# Principal Component Analysis(PCA)

Jiman Hwang

Disclaimer: This document is for self-study only and may contain false information.

## 1 Abstract

This is a technique that can diminish the dimension of given data by adjusting axes in a way of maximizing variance of newly defined variance. Also, the new variables become uncorrelated.

## 2 Explanation

### 2.1 Prerequisites

Suppose we have data from observation(all real values). Having $n$ sources(e.g. sensors), we experimented $N$ times, and obtained Table 1.

Table 1: Given data

| Source \ Serial # | 1 | 2 | $\cdots$ | N |
|---|---|---|---|---|
| 1 | $x_{11}$ | $x_{12}$ | $\cdots$ | $x_{1N}$ |
| 2 | $x_{21}$ | $x_{22}$ | $\cdots$ | $x_{2N}$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\ddots$ | $\vdots$ |
| n | $x_{n1}$ | $x_{n2}$ | $\cdots$ | $x_{nN}$ |

We handle it by defining a random variable for each source assuming each value in a source is equally probable. Let $X_i$ denote the random variable for source $i$ where

$$\forall j \ \mathrm{P}_{X_i}\left(x_{ij}\right) = \frac{1}{N}$$

So far, each source has one random variable, and we build a random vector

$$\mathbf{X} \overset{\text{def}}{=} \begin{bmatrix} X_1 & \cdots & X_n \end{bmatrix}^T$$

## 2.2 Overview

Because we'll focus on the variation of data, the average is useless. Thus, our first task is to make its average 0 by translation. Then, align the data by rotation so that

$$\boxed{\begin{array}{l}\text{it reveals the largest variation along } 1^{\text{st}} \text{ axis,}\\ \text{the largest variation along } 2^{\text{nd}} \text{ axis ignoring } 1^{\text{st}} \text{ dimensional values,}\\ \text{the largest variation along } 3^{\text{rd}} \text{ axis ignoring } 1^{\text{st}}, 2^{\text{nd}} \text{ dimensional values,}\\ \text{the largest variation along } 4^{\text{th}} \text{ axis ignoring } 1^{\text{st}}, 2^{\text{nd}}, 3^{\text{rd}} \text{ dimensional values,}\\ \cdots \text{ and so on}\end{array}} \tag{1}$$

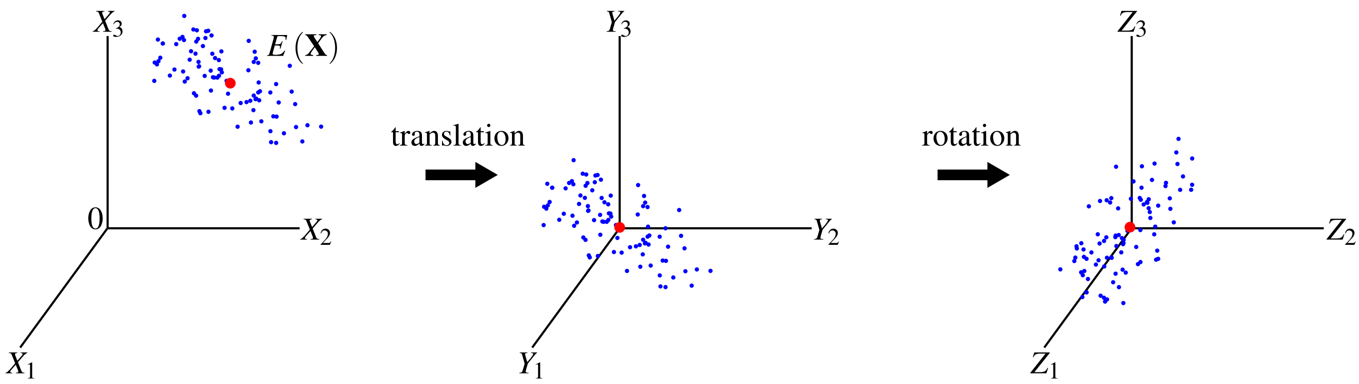


Figure 1: PCA overview

## 2.3 Getting transformation

To begin with, we have to move the data making its average 0. Define a random vector $\mathbf{Y}$

$$\mathbf{Y} \stackrel{\text{def}}{=} \begin{bmatrix} Y_1 & \cdots & Y_n \end{bmatrix}^T \stackrel{\text{def}}{=} \mathbf{X} - E(\mathbf{X})$$

Next, rotation is a linear transformation represented by an orthogonal matrix. Define a random vector $\mathbf{Z}$ and $n \times n$ matrix $U$

$$\mathbf{Z} \stackrel{\text{def}}{=} \begin{bmatrix} Z_1 & \cdots & Z_n \end{bmatrix}^T \stackrel{\text{def}}{=} U\mathbf{Y} \quad \left(U^T U = I\right)$$

where $Z_1, Z_2, \cdots, Z_n$ satisfy (1). To figure out $\operatorname{Var}(Z_i)$, get the covariance matrix of $\mathbf{Z}$, $\mathbf{C_Z}$, which contains the variances on its diagonal.

Note that for any random vector $\mathbf{A} = \begin{bmatrix} A_1 & \cdots & A_n \end{bmatrix}^T \in \mathbb{R}^n$ such that $\mathrm{E}(\mathbf{A}) = \mathbf{0}$,

$$\mathbf{C_A} = \left[\operatorname{Cov}(A_i, A_j)\right] = \left[\mathrm{E}(A_i A_j) - \mathrm{E}(A_i)\,\mathrm{E}(A_j)\right] = \left[\mathrm{E}(A_i A_j)\right] = \mathrm{E}\left(\mathbf{A}\mathbf{A}^T\right)$$

This follows

$$\mathbf{C_Z} = \mathrm{E}\left(\mathbf{Z}\mathbf{Z}^T\right), \; \mathbf{C_Y} = \mathrm{E}\left(\mathbf{Y}\mathbf{Y}^T\right)$$

Moreover,

$$\mathbf{C_Z} = \mathrm{E}\left(\mathbf{Z}\mathbf{Z}^T\right) = \mathrm{E}\left(U\mathbf{Y}\mathbf{Y}^T U^T\right) = U\mathrm{E}\left(\mathbf{Y}\mathbf{Y}^T\right)U^T = U\mathbf{C_Y}U^T \tag{2}$$

Let $\mathbf{e}_1, \cdots, \mathbf{e}_n$ denote standard basis. Then,

$$\mathrm{Var}\left(Z_i\right) = \mathbf{e}_i^T \mathbf{C_Z} \mathbf{e}_i \tag{3}$$

Note that $\mathbf{C_Y}$ is symmetry, hence orthogonally diagonalizable(Spectral theorem at p.397 in [Lay11]).

$$\mathbf{C_Y} = PDP^T \quad \left(P^T P = I,\ D = \mathbf{diag}\left\{\lambda_1, \lambda_2, \cdots, \lambda_n\right\}\right) \tag{4}$$

where $\lambda_i \in \mathbb{R}$. We assume $\lambda_1 \geq \cdots \geq \lambda_n$ since this is always possible by exchanging rows and columns of $P$ and $D$. In addition, $\mathbf{C_Y}$ is positive semi-definite(above Thm 5 at p.405 in [Lay11]) because

$$\forall \mathbf{u} \in \mathbb{C}^n \quad \mathbf{u}^T \mathbf{C_Y} \mathbf{u} = \mathbf{u}^T E\left(\mathbf{Y}\mathbf{Y}^T\right)\mathbf{u} = E\left(\mathbf{u}^T \mathbf{Y}\mathbf{Y}^T \mathbf{u}\right) = E\left(\left\|\mathbf{Y}^T\mathbf{u}\right\|^2\right) \geq 0$$

It follows

$$\lambda_1 \geq \cdots \geq \lambda_n \geq 0$$

Using (2), (3), (4)

$$\mathrm{Var}\left(Z_i\right) = \mathbf{e}_i^T U P D P^T U^T \mathbf{e}_i$$

Let $\mathbf{z}_i = \begin{bmatrix} z_{1i} & \cdots & z_{ni} \end{bmatrix}^T = P^T U^T \mathbf{e}_i$ then,

$$\mathrm{Var}\left(Z_i\right) = \mathbf{z}_i^T D \mathbf{z}_i = \lambda_1 z_{1i}^2 + \cdots + \lambda_n z_{ni}^2$$

Since $\|\mathbf{z}_i\| = 1$, $\mathbf{z}_1 = \mathbf{e}_1$ or $UP\mathbf{e}_1 = \mathbf{e}_1$. Note that $z_{1i} = 0$ for $i \geq 2$ because

$$z_{1i} = \mathbf{e}_1^T \mathbf{z}_i = \mathbf{e}_1^T P^T U^T \mathbf{e}_i = \left(UP\mathbf{e}_1\right)^T \mathbf{e}_i = \mathbf{e}_1^T \mathbf{e}_i = 0 \quad \forall i \geq 2$$

This implies

$$\mathrm{Var}\left(Z_2\right) = \mathbf{z}_2^T D \mathbf{z}_2 = \lambda_2 z_{22}^2 + \lambda_3 z_{32}^2 + \cdots + \lambda_n z_{n2}^2$$

To maximize it on the condition $\|\mathbf{z}_i\| = 1$, it should be $\mathbf{z}_2 = \mathbf{e}_2$. Doing this method repeatedly, we have

$$\forall i \quad UP\mathbf{e}_i = \mathbf{e}_i$$

Assembling them,

$$\begin{aligned} &\begin{bmatrix} UP\mathbf{e}_1 & \cdots & UP\mathbf{e}_n \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_n \end{bmatrix} \\ \Rightarrow\ & UP\begin{bmatrix} \mathbf{e}_1 & \cdots & \mathbf{e}_n \end{bmatrix} = I \\ \Rightarrow\ & UP = I \\ \Rightarrow\ & U = P^T \end{aligned}$$

That is, rotating the translated data in accordance with $P^T$ gives the new data that satisfies (1).

## 2.4 Additional features of transformed data

Aside from transformed data features (1), there are more to know.

First, the distribution along each axis is uncorrelated to one another because $\mathbf{C_Z}$ is diagonal.

Also, each column vector of $P$ indicates the direction through which the translated data satisfies (1).

Furthermore, we can reduce the dimension, or project onto subspace, during the transformation minimizing the loss. Suppose $P = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_n\end{bmatrix}$. If we want to put the data into $k < n$ dimensional space, we may apply

$$P' = \begin{bmatrix}\mathbf{v}_1 & \cdots & \mathbf{v}_k\end{bmatrix}$$

to the translated data. Because $\mathbf{v}_1, \cdots, \mathbf{v}_k$ are the most $k$ significant basis in variation, the transformed data shows the minimum loss having $k$ dimensions.

## 2.5 Applying to data

Given data Table 1, we handle it as a matrix.

$$R \stackrel{\text{def}}{=} \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nN} \end{bmatrix}$$

To begin with, translate it.

$$\widetilde{R} \stackrel{\text{def}}{=} \begin{bmatrix} \widetilde{x}_{11} & \cdots & \widetilde{x}_{1N} \\ \vdots & \ddots & \vdots \\ \widetilde{x}_{n1} & \cdots & \widetilde{x}_{nN} \end{bmatrix} \stackrel{\text{def}}{=} R - \frac{1}{N}\sum_{j=1}^{N} \begin{bmatrix} x_{1j} \\ \vdots \\ x_{nj} \end{bmatrix}$$

Next, build the covariance matrix.

$$\mathbf{C}_{\widetilde{\mathbf{R}}} = \frac{1}{N}\widetilde{R}\widetilde{R}^T$$

Then orthogonally diagonalize $\mathbf{C}_{\widetilde{\mathbf{R}}}$, that is, get $P$ and $D$ such that

$$\mathbf{C}_{\widetilde{\mathbf{R}}} = PDP^T$$

where $P$ is orthogonal, $D = \mathbf{diag}\{\lambda_1, \cdots, \lambda_n\}$ and $\lambda_1 \geq \cdots \geq \lambda_n$.

Assuming we want to leave $k$ dimensions, extract the first $k$ columns from $P$. Let $Q$ denote this, then $Q^T\widetilde{R}$ is the projected result.

## 2.6 Example

There are 3 sources whose distributions are

$$X_1 \sim \text{Uniform}(1/2, 3/2)$$
$$X_2 \sim \text{Uniform}(1/2, 5/2)$$
$$X_3 = X_1 + X_2 + 1$$

and we obtained $N = 500$ samples from them. Proceeding the transformation,

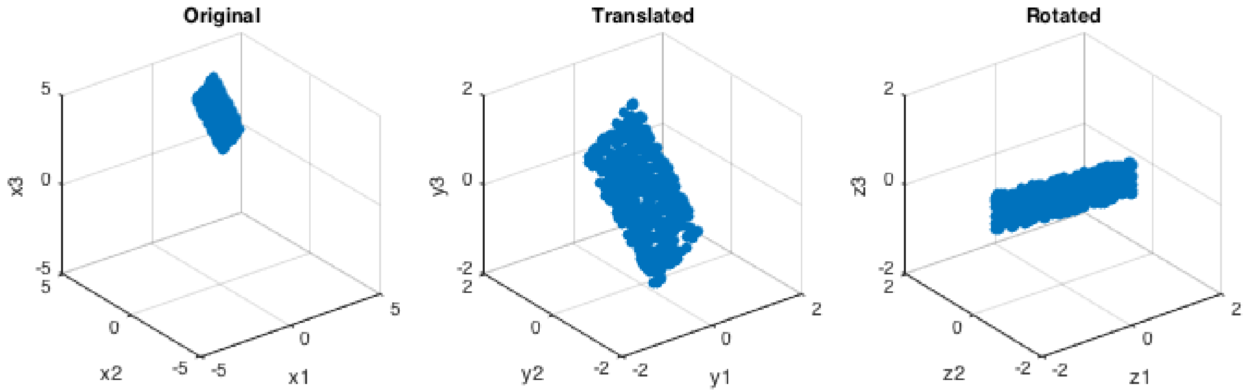


Figure 2: PCA Example(3D)

and the projection onto $z_1 - z_2$ plane.

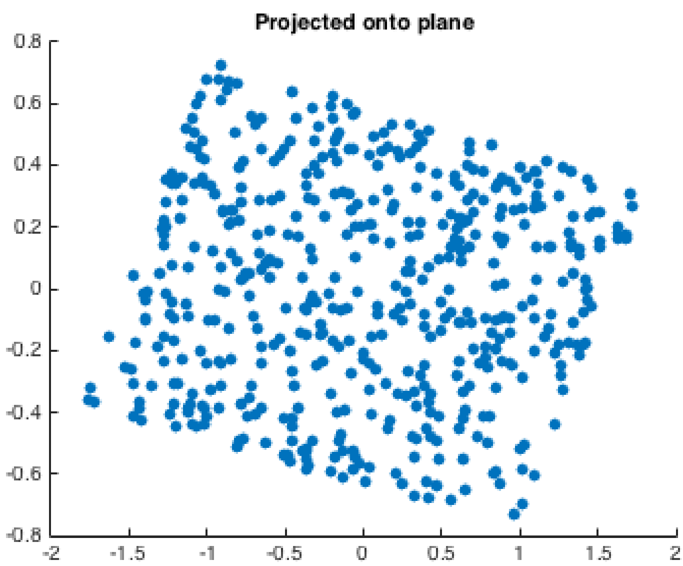


Figure 3: Projection onto plane

Used Matlab code is follow.

```
% setting
n = 3;     N = 500;
x1 = 0.5+rand(1,N);
x2 = 0.5+2*rand(1,N);
x3 = x1+x2+1;

% translation
y1 = x1-mean(x1);     y2 = x2-mean(x2);     y3 = x3-mean(x3)
   ;

% rotation
Y = [y1; y2; y3];
[P, D] = eig((1/N)*Y*Y');
if issorted(diag(D))
    error('D is not in descending order. Rerun the script.
       ');
end
Z = P'*Y;
z1 = Z(1,:);     z2 = Z(2,:);     z3 = Z(3,:);

% plot raw data
subplot(1,3,1);
scatter3(x1,x2,x3,'filled');
axis([-5 5 -5 5 -5 5]);
xlabel('x1');     ylabel('x2');     zlabel('x3');
title('Original');

% plot translated data
subplot(1,3,2);
scatter3(y1,y2,y3,'filled');
axis([-2 2 -2 2 -2 2]);
xlabel('y1');     ylabel('y2');     zlabel('y3');
set(gca,'xtick',-2:2:2);
set(gca,'ytick',-2:2:2);
set(gca,'ztick',-2:2:2);
title('Translated');

% plot rotated data
subplot(1,3,3);
scatter3(z1,z2,z3,'filled');
axis([-2 2 -2 2 -2 2]);
xlabel('z1');     ylabel('z2');     zlabel('z3');
set(gca,'xtick',-2:2:2);
set(gca,'ytick',-2:2:2);
```

```
set(gca,'ztick',-2:2:2);
title('Rotated');

% reducing dimension
figure
k = 2;  % reduced dimension
Q = P(:,1:k);
size(Q)
size(Y)
R = Q'*Y;
r1 = R(1,:);
r2 = R(2,:);
scatter(r1,r2,'filled')
title('Projected onto plane');
```

Code 1: Example code

# References


[Lay11] David C. Lay. *Linear Algebra and Its Applications*. Pearson, 4 edition, 2011.