Seminar Topic

Principal Component Analysis (PCA)

Name – Ranjan Sarkar
Roll – 21MA40022
2-Year MSc, 2nd Year
Department of Mathematics, IIT Kharagpur
E-mail Id – *ranjansarkar717@kgpian.iitkgp.ac.in*

Date – September 14, 2022

Indian Institute of Technology
Kharagpur

Content

1. Introduction
2. PCA Method
   i. Geometric Interpretation
   ii. Mathematical Formulation
3. Applications
4. References

Introduction

- PCA was invented in ***1901*** by British Mathematician and Biostatistician ***Karl Pearson***, as an analogue of the principal axis theorem in mechanics; it was later independently developed and named by Harold Hotelling in the 1930s.
- This method becomes too popular nowadays in Machine Learning for the sudden growth of the data. To reduce the dimension of higher-dimensional data and to visualize it, we use it.

What is Principal Component?

- The principal components of a collection of points in a real coordinate space are a sequence of p unit vectors, where the i^{th} vector is the direction of a line that best fits the data while being orthogonal to the first i-1 vectors.
- Here, a best-fitting line is defined as one that minimizes the average squared perpendicular distance from the points to the line.
- These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated.

Principal Component Analysis (PCA) is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. **PCA** is a statistical procedure which uses an orthogonal transformation to reduce the dimension of a data (i.e., no. of attributes / number of columns in a given data matrix)

PCA Method

Description

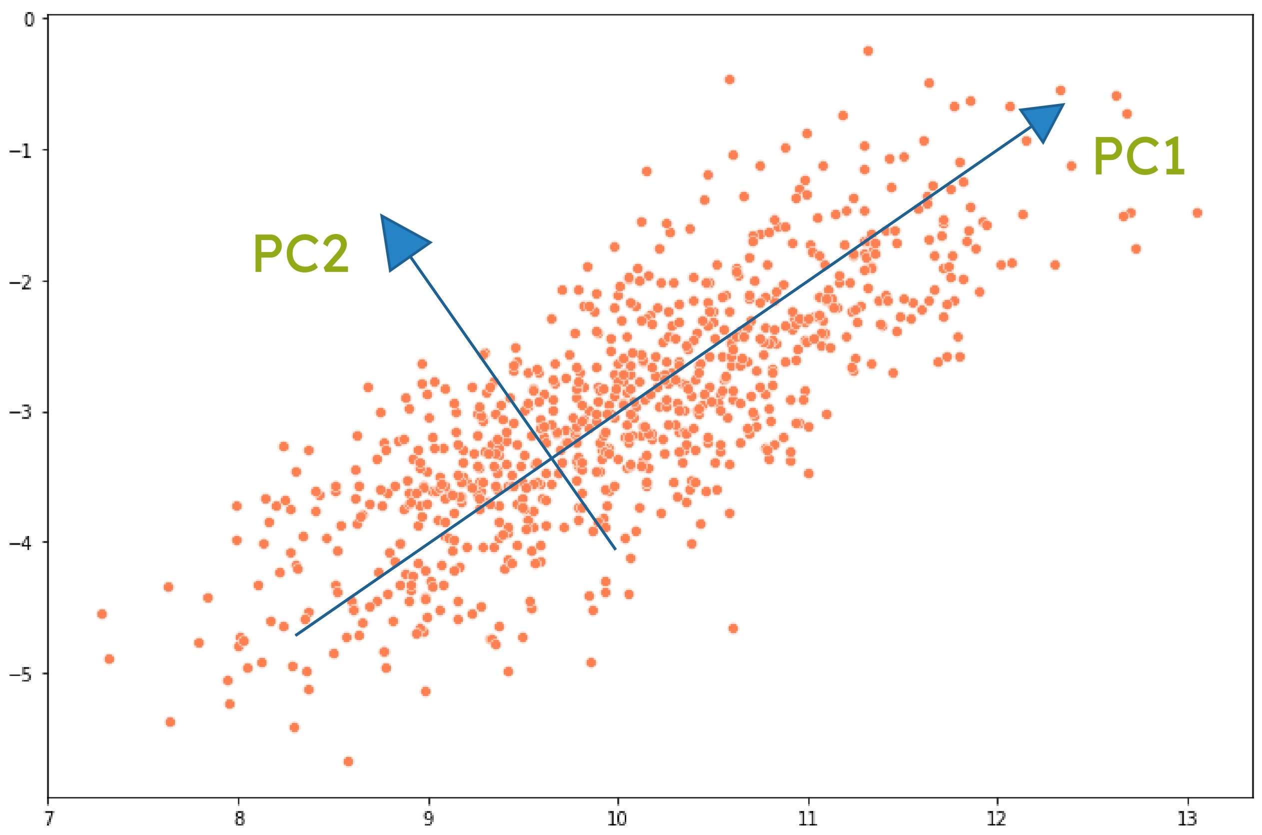
	Feature (1)	Feature (2)	Feature (3)	…	…	Feature (N)
Sample 1	129	55	1.2	…	…	512
Sample 2	22	10	2.2	…	…	543
Sample 3	107	33	4.1	…	…	123
. . .	. . .	. . .	. . .	… … …	… … …	. . .
Sample k	66	33	3.9	…	…	726

A

- This is basically a k x N Matrix **A**, whose Columns are the *Features* and Rows are the *Samples*.
- For large values of N, we need to find p << N so that we can use a k x p Matrix **A’** in place of **A** which can describe the data of almost fully.

PCA Method

Geometrical Interpretation

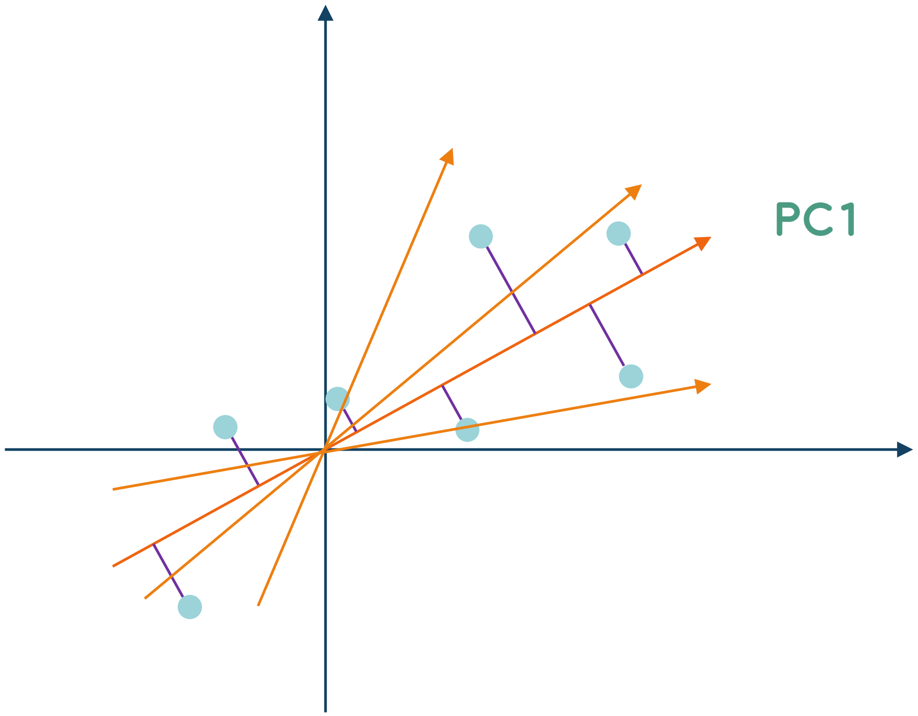


- **PC1** describes the maximum variance of the data.
- After PC1, **PC2** describes the variance of the data as 2nd maximum.

Goal

to find the direction of these PC1, PC2…

Geometrical Interpretation



- **For PC1,** we will find a direction which can describe the variance of the data mostly.
- Basically, each Principal Component direction is ***the Linear Combination of the existing axis.***
- Then, we will project the data on the direction of the PCs.

Goal

to find the PC1, we have to

Maximize [Var (projected data)]

PCA Method

Problem Framing

For, j^{th} column ($j = 1, 2, 3, \ldots, N$)

$$x_{ij} = a_{ij} - \bar{a}_j \qquad \forall\, i = 1, 2, 3, \ldots, k$$

$$\boldsymbol{X} = \begin{pmatrix} x_1^T \\ x_2^T \\ x_3^T \\ \cdot \\ \cdot \\ \cdot \\ x_k^T \end{pmatrix}$$

- ❑ First, centralize the data for each column of data Matrix **A** at O (origin of the space) i.e. Mean of each column = 0
- ❑ Let, the new Matrix be $\mathbf{X_{k*N}}$
- ❑ We have rows of the X as a Data Point in N-dim space.
- ❑ We are projecting this rows along Unit Vector of **PC1.**

Problem Framing

For, j^{th} column (j = 1, 2, 3, …, N)

$$x_{ij} = a_{ij} - \bar{a}_j \qquad \forall\ i = 1, 2, 3, \ldots, k$$

- First, centralize the data for each column of data Matrix **A** at O (origin of the space) i.e. Mean of each column = 0
- Let, the new Matrix be $\mathbf{X_{k*N}}$

$$\boldsymbol{Max} \sum_{i=1}^{k} \left((x_i)^T \hat{u} \right)^2 \qquad s.t.,\ u^T u = 1$$

- We have rows of the X as a Data Point in N-dim space.
- We are projecting this rows along Unit Vector of **PC1.**

PCA Method

Mathematical Formulation

$$\boldsymbol{Max} \quad \hat{u}^T \, \boldsymbol{C} \, \hat{u}$$

$\boldsymbol{C}$ = Covariance Matrix

$$\boldsymbol{var}(\boldsymbol{X}) = \frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})^2$$

$$\boldsymbol{cov}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{k}\sum_{i=1}^{k}(x_i - \bar{x})(y_i - \bar{y})$$

$$\begin{array}{cc} & \begin{array}{ccc} \text{x} & \text{y} & \text{z} \end{array} \\ \begin{array}{c} \text{x} \\ \text{y} \\ \text{z} \end{array} & \begin{bmatrix} var(x) & cov(x, y) & cov(x, z) \\ cov(x, y) & var(y) & cov(y, z) \\ cov(x, z) & cov(y, z) & var(z) \end{bmatrix} \end{array}$$

Solving the Maximization Problem by using Lagrange Multiplier Method

$$\nabla f = \nabla \left[\hat{u}^T C \, \hat{u} + \lambda \, (u^T u = 1) \right]$$

- u *is unknown here*
- $\lambda = Lagrange\ Multiplier$

$$= 2C\hat{u} - 2\lambda\hat{u} = 0$$

$$\Rightarrow Cu = \lambda u$$

$$\Rightarrow u^T C u = \lambda u^T u$$

$$\Rightarrow u^T C u = \lambda$$

- So, we are finding the maximum Eigen Value of the Covariance Matrix.
- Since, Covariance Matrix is symmetric and positive semi-definite, so every eigen value will be a non-negative real number and corresponding Eigen Vectors are orthogonal.
- By arranging the eigen values in descending order, we will find the sequence of Eigen Vectors which will give the Principal Components.

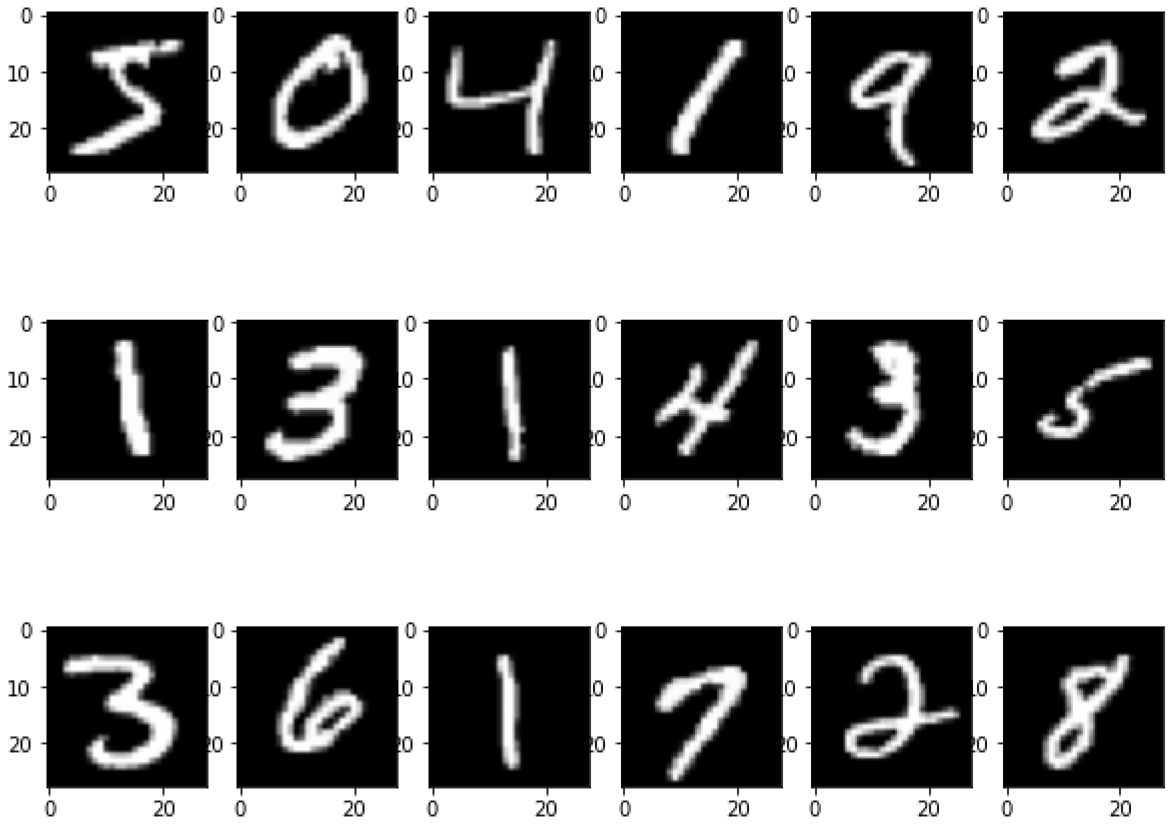
Process to Find Principal Components

- First, Calculate the *Covariance Matrix* ($\boldsymbol{C}$)
- Then, Find the *Eigen Values* of $\boldsymbol{C}$ and arrange them in ***descending order***.
- After getting that descending sequence of Eigen Values, ***write down the Eigen Vectors corresponding to those eigen values.***
- Finally, **PC1** will be the *Eigen Vector* corresponding to the *highest Eigen Value*, **PC2** will be the *Eigen Vectors* corresponding to the *Second Highest Eigen Value* and So on.
- Furthermore, you can calculate the *Contribution Score* of each PC.
- According to the contribution, you choose some of the Principal Components.

Application

Visualization of a 28x28 (=784) dim Image data in 2D

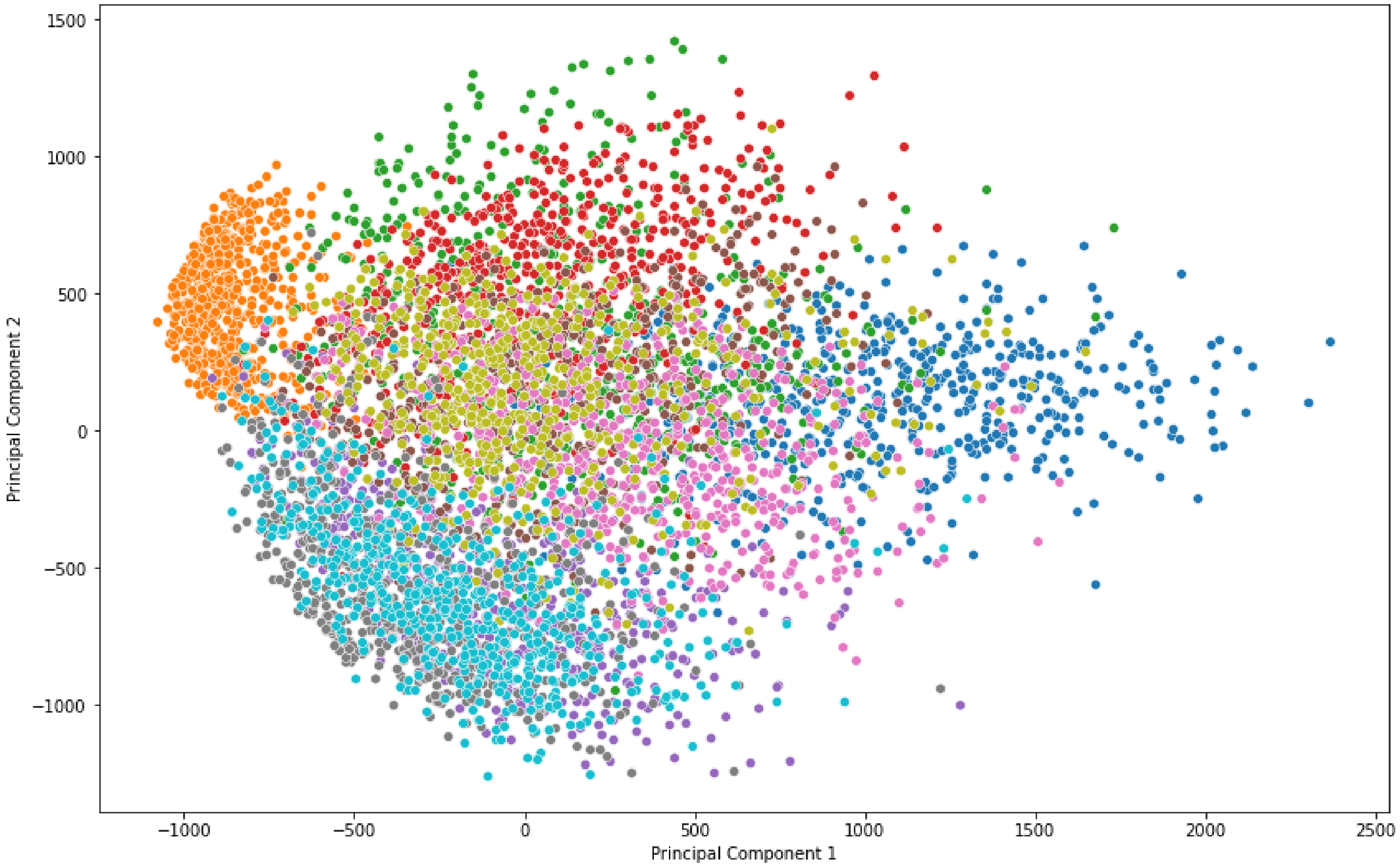
- This is a sample of 28x28 pixel Image Dataset.
- We need 28*28 = 784 values (Grayscale pixel value Range: 0 – 255) to describe a single image.
- Each Datapoint is a point in 784 dim Space which can’t be visualized.
- So, we need PCA to transform this data to a lower version of itself.



Application

Visualization of a 28x28 (=784) dim Image data in 2D

- After applying PCA, we transformed the higher dimensional data to it's best 2 dimensional representation.
- Each color denotes a different numbers between 0 to 9.



Applications

- PCA is predominantly used as a dimensionality reduction technique in domains like

Facial Recognition	***Pattern Recognition***
Computer Vision	***Image Compression***
Visualization	***Time Series Prediction***

- PCA is used to preprocess the raw data before putting it in Machine Learning model.
- It is often used to help in dealing with multi-collinearity before a model is developed.
- PCA is also applied in ***Healthcare industries*** in multiple areas like patient insurance data where there are multiple sources of data and with a huge number of variables that are correlated to each other.

References

- G. T. Reddy et al., "Analysis of Dimensionality Reduction Techniques on Big Data," in IEEE Access, vol. 8, pp. 54776-54788, 2020, doi: 10.1109/ACCESS.2020.2980942. Link: IEEE Xplore Full-Text PDF
- Chapter 2: 12.2. PCA (Page: 498 – 510) of the Book: An Introduction to Statistical Learning (ISLR) Version 2
- Wikipedia: Principal component analysis – Wikipedia
- Some YouTube Channels to understand it Visually:
  - StatQuest with Josh Starmer | PCA Step-by-Step
  - Visually Explained | PCA
  - Serrano.Academy | PCA

THANK YOU!

by: Ranjan Sarkar