EasyChair Preprint

№ 4384

Principal Component Analysis

Arun Kumar and Pankaj Kumar Saini

October 13, 2020

PRINCIPAL COMPONENT ANALYSIS

Arun Kumar and Pankaj Kumar Saini
Mit2020116@iiita.ac.in Mit2020117@iiita.ac.in
Indian Institute of Information Technology, Prayagraj

Abstract

Principal Component analysis or aka PCA is one of the most important dimensionality reduction technique out there. This paper is devoted towards why we need PCA, what are the steps to be taken and what are the benefits of using Principal component analysis. While in Data Exploratory Analysis we need to reduce the dimension in such a way that the maximum of what we need is to be captured.

Keyword: **-** Principal component analysis, dimension reduction, data exploratory analysis

Introduction

Principal component analysis was invented in 1901 by Karl Pearson, as an analogue of the principal axis theorem in mechanics; it was later independently developed and named by Harold Hotelling in the 1930's. [1] Depending upon the field of the application, it is also named the discrete Karhunen-Lo`eve transform (KTL) in signal processing, the Hotelling transform in multivariate quality control, proper orthogonal decomposition (POD) in mechanical engineering, singular value decomposition (SVD) of X (Golub and Van Loan 1983), eigenvalue decomposition (EVD) of $\mathbf{X}^T\mathbf{X}$ in linear algebra, factor analysis (for a discussion of the difference between PCA and factor analysis) Eckart-Young theorem (Harman, 1960), or empirical orthogonal functions (EOF) in meteorological science, empirical eigenfunction decomposition
(Sirovich, 1987), empirical component analysis (Lorenz, 1956), quasiharmonic modes (Brooks et al. 1988), spectral decomposition in noise and vibration, and empirical modal analysis in structural dynamics. Principal Component analysis or aka PCA is one of the most important dimension rotationality technique out there. Principal component analysis (PCA) is the process of computing the principal components and using them to perform a change in basis on the data, sometimes using only the first few principal components and ignoring the rest.

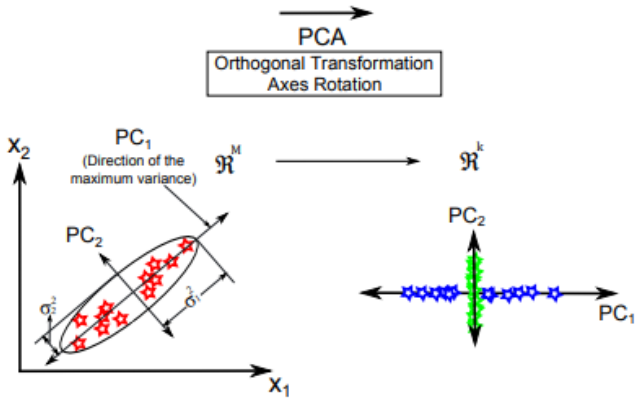

Figure 1: Example of the two-dimensional data (x1, x2). The original data are on the left with the original coordinate, i.e. x1 and x2, the variance of each variable is graphically represented and the direction of the maximum variance, i.e. the principal component P C1, is shown; on the right the original data are projected on the first (blue stars) and second (green stars) principal components.

Principal component analysis is used in exploratory data analysis for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data. Principal component analysis is the simplest of the true eigenvector-based multivariate analyses and is closely related to factor analysis. Factor analysis typically incorporates more domain specific assumption and solves eigenvectors of a slightly different matrix.

The steps to be taken for PCA

1. **Standardization of data**

   Before proceeding with PCA, we need to perform the standardization of the data.

   Performing standardization is a crucial step because the original variables may have different scales. We need to bring them to a similar range to get reasonable covariance analysis.

2. **Covariance Matrix Method**:
In this method, there are two main steps to calculate the PCs of the PCA space. First, the covariance matrix of the data matrix (X) is calculated. Second, the eigenvalues and eigenvectors of the covariance matrix are calculated. Figure 2 illustrates the visualized steps of calculating the PCs using the covariance matrix method.

Calculating Covariance Matrix (Σ) & Block Diagram:
The variance of any variable measures the deviation of that variable from its mean value and it is defined as follows, σ^2(x) = V ar(x) = E((x − μ)^2) = E{x2} − (E{x})^2, where μ represents the mean of the variable x, and E(x) represents the expected value of x. The covariance matrix is used when the number of variables more than one and it is defined as follows, Σij = E{xixj} − E{xi}E{xj} = E[(xi − μi)(xj − μj)]. As shown in Figure 2, step(A), after calculating the mean of each variable in the data matrix, the mean-centring data are calculated by subtracting the mean (μ ∈ R(M×1)) from each sample as follows, D = {d1, d2, . . . , dN } = {x1 − μ, x2 − μ, . . . , xN − μ} .The covariance matrix is then calculated as follows, Σ = DDT(see Figure 2, step (B)).

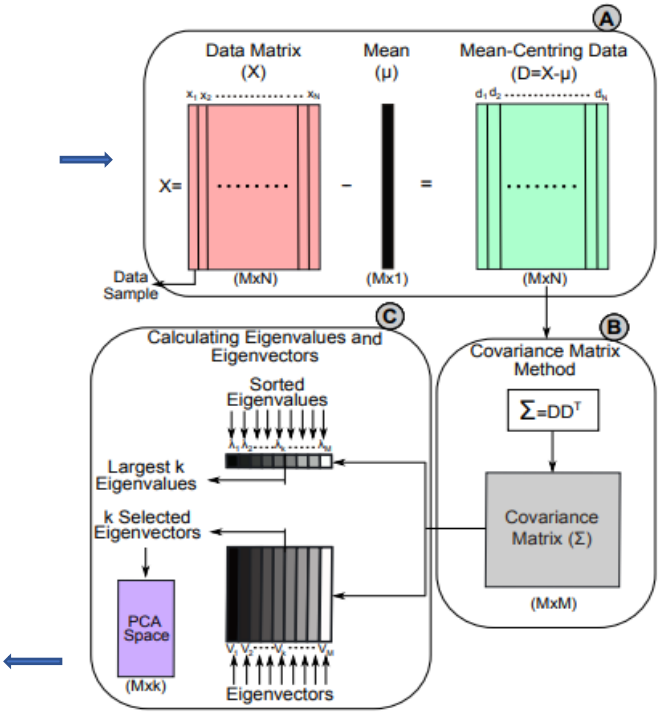

Figure 2: Visualized steps to calculate the PCA space using the covariance matrix method & Block Diagram

xi, i = 1, . . . , M, while the off-diagonal entries represent the covariance between two different variables as shown in Equation (1). A positive value in covariance matrix means a positive correlation between the two variables, while the negative value indicates a negative correlation and zero value indicate that the two variables are uncorrelated or statistically independent.

$$\begin{pmatrix} Var(x_1,x_1) & Cov(x_1,x_2) & \dots & Cov(x_1,x_M) \\ Cov(x_2,x_1) & Var(x_2,x_2) & \dots & Cov(x_2,x_M) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(x_M,x_1) & Cov(x_M,x_2) & & Var(x_M,x_M) \end{pmatrix} \quad (1)$$

3. **Calculating Eigenvalues (λ) and Eigenvectors (V):**
The covariance matrix is solved by calculating the eigenvalues (λ) and eigenvectors (V) as follows:
V Σ = λV (2)
where V and λ represent the eigenvectors and eigenvalues of the covariance matrix, respectively.

The eigenvalues are scalar values, while the eigenvectors are non-zero vectors, which represent the principal components, i.e. each eigenvector represents one principal component. The eigenvectors represent the directions of the PCA space, and the corresponding eigenvalues represent the scaling factor, length, magnitude, or the robustness of the eigenvectors. The eigenvector with the highest eigenvalue represents the first principal component and it has the maximum variance as shown in Figure 1 (Hyvärinen, 1970). The eigenvalues may be equal when the PCs have equal variances and hence all the eigenvectors are the same and we cannot decide which eigenvectors are used to construct the PCA space.

Pseudo Algorithm:
Algorithm 1: Calculating PCs using Covariance Matrix Method
1: Given a data matrix (X = [x1, x2, . . ., xN]), where N represents the total number of samples and xi represents the i th sample.

2: Compute the mean of all samples as follows:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$$

3: Subtract the mean from all samples as follows:

$$D = \{d_1, d_2, \ldots, d_N\} = \sum_{i=1}^{N} x_i - \mu$$

4: Compute the covariance matrix as follows:

$$\Sigma = \frac{1}{N-1} D \times D^T$$

5: Compute the eigenvectors V and eigenvalues λ of the covariance matrix (Σ).
6: Sort eigenvectors according to their corresponding eigenvalues.
7: Select the eigenvectors that have the largest eigenvalues W = {v1, . . ., vk}. The selected eigenvectors (W) represent the projection space of PCA.
8: All samples are projected on the lower dimensional space of PCA (W) as follows,
Y = W^T D.

4. **Sorting the eigen values in decreasing order**
After completing Eigendecomposition, we need to arrange them in decreasing order so that we can select the higher values which captures most of the features.[9]

5. **Selecting the number of principal components**
The first principal component will capture most of the variance from the original variables and the second principal component captures the second highest variance and so on…[9]

6. **Selecting the principal components**
PCA can be done only on the numerical variables. If you have categorical data, then you need to convert into numerical features before applying PCA.[9]

There are some Limitations as well in PCA
The result of PCA depends on the scaling of the variables. This can be cured by scaling each feature by its standard deviation, so that one ends up with dimensionless features with unital variance.
The applicability of PCA as described above is limited by certain assumptions made in its derivation. In particular PCA captures linear correlation between the features but fails when this assumption is violated.
Another limitation is the mean-removal process before constructing the covariance matrix for PCA. As an alternative method, non-negative matrix factorization focusing only on the non-negative elements in the matrices.

Time Complexity:

The algorithm has two computationally intensive steps:

- Computing the covariance matrix
- Computing the eigenvalue decomposition of the covariance matrix.

Assuming your dataset is 'X' in **R^{nxp}** where n: number of samples, p: dimensions of a sample, you are interested in the eigenanalysis of **X^TX** which is the main computational cost of PCA.
Now matrices **X^TX** in **R^{pxp}** and **XX^T** in **R^{nxn}** have the same **min(n, p)** non negative eigenvalues and eigenvectors. Assuming p less than n you can solve the eigenanalysis in **O(p^3).** If p greater than n (for example in computer vision in many cases the dimensionality of sample -number of pixels- is greater than the number of samples available) you can perform eigenanalysis in **O(n^3)** time.
In any case you can get the eigenvectors of one matrix from the eigenvalues and eigenvectors of the other matrix and do that in **O(min(p, n)^3)** time.

Application in Real World:

PCA is predominantly used as a dimensionality reduction technique in domains like facial recognition, computer vision and image compression. It is also used for finding patterns in data of high dimension in the field of finance, data mining, bioinformatics, psychology, etc.
It is one of the most widely used dimension reduction techniques to transform larger dataset into smaller dataset by identifying the correlation and patterns with preserving most of the valuable information.
It is used to overcome the features redundancy in the dataset. Also, it aims to capture valuable information explaining high variance which results in providing the best accuracy. It decreases the complexity of the model and increases computational efficiency.[10]

Conclusion:

PCA is a simple but effective method to reduce dimensions of linearly distributed data. Image data compression using PCA shows an efficient way to store huge imagery data with reduced dimensions and without loss of generality. However, in general situation prior knowledge of the data shape is strongly required to attain satisfying PCA result. If the given data set is nonlinear or multimodal distribution, PCA fails to provide meaningful data reduction.

After applying PCA, you'll have a set of Principal Components, ranked in descending order of how much they contribute to describing patterns in the data. In statistical parlance, they are ranked according to how much variance they explain.

The first principal component is the most important at describing variance in the data. The remaining principal components are less critical expressing the variability of patterns in the data.

Behind the scenes, Principal Component Analysis uses statistical tools to identify noise and redundancy in the dataset. It uses the covariance matrix to analyze:

Variance of each feature. It will show if a feature is relevant or pure noise. Strength of linear relationship between pairs of features. This helps to spot redundant features.

So, at the end of the day, what PCA will produce a set of principal components which:

- Reduce noise, by maximizing feature variance.
- Reduce redundancy, by minimizing the covariance between pairs of features.

The basis of PCA is the covariance matrix and, in practice, there are two approaches to identify the principal components:

- Calculate the eigenvectors of the covariance matrix.
- Calculate the Singular value Decomposition of the covariance matrix.

References:

[1] en.wikipedia.org/wiki/Principal_component_analysis

[9] towardsdatascience.com/all-you-need-to-know-about-pca-technique-in-machine-learning-443b0c2be9a1

[10] towardsdatascience.com/principal-component-analysis-algorithm-in-real-life-discovering-patterns-in-a-real-estate-dataset-18134c57ffe7#:~:text=One%20of%20the%20most%20popular,its%20core%20patterns%20and%20characteristics.

[3] International Journal of Engineering Research & Technology (IJERT)
http://www.ijert.org ISSN: 2278-0181

[4] Eastern Mediterranean University January 2014 Gazimağusa

[5] Kadappagari Vijaya Kumar and Atul Negi, "A review of principal component analysis methods",
submitted to Journal of Pattern Recognition Research, on Jan. 13th 2009.

APPENDIX

IMPLEMENTATION :

```
Import numpy as np
def pca( number_components,data):
if not 0<= number_components<=data.Shape[1]:
raise ValueError('The number of features are less than the number of components')
#calculate the covariance matrix
cov_matrix=np.cov(data.T);
#calculate the eigen things
eig_vals,eig_vecs=np.linalg.eigh(cov_matrix);
#codes are the same
eig_pairs=[(np.abs(eig_vals[i]),eig_vecs[:,i]) for i in range(len(eig_vals))]
#Sorting All of them
eig_pairs.sort(key=lamndax:x[0],reverse=True)
#Getting the selected vectors in a form of matrix
final=[eig_pairs[i][1].reshape(data.shape[1]) for i in range (number_components)]
#Creating the Projection Matrix, multiplying by identify in an addons
projection_matrix=np.hstack((final));
#transforming the data
y=data.dot(projection_matrix);
Return y;
```