Principal Component Analysis (PCA)

Jayin Khanna
MAT399: UG Seminar Presentation 1 Report
Shiv Nadar University

12 February, 2025

Abstract

Principal Component Analysis (PCA) is widely used as a dimensionality reduction technique in **Data analysis and Machine Learning**. The technique is used to transform high-dimensional data to a lower-dimension subspace while preserving the maximum variance of the data. This is done by computing the eigenvectors and eigenvalues of the data covariance matrix. This helps identify the principal components that capture and retain the maximum variance of the data. These principal components are merely the linear combinations of the existing features of the data.
This technique has applications in a wide variety of fields. **It is applied in Finance, Image processing, Bio-informatics etc. In general, any field that has a high number of features** and needs dimensionality reduction for computational efficiency or other reasons. The algorithm is limited to linear datasets and might not be effective in datasets that have non-linear relationships (the technique can be applied to transformed variables). Despite this, PCA is an important tool in many applications.

Introduction

Motivation and Main Idea

Feature Selection

Start with as many features as you can collect and then find a good subset of features.

- Project the given data onto a lower-dimensional subspace such that:
  - Reconstruction error is minimized.
  - Variance of the projected data is maximized.

Idea through an Example

Consider a dataset with five independent variables (features), and we aim to reduce the dimensionality using PCA.

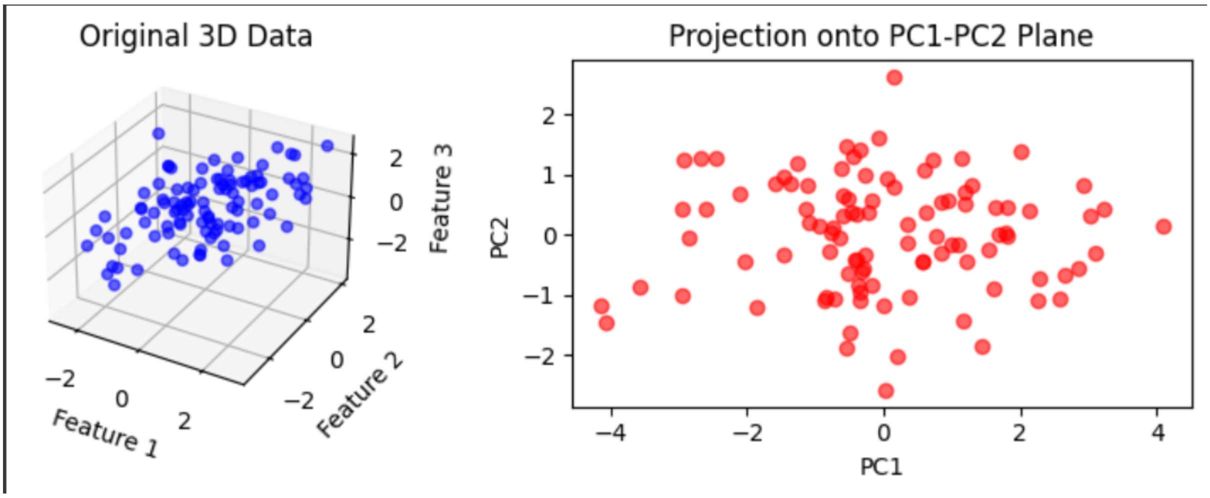

Figure 1: PCA Visualisation

Sample	Feature 1	Feature 2	Feature 3	Feature 4 Feature 5	Feature 6	…
1	8.2	4.5	65	78	34	…
2	7.8	4.8	62	82	36	…
3	9.0	4.2	70	85	38	…
4	8.5	4.3	68	80	35	…
5	7.9	4.7	63	76	32	…
⋮	⋮	⋮	⋮	⋮	⋮	⋱

For example, if we choose $m = 2$, we project the original 5D data onto a 2D subspace while preserving the most important variance.

Applications

1. Applications in Finance
2. To Visualize higher dimensional data in Lower (Comprehendible) dimensions
3. Image compression for Biometric scanners
4. Dimensionality reduction in Machine Learning and Deep Learning
5. ChatGPT uses it, what else do you need?!
6. NLP applications and Word2Vec (Mapping words to higher dimensional vector spaces) and then using PCA to reduce dimensions for easier computation
7. Applications in anything that has annoyingly many characteristics

Intuitive Idea of the algorithm :

How does the algorithm find the 'right' lower dimensional space?
This section will also aim to clarify what 'right' really means in terms of dimensionality reduction.

Given some data points in higher dimensional space, say we choose an arbitrary 2-dimensional subspace to project on. This is what a 'bad' PCA algorithm might do:

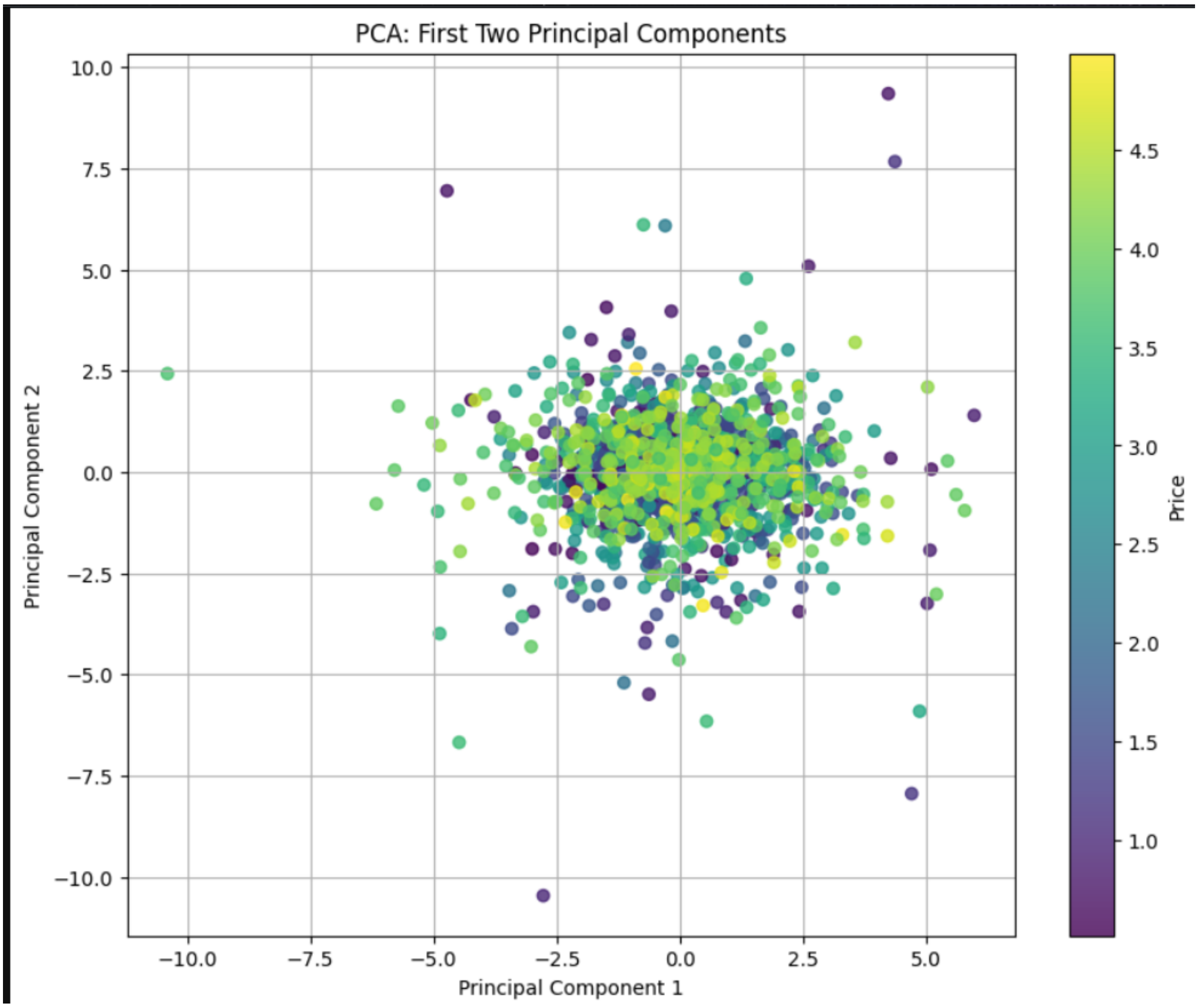

Figure 2: PCA 1

Cluster all the points in one small space, without preserving any of the variance in the data points, making it into an an indistinguishable cluster of points, where different features can not be seperated easily.

Therefore, the subspace plane must be selected so that, maximum varinace within the datapoints in preserved after projecting it onto the lower-dimensional subspace.

The is done by finding the direction where the data points have the highest variance and using subspace parallel, to that direction as your projection subspace. This might look something like the below image:

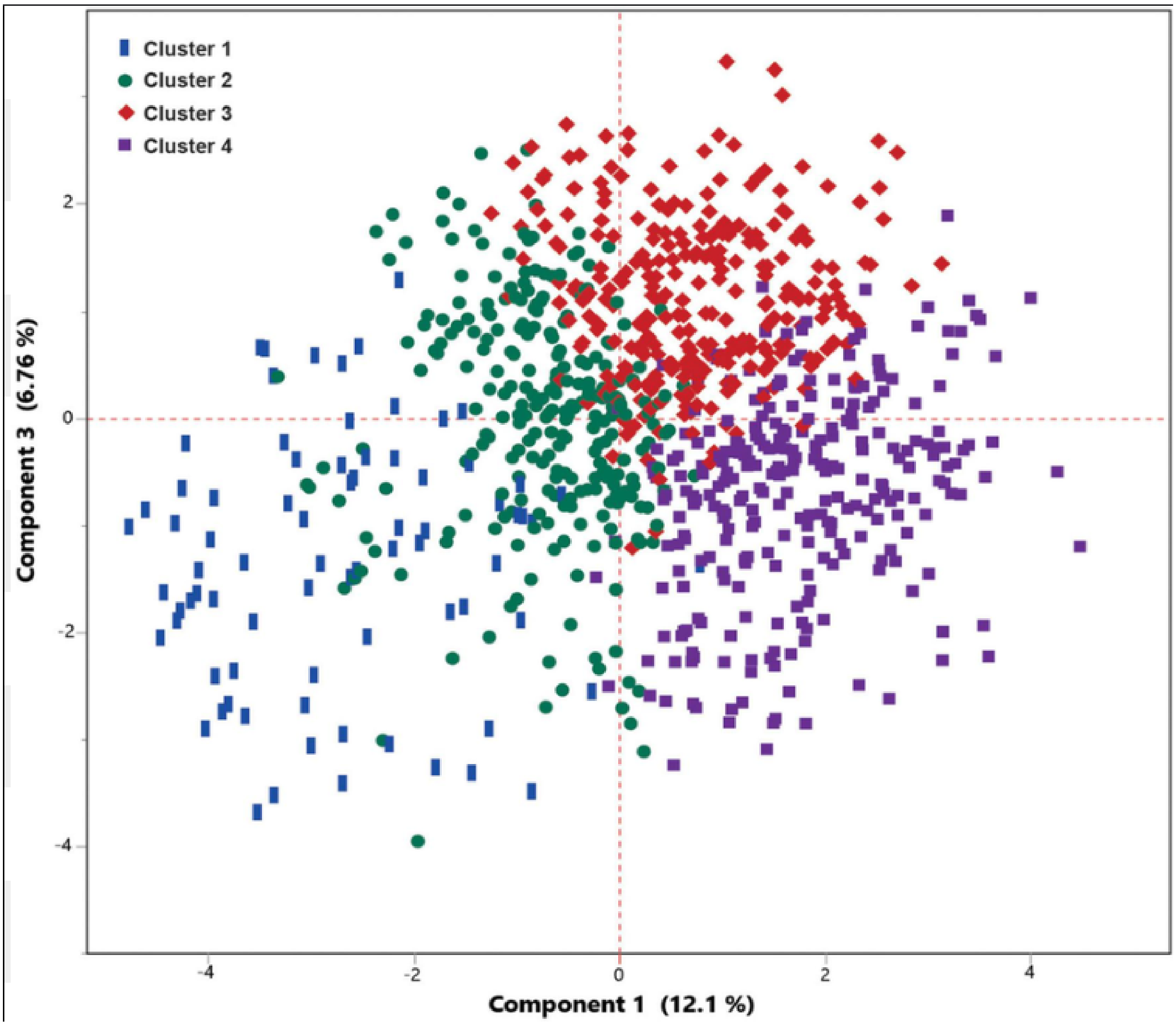

Figure 3: PCA 2

Theory

The theory section is divided into the following sections:

- **Problem Statement**
- **Goal of the PCA Algorithm**
- **Expressing Each Data Point**
- **The Minimization Problem**
- **Finding the Optimal Projection**

- **Error Formulation**
- **Optimization Problem**
- **PCA Algorithm Up-shot**

Problem Formulation

Given: Dataset $D = \{x_1, x_2, \dots, x_n\}$, where $x_i \in \mathbb{R}^d$.
Goal: Project D onto an m-dimensional subspace where m is an ”input parameter”.

Arriving at PCA Algorithm

Let $\mathcal{B} = \{u_1, u_2, \dots, u_m\}$ be an orthonormal basis for an m-dimensional subspace.

The idea is to use $\mathcal{B}$ and find the best projection.

Now extend $\mathcal{B}$ to an orthonormal basis for $\mathbb{R}^d$:

$$\mathcal{B}' = \{u_1, u_2, \dots, u_m, u_{m+1}, \dots, u_d\}$$

Any vector $x \in \mathbb{R}^d$ can be written using $\mathcal{B}'$ as follows:

$$x_i = \sum_{j=1}^{d} \beta_j u_j$$

where $\beta_j = x_i^T u_j$ for all j.

Expressing Each Data Point

$$x_i = \underbrace{\sum_{j=1}^{m} z_{ij} u_j}_{\text{Projection onto } m\text{-dimensional subspace}} + \underbrace{\sum_{j=m+1}^{d} \beta_j u_j}_{\text{Reconstruction error}}$$

Now, we find the optimal z_{ij} and β_j to minimize the squared error.

$$J_s = \frac{1}{n} \sum_{i=1}^{n} \|x_i - \hat{x}_i\|^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{j=1}^{m} (x_i^T u_j - z_{ij}) + \sum_{j=m+1}^{d} (x_i^T u_j - \beta_j) \right]^2$$

$$= \frac{1}{n} \sum_{i=1}^{n} \left[\sum_{j=1}^{m} (x_i^T u_j - z_{ij})^2 + \sum_{j=m+1}^{d} (x_i^T u_j - \beta_j)^2 \right]$$

Minimization

$$\frac{1}{n}\sum_{i=1}^{n}\left[\sum_{j=1}^{m}(x_i^T u_j - z_{ij})^2 + \sum_{j=m+1}^{d}(x_i^T u_j - \beta_j)^2\right]$$

To find the minimum, we take the partial derivatives and set it to zero:

$$\frac{\partial J}{\partial z_{ij}} = 0 \Rightarrow 2(x_i^T u_j - z_{ij}) = 0$$

$$\boxed{z_{ij} = x_i^T u_j}$$

Thus,

$$\boxed{\beta_j = \frac{1}{n}\sum_{i=1}^{n} x_i^T u_j = \bar{x}^T u_j}$$

where

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Optimal Projection

For a given m-dimensional subspace spanned by $\mathcal{B} = \{u_1, u_2, \ldots, u_m\}$, the best projected data is:

$$\hat{x_i} = \sum_{j=1}^{m}(x_i^T u_j)u_j + \sum_{j=m+1}^{d}(\bar{x}^T u_j)u_j$$

$$x_i - \bar{x} = \sum_{j=m+1}^{d}(x_i^T u_j - \bar{x}^T u_j)u_j$$

Since u_j are orthonormal:

$$\|x_i - \bar{x}\|^2 = \sum_{j=m+1}^{d}(x_i^T u_j - \bar{x}^T u_j)^2$$

$$\|x_i - \bar{x}\|^2 = \sum_{j=m+1}^{d}[(x_i^T - \bar{x})u_j]^2$$

$$= \sum_{j=m+1}^{d}\big((x_i - \bar{x})^T u_j\big)^2$$

Error Formulation

With the optimal z_{ij}, the squared error becomes:

$$J^* = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=m+1}^{d} \left((x_i - \bar{x})^T u_j \right)^2$$

$$= \frac{1}{n} \sum_{j=m+1}^{d} \sum_{i=1}^{n} \left((x_i - \bar{x})^T u_j \right)^T \left((x_i - \bar{x})^T u_j \right)$$

$$= \sum_{j=m+1}^{d} u_j^T \left[\frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T \right] u_j$$

Defining the covariance matrix:

$$\boxed{C = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T}$$

we get:

$$J^* = \sum_{j} u_j^T C u_j$$

Our goal is to choose $u_1, \ldots, u_m$ such that J is minimized.

Optimization Problem

Consider the constrained optimization problem:

$$\min_u u^T C u, \quad \text{subject to } u^T u = 1$$

Using Lagrange multipliers:

$$\mathcal{L}(u, \lambda) = u^T C u + \lambda(1 - u^T u)$$

Setting the gradient to zero:

$$\nabla_u \mathcal{L}(u, \lambda) = 0 \Rightarrow Cu = \lambda u$$

This means that the optimal u_i s are eigenvectors of C, corresponding to the smallest eigenvalues.

$$\boxed{C = \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T}$$

Since C is a real symmetric matrix, all its eigenvalues are real, and an orthonormal basis of eigenvectors exists.

Let $u_1, u_2, \ldots, u_m$ correspond to eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ such that:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_d$$

PCA Algorithm

1. **Input:** Data points $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$.
2. Compute mean:
$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$$
3. Compute covariance matrix:
$$C = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(x_i - \bar{x})^T$$
4. Compute eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_d$ and corresponding eigenvectors $u_1, u_2, \ldots, u_d$.
5. Choose the top m eigenvectors $u_1, \ldots, u_m$.
6. Project data onto the new subspace:
$$z_i = \sum_{j=1}^{m}(x_i^T u_j)u_j$$

Conclusion

PCA reduces the dimensionality of data by selecting principal components that maximize variance while minimizing reconstruction error. This technique is widely used in data preprocessing, compression, and easier visualization oh higher dimensional data.

References

[1] IIT Madras BS in Data Science and Applications video lectures on Machine Learning Practices