	0.3381
0.5	0.7	-1.31	-1.21	1.7161	1.4641	1.5851
2.2	2.9	0.39	0.99	0.1521	0.9801	0.3861
1.9	2.2	0.09	0.29	0.0081	0.0841	0.0261
3.1	3	1.29	1.09	1.6641	1.1881	1.4061
2.3	2.7	0.49	0.79	0.2401	0.6241	0.3871
2	1.6	0.19	-0.31	0.0361	0.0961	-0.0589
1	1.1	-0.81	-0.81	0.6561	0.6561	0.6561
1.5	1.6	-0.31	-0.31	0.0961	0.0961	0.0961
1.1	0.9	-0.71	-1.01	0.5041	1.0201	0.7171

$\bar{X}$=1.81

$\bar{Y} = 1.91$

Cov(X,X) = 0.6165

Cov(Y,Y) = 0.7165

Cov(X,Y)= Cov(Y,X) = 0.6154

Example PCA

Example:

$$\begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} \xrightarrow[\det(A-\lambda I)=0]{\text{Compute eigenvalues}} \begin{bmatrix} 0.6165 & 0.6154 \\ 0.6154 & 0.7165 \end{bmatrix} - \lambda \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0.6165-\lambda & 0.6154 \\ 0.6154 & 0.7165-\lambda \end{bmatrix}$$

Find determinate by equating to zero

$$\lambda_1 = 1.284028$$

$$\lambda_2 = 0.049083$$

Example PCA

Example:

0.6165-λ_1	0.6154
0.6154	0.7165-λ_1

0.6165-λ_2	0.6154
0.6154	0.7165-λ_2

Compute eigenvectors

$$[\boldsymbol{A} - \boldsymbol{\lambda I}]\,\boldsymbol{X} = \boldsymbol{0}$$

$$\lambda_1 = 1.284028$$

$$\lambda_2 = 0.049083$$

V_1

$$\begin{bmatrix} -0.6675 & 0.6154 \\ 0.6154 & -0.5675 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

V_2

$$\begin{bmatrix} 0.5674 & 0.6154 \\ 0.6154 & 0.6674 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 0$$

Example PCA

Example:

$V_1 =$

0.67787
0.73517

First Principal Component (PC1)

$V_2 =$

-0.73517
0.67787

Second Principal Component (PC2)

"vector corresponding to highest eigenvalue of considered as PC1 followed by other component as per their eigenvalue."

- To calculate the **percentage of information** explained by PC1 and PC2, divide each component by sum of eigenvalues

PC1 = 96%

PC2 = 4%

divide the eigenvalue of each component by the sum of eigenvalues.

Kernel PCA

- PCA is a linear method. That is it can only be applied to datasets which are linearly separable.
- It does an excellent job for datasets, which are linearly separable. But, if we use it to non-linear datasets, we might get a result which may not be the optimal dimensionality reduction.
- Kernel PCA uses a kernel function to project dataset into a higher dimensional feature space, where it is linearly separable. It is similar to the idea of Support Vector Machines.

Linear Discriminant Analysis (LDA)

- Both PCA and LDA are linear transformation techniques used for dimensionality reduction.
- However, PCA is an unsupervised while LDA is a supervised dimensionality reduction technique.
- PCA has no concern with the class labels. In simple words, PCA summarizes the feature set without relying on the output.
- Unlike PCA, LDA tries to reduce dimensions of the feature set while retaining the information that discriminates output classes.
- It is used for modeling differences in groups i.e. separating two or more classes.
- It is used to project the features in higher dimension space into a lower dimension space.