CH510L Machine Learning in Process Engineering

Course Project

Probabilistic Principal Component Analysis

Group -8

Table of Contents

Problem Statement

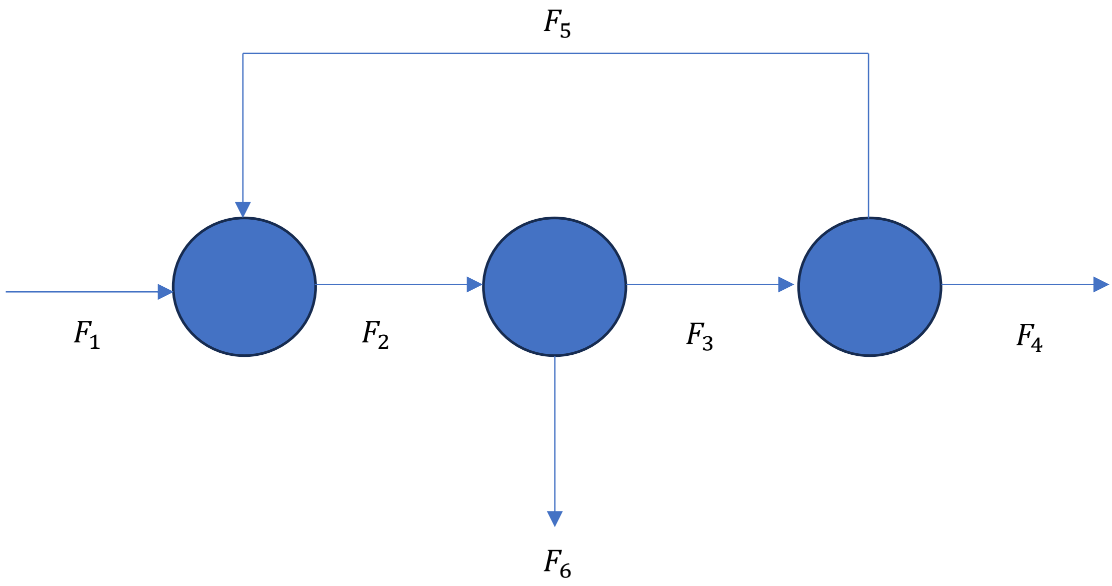
To efficiently extract relevant features and capture the underlying structure of the Flow network system and to identify the stead state flow network model by implementing Principal Component Analysis (PCA), Probabilistic Principal Component Analysis (PPCA) with Maximum Likelihood Estimation (MLE), and PPCA with Expectation-Maximization (EM) algorithms.

Objectives

- Simulate flow network dataset.
- Implement the PCA model to identify the stead state flow network model.
- Implement the PPCA model to identify the stead state flow network model.

Theory of Solution Methods

Flow network



The flow process consists of 6 various flowrates (F_1, F_2, F_3, F_4, F_4 *and* F_6).

Balancing of the flowrates:

$F_5 = F_2 - F_1$……………………….. (1)

$F_6 = F_2 - F_3$………………………...(2)

$F_4 = F_3 - F_2 + F_1$ …………………...(3)

So, there are 3 variables and 3 equations. F_1, F_2 and F_3 are independent variables and F_4, F_5 and F_6 are dependent variables.

Principal component analysis

Principal Component Analysis, is a technique for dimensionality of datasets, by increasing interpretability but at the same time minimizing the information loss. It involves finding the principal axes of a set of observed data vectors through maximum likelihood estimation of parameters in a latent variable model that is closely related to factor analysis . The most common derivation of PCA is in terms of a standardized linear projection which maximizes the variance in the projected space . In other words, PCA identifies the directions in which the data varies the most and projects the data onto those directions, reducing the dimensionality of the data while retaining as much of the original information as possible. The principal axes are determined by computing the eigenvectors and eigenvalues of the sample covariance matrix . PCA has many applications, including data compression, image processing, visualization, exploratory data analysis, pattern recognition, and time series prediction.

Principal component: They are the new variables that are constructed as linear combinations or mixtures of initial variables

Properties of Dimensionality reduction

- No of principal components is always less than or equal to no of attributes
- Principal components are orthogonal

- Priority of principal components decreases as number of principal components increases.

For given set of observed of dimensional data vectors t_n

$t_n \in d \times 1 \qquad$ n $\in \{1, 2, 3 \dots N\}$

Let unit vectors along q-principal axes be w_j

$w_j \in d \times 1 \qquad$ j $\in \{1, 2, 3 \dots q\}$

Let q-principal components of observed vector t_n on q-principal axes after shifting origin to $\bar{t}$ be x_n

$x_n \ \in d \times 1 \qquad$ n $\in \{1, 2, 3 \dots N\}$

Then

projection of $(t_n - \bar{t})$ on w_1 is $w_1^T(t_n - \bar{t}) \qquad [\because w \ is \ a \ unit \ vector]$

projection of $(t_n - \bar{t})$ on w_2 is $w_2^T(t_n - \bar{t})$

$\vdots$

projection of $(t_n - \bar{t})$ on w_q is $w_q^T(t_n - \bar{t})$

$$x_n = \begin{bmatrix} w_1^T(t_n - \bar{t}) \\ w_2^T(t_n - \bar{t}) \\ \vdots \\ w_q^T(t_n - \bar{t}) \end{bmatrix} = \begin{bmatrix} w_1^T \\ w_2^T \\ \vdots \\ w_q^T \end{bmatrix} [t_n - \bar{t}]$$

$$x_n = W^T(t_n - \bar{t})$$

where $W = (w_1, w_2, w_3 \dots w_q) \qquad W \in d \times q$

$$\bar{x} = W^T \frac{\sum t_n}{N} - W^T \frac{\sum \bar{t}}{N}$$

$$= W^T\bar{t} - W^T\bar{t} = 0$$

Variance of $x_n = var(x_n) = E[(x_n - \bar{x})(x_n - \bar{x})^T]$

$$var(x_n) = E[x_n {x_n}^T]$$

$$= E[W^T(t_n - \bar{t})(t_n - \bar{t})^T W]$$

$$= W^T E[(t_n - \bar{t})(t_n - \bar{t})^T]W$$

$$= W^T S W$$

Here S is sample covariance matrix $S = \frac{\sum_n (t_n - \bar{t})(t_n - \bar{t})^T}{N}, \quad S \in d \times d$

As mean of these projection x_n is zero. We can maximize magnitude of projection by maximizing variance of x_n

$$\max \quad W^T S W$$

$$s.t \quad W^T W = I$$

Using Lagrange Multipliers $L = W^T S W + \lambda(I - W^T W) - - - (4)$

$$\frac{\partial L}{\partial W} = 0$$

$$\frac{\partial L}{\partial W} = 2SW - 2\lambda W = 0$$

$$SW = \lambda W - - - (5)$$

Substituting above equation (2) in equation (1)

$$L = W^T \lambda W + \lambda(I - W^T W)$$

$$= W^T \lambda W + \lambda - \lambda W^T W$$

$$= \lambda W^T W + \lambda - \lambda W^T W = \lambda$$

Here L should be maximum, So, λ should be maximum.

And from $-(2)$

$$S[w_1\ w_2\ \ldots\ w_q] = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_q \end{bmatrix} [w_1\ w_2\ \ldots\ w_q]$$

$$Sw_j = \lambda_j w_j$$

Where w_j are eigen vectors corresponding to principal eigen values (Highest eigen values) of sample covariance matrix of S.

So, W_{Pca} is Obtained, the data vector t_n can be represented as,

$$t_n = Wx_n + \bar{t} \quad --- (6)$$

Using SVD:

If our data vector t_n is mean centred and is represented horizontally then we can do PCA using SVD also

$$T = [t_1 \quad t_2 \quad \ldots \quad t_n] \quad \rightarrow \quad T^T = \begin{bmatrix} t_1^T \\ t_2^T \\ \vdots \\ t_n^T \end{bmatrix}$$

Here T^T is data set where our data is represented horizontally (dimension × No.of samples)

Taking SVD of $T^T = U\Lambda V^T$

Where V eigen vector matrix of $T.T^T$

In this mean centred case, our sample covariance matrix is $S = \frac{\sum_n t_n.t_n^T}{N} = \frac{T.T^T}{N}$

S and $T.T^T$ will be having same eigen vectors so here we can directly do SVD and obtain our eigen vectors

Let Principal eigen vectors in V be V_1 $(Here\ V_1 = W)$

And Remaining eigen vectors in V be V_2

$$T^T = U_1\Lambda_1 V_1^T + U_2\Lambda_2 V_2^T$$

$$T^T = U_1\Lambda_1 V_1^T$$

Multiplying V_2 on both sides

$$T^T.V_2 = U_1\Lambda_1 V_1^T.V_2 = 0$$

Taking transpose on both sides

$$V_2^T.T = 0$$

Here $V_2^T\left(Also\ written\ as\ \hat{A}\right)$ represents about our model

Probabilistic Principal component analysis

Probabilistic Principal Component Analysis, is a probabilistic version of PCA that models the observed data as a linear transformation of a lower-dimensional latent variable . It is a generative model that assumes the observed data is generated by a Gaussian distribution with a mean that is a linear function of the latent variable and a covariance matrix that is proportional to the identity matrix . The latent variable is also assumed to be Gaussian with a zero mean and a diagonal covariance matrix . PPCA can be used for dimensionality reduction, missing data imputation, and density estimation . It is closely related to factor analysis, but is more flexible and can handle non-Gaussian data . The number of principal components in PPCA can be chosen to control the model complexity and the predictive power of the model . PPCA can be estimated using an EM algorithm that iteratively maximizes the likelihood function .

Using Maximum Likelihood estimation:

From equation –(6) after ε noise is added to our data vector t_n

$$t_n = Wx + \mu + \varepsilon - - - (7)$$

$t \in d \times 1, \quad W \in d \times q \ \ , \ x_n \in q \times 1$

Consider noise ε of mean 0 and variance Ψ

$\varepsilon \sim \mathcal{N}(0, \Psi)$

$E(t_n) = E(Wx + \mu + \varepsilon)$

$\qquad = WE(x) + \mu + E(\varepsilon) \rightarrow W(0) + \mu + 0$

$E(t_n) = \mu$

$var(t_n) = E[(t_n - \mu)(t_n - \mu)^T]$

$\qquad = E[(Wx + \varepsilon)(Wx + \varepsilon)^T] = E[(Wx + \varepsilon)(x^T W^T + \varepsilon^T)]$

$\qquad = E[Wxx^T W^T + \ \varepsilon x^T W^T + Wx\varepsilon^T + \varepsilon\varepsilon^T]$

$\qquad = E[Wxx^T W^T] + E[2Wx\varepsilon^T] + E[\varepsilon\varepsilon^T]$

$\qquad = WE[xx^T]W^T + E[2Wx\varepsilon^T] + E[\varepsilon\varepsilon^T]$

$\qquad = WE[(x-0)(x-0)^T]W^T + 2WE[x\varepsilon^T] + E[(\varepsilon - 0)(\varepsilon - 0)^T]$

$\qquad = WW^T + 2W[var(x\varepsilon^T) + E(x).E(\varepsilon^T)] + \Psi$

$\qquad = WW^T + \Psi$

$t \sim \mathcal{N}(\mu, WW^T + \Psi)$

Taking $\Psi = \sigma^2 I$

Assuming single Gaussian mixture model.

$P(t) = L = \prod_{i=1}^{N} \mathcal{N}(t_i | x, \mu, WW^T + \sigma^2 I)$

$L = \log(L) = \sum_{i=1}^{N} \log \mathcal{N}(t_i|x, \mu, WW^T + \sigma^2 I)$

Let us consider $C = WW^T + \sigma^2 I$

$L = \frac{N}{2}\log\,(\frac{1}{(2\pi)^{\frac{d}{2}}} \times \frac{1}{|C|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(t_n - \mu)^T C^{-1}(t_n - \mu)\right)$

$L = \frac{N}{2}\log\left(\frac{1}{2\pi}\right)^d + \frac{N}{2}\log(|C|^{-1}) - \sum_{n=1}^{N}\frac{1}{2}(t_n - \mu)^T C^{-1}(t_n - \mu) - - - (8)$

$L = -\frac{N}{2}[d log(2\pi) + \log(|C|) + tr(\frac{1}{N}\sum_{i=1}^{N} C^{-1}(t_n - \mu)^T(t_n - \mu))]$

$[\because x^T A x = tr[x^T A x) = tr(xx^T A) = tr(Axx^T)]$

$[\because S = \frac{1}{N}\sum_{n=1}^{N}(t_n - \mu)(t_n - \mu)^T]$

This is a sample covariance matrix

$L = -\frac{N}{2}[d\log(2\pi) + \log|C| + tr(C^{-1}S)] - \log likelihood - - - (9)$

$\frac{\partial L}{\partial \mu} = 0 \;\rightarrow\; -\frac{\partial}{\partial \mu}[\sum_{n=1}^{N}(t_n - \mu)^T C^{-1}(t_n - \mu)] \;= 0 - \;from\;equation\;(8)$

$\sum_{n=1}^{N}[\frac{\partial (t_n-\mu)^T C^{-1}(t_n-\mu)}{\partial (t_n-\mu)} . \frac{\partial (t_n-\mu)}{\partial \mu}] = 0$

$\sum_{n=1}^{N}\left[-1(C^{-1} + C^{-T})[t_n - \mu]\right] = 0$

$\sum_{n=1}^{N}(t_n - \mu) = 0 \;\leftrightarrow \sum_{n=1}^{N} t_n = \;\sum_{n=1}^{N}\mu$

$\therefore\; \mu = \sum_{n=1}^{N}\frac{t_n}{N} - - - \;\; (10)$

Also,

$\frac{\partial L}{\partial W} = 0$

$L = -\frac{N}{2}\{\, dln(2\pi) + \ln|C| + tr(C^{-1}S)\}$

$\frac{dL}{dW} = -\frac{N}{2}[\frac{d}{dW}\ln|C| + \frac{d}{dW}tr(C^{-1}S)]$

$= -\frac{N}{2}[\frac{\partial}{\partial C}\ln|C|.\frac{\partial C}{\partial W} + \frac{\partial}{\partial C}tr(C^{-1}S).\frac{\partial C}{\partial W}]$

$\because \frac{\partial}{\partial x}\left(tr(x^{-1}A)\right) = -[x^{-1}A^T x^{-1}]^T$

$= -\frac{N}{2}\left[\frac{1}{|C|}.\frac{|C|}{C}.2(W) - [C^{-1}S^T C^{-1}]^T(2W)\right]$

$= -\frac{N}{2}(2)[\,C^{-1}W - [C^{-1}SC^{-1}]W\,] \qquad [\because C = C^T]$

$= -N[C^{-1}W - C^{-1}SC^{-1}W]$

$= N[C^{-1}SC^{-1}W - C^{-1}W]$

$$\frac{\partial L}{\partial W} = N(C^{-1}SC^{-1}W - C^{-1}W) = 0$$

$SC^{-1}W = W$

For above equation three solutions :- 1. W=0

2. C=S $\Longrightarrow WW^T + \sigma^2 I = S --- (11)$

3. $W \neq 0, C \neq S$

Also, SVD of W $\Longrightarrow$ $W = ULV^T - -- (12)$

Substitute (6) in (5) we get:-

$(ULV^T)(ULV^T)^T + \sigma^2 I = S$

$ULV^T VL^T U^T + \sigma^2 I = S$

$UL^2 U^T U + \sigma^2 I.U = SU$

$UL^2 + \sigma^2 U = SU$

$$U[L^2 + \sigma^2 I] = SU - - - (13)$$

For $l_j \neq 0 \; eq - (12) \; implies \; \Rightarrow Su_j = (\sigma^2 I + l_j^2) u_j$

$$l_j = (\lambda_j - \sigma^2)^{\frac{1}{2}}$$

For $l_j = 0, u_j is\ arbitary$

$$W = U_q(\Lambda_q - \sigma^2 I)^{\frac{1}{2}} R - - - (14)$$

$U_q - q\ columns\ of\ u\ (eigen\ vectors\ of\ s)\ \Lambda_q =$

$$\begin{cases} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \vdots \\ \lambda_q \end{cases} \quad coressponding\ eigen\ values\ in\ diagonal\ matrix\ \Lambda_q$$

Substitute eq (14) in eq(9)

$$L = -\frac{N}{2}\{ dln(2\pi) + \sum_{j=1}^{q'} \ln(\lambda_j) + \left(\frac{1}{\sigma^2}\right) \sum_{j=q'+1}^{d} \lambda_j + (d - q') \ln(\sigma^2) + q' \}$$

$$\frac{\partial L}{\partial \sigma} = 0$$

$$\sigma^2 = \left(\frac{1}{d - q'}\right) \sum_{j=q'+1}^{d} \lambda_j$$

Using Expectation Maximization algorithm :

We can use Expectation Maximization algorithm to obtain W

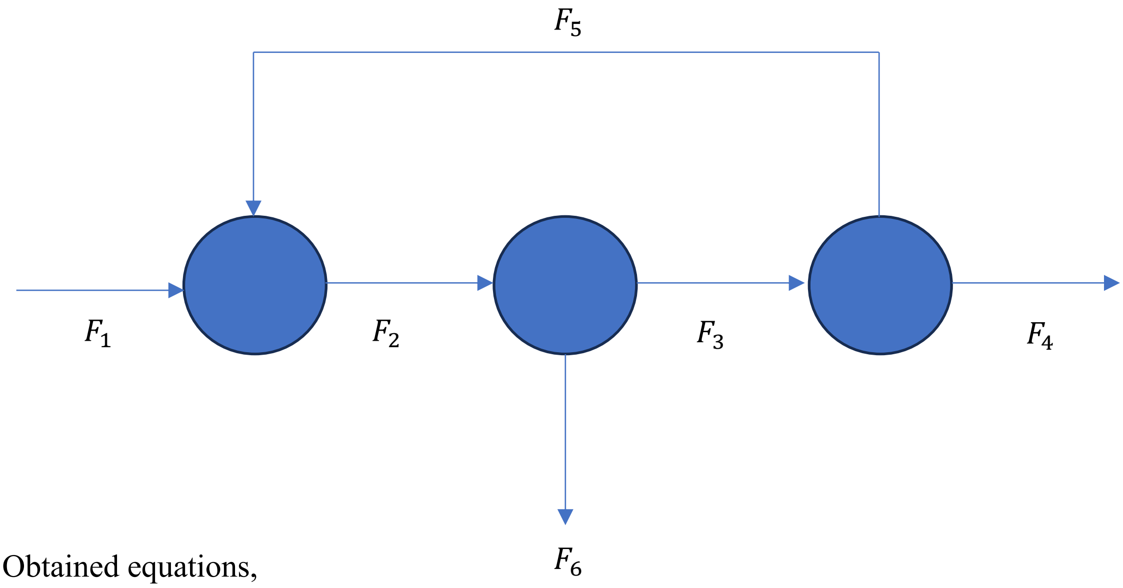
W obtained in this EM PPCA is

$$W = SW(\sigma^2 I + M^{-1} W^T S W)^{-1}$$

Where $M = W^T W + \sigma^2 I$

Solution Methodology

Generating dataset of Flow network



Obtained equations,

$F_5 = F_2 - F_1$……………………….. (1)

$F_6 = F_2 - F_3$………………………...(2)

$F_4 = F_3 - F_2 + F_1$ …………………...(3)

We can represent this equation as,

$$\begin{bmatrix} 1 & -1 & 0 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \\ F_5 \\ F_6 \end{bmatrix} = 0$$

We have generated the True dataset, by generating the 100 set of random numbers of F_1, F_2 *and* F_3 and then we calculated the corresponding F_4, F_5 *and* F_6 from the obtained equations 1,2 and 3.

Here, $F_5, F_6 > 0$, so $F_2 > F_1, F_3$

$F_4 > 0$, so $F_2 < F_1 + F_3$

So, we can decide the range of variables on this basis.

The range of $F_1 => (70\ , 100)\frac{K.mol}{hr}$

$F_2 => (\ 110, 130)\frac{K.mol}{hr}$

$F_3 => (\ 70, 100)\frac{K.mol}{hr}$

We have exported the True data ($data_{true}$) in to excel.

Then for $data_{true}$ we have added $noise$ which is our working dataset t,

$$data = data_{true} + noise$$

The noise is generated with the mean $= 0$.

Such that we obtain the working dataset.

Implementation of Principal Component Analysis

We are using python for simulation of the algorithm,

1) Import NumPy, Pandas , Matplotlib and PCA from sklearn.decomposition.
2) Read the dataset $\boldsymbol{data}$ and $\boldsymbol{noise}$
3) Get the NumPy array for $\boldsymbol{data}$ and transpose it , $\boldsymbol{data^T} = \boldsymbol{T}$
4) Making the t to mean centric, by $T = T - \bar{T}$
5) Constructing Covariance matrix S and Calculating Eigen Values and Vectors of S as **VAL** and **VEC**.
6) Plotting Singular Components vs Eigen values (From the graph we will compare the eigen values from that highest eigen values will be principal components, here it is **3**)
7) Getting Principal axes ($\boldsymbol{W_{PCA}}$) from the first 3 vectors of VAL.

8) Obtaining Row-echelon form for the transpose of $\boldsymbol{W_{PCA}}$.

Using SVD method:

1) Import NumPy, Pandas , Matplotlib and PCA from sklearn.decomposition.
2) Read the dataset $\boldsymbol{data}$ and $\boldsymbol{noise}$
3) Get the NumPy array for $\boldsymbol{data}$ and transpose it , $\boldsymbol{data^T} = \boldsymbol{T}$
4) Making the t to mean centric, by $T = T - \bar{T}$
5) Perform SVD for T^T, $T^T = U\Lambda V^T$
6) Plotting Singular Components vs Eigen values (Λ) (Highest eigen values will be the principal components, here it is **3**)
7) We will obtain V_1^T and V_2^T. (First 3 and last 3 vectors of V^T).
8) Row-echelon form of V_2^T determine the model equation.

Implementation of Probabilistic Principal Component Analysis

Using Maximum Likelihood:

1) Import NumPy, Pandas , Matplotlib and PCA from sklearn.decomposition.
2) Read the dataset $\boldsymbol{data}$ and $\boldsymbol{noise}$
3) Get the NumPy array for $\boldsymbol{data}$ and transpose it , $\boldsymbol{data^T} = \boldsymbol{T}$
4) Making the t to mean centric, by $T = T - \bar{T}$
5) Constructing Covariance matrix S and Calculating Eigen Values and Vectors of S as **VAL** and **VEC**.
6) Plotting Singular Components vs Eigen values (From the graph we will compare the eigen values from that highest eigen values will be principal components, here it is 3)
7) Getting Principal axes ($\boldsymbol{U_q}$) from the first 3 vectors of VAL.
8) Getting variance for $\boldsymbol{noise}$ as **var_error**.

9) Getting square root of difference between highest 3 eigen values of VAL and corresponding var_error (from Equation(8)) as $\boldsymbol{L_q}$.

$$L_q = \left(\Lambda_q - \sigma^2 I\right)^{\frac{1}{2}}$$

$$W_{ML} = U_q L_q --- (13)$$

10) According to the above equation we can obtain $\boldsymbol{W_{ML}}$ by multiplying the U_q and L_q.

11) Obtaining Row-echelon form for the transpose of $\boldsymbol{W_{ML}}$.

Using Expectation and Maximization:

1) Import NumPy, Pandas , Matplotlib and PCA from sklearn.decomposition.
2) Read the dataset $\boldsymbol{data}$ and $\boldsymbol{noise}$
3) Get the NumPy array for $\boldsymbol{data}$ and transpose it , $\boldsymbol{data^T} = \boldsymbol{T}$
4) Making the t to mean centric, by $T = T - \bar{T}$
5) Constructing Covariance matrix S and Calculating Eigen Values and Vectors of S as **VAL** and **VEC**.
6) Plotting Singular Components vs Eigen values (From the graph we will compare the eigen values from that highest eigen values will be principal components, here it is 3)
7) Getting Principal axes ($\boldsymbol{W}$) from the first 3 vectors of VAL.
8) Obtaining $\boldsymbol{M}$ by addition of var_error and $\boldsymbol{W^T W}$. (from equation(10))

$$M = W^T W + \sigma^2 I --- (10)$$

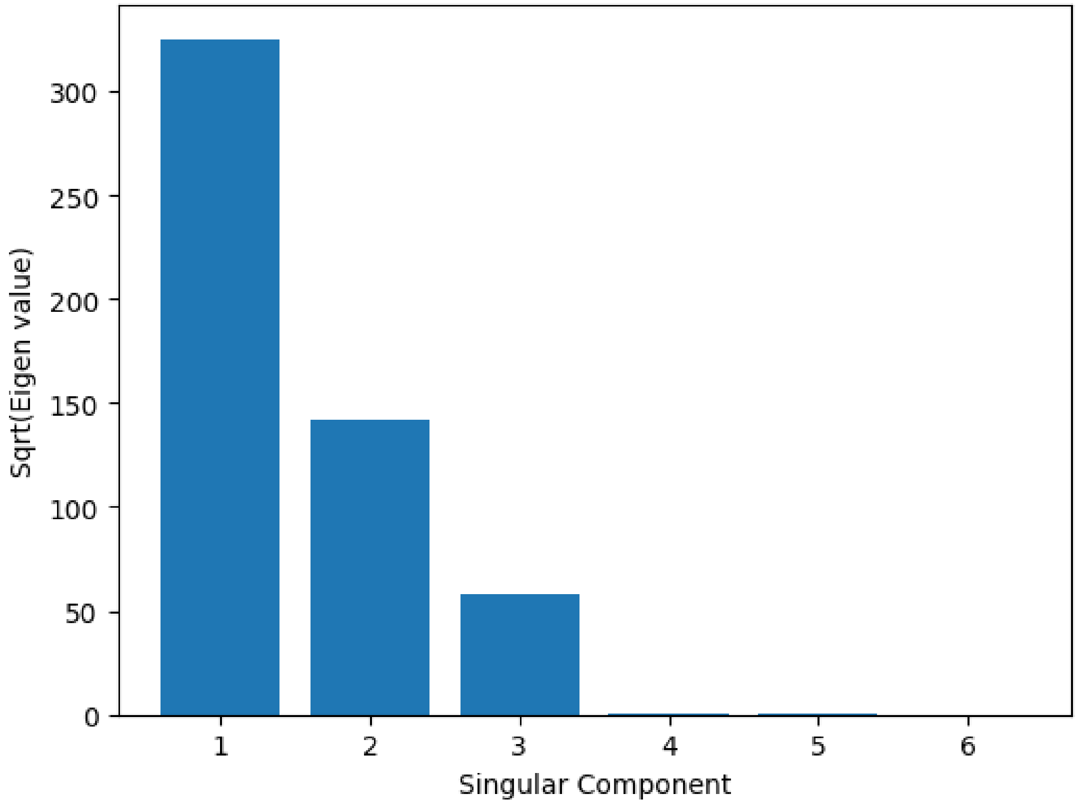
9) Obtained $\boldsymbol{W_{EM}}$ by following the equation(12)

$$W_{EM} = SW(\sigma^2 I + M^{-1} W^T S W)^{-1} --- (12)$$

10) Obtaining Row-echelon form for the transpose of $\boldsymbol{W_{EM}}$.

Simulation Results

1) Number of Principal Components



$$Graph - 1$$

From the graph we can say that there are **3 principal components**.

2) We obtain Principal $\boldsymbol{W}$ axes using PCA, PPCA (MLE) and PPCA(EM)

<u>For Principal Component analysis (PCA)</u>

$$\boldsymbol{W_{PCA}} = \begin{bmatrix} 0.27411196 & 0.50933345 & -0.41426578 \\ -0.12930805 & 0.00528618 & -0.69586624 \\ 0.28988268 & -0.4946646 & -0.41814181 \\ 0.69160606 & 0.01487125 & -0.12387658 \\ -0.40307981 & -0.50576227 & -0.28266645 \\ -0.42815631 & 0.48973646 & -0.2721502 \end{bmatrix}$$

$\boldsymbol{Rowechelon(W_{PCA}^T)}$

$$= \begin{bmatrix} 1 & 0 & 0 & 1.000101428569150 & -1.00101260169317 & -0.0241598101499193 \\ 0 & 1 & 0 & -1.01158150896018 & 1.00067360205053 & 1.00885568491893 \\ 1 & 0 & 1 & 0.988885102793565 & 0.00243156527047538 & -1.00413263434885 \end{bmatrix}$$

For Probabilistic Principal Component analysis (PCA) using MLE

$$W_{PCA} = \begin{bmatrix} 4.94125969 & 6.05801857 & -3.14249203 \\ -2.33096226 & 0.06287395 & -5.27862597 \\ 5.2255493 & -5.88354706 & -3.17189435 \\ 12.46718705 & 0.17687887 & -0.93968942 \\ -7.26608936 & -6.01554286 & -2.14422021 \\ -7.71812901 & 5.82493176 & -2.06444718 \end{bmatrix}$$

$$\boldsymbol{Rowechelon}(\boldsymbol{W_{ML}^{T}})$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1.00010142856915 & -1.00101260169317 & -0.0241598101499192 \\ 0 & 1 & 0 & -1.01158150896018 & 1.00067360205053 & 1.00885568491893 \\ 1 & 0 & 1 & 0.988885102793565 & 0.00243156527047528 & -1.00413263434885 \end{bmatrix}$$

For Probabilistic Principal Component analysis (PCA) using EM

$$W_{EM} = \begin{bmatrix} 0.35348392 & 0.64367529 & -0.50754435 \\ -0.16675053 & 0.00668047 & -0.85255165 \\ 0.3738212 & -0.62513738 & -0.51229312 \\ 0.89186774 & 0.0187937 & -0.15176938 \\ -0.51979574 & -0.63916217 & -0.34631332 \\ -0.55213339 & 0.6189094 & -0.33342917 \end{bmatrix}$$

$$\boldsymbol{Rowechelon}(\boldsymbol{W_{EM}^{T}})$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1.00010142856915 & -1.00101260169317 & -0.0241598101499207 \\ 0 & 1 & 0 & -1.01158150896018 & 1.00067360205053 & 1.00885568491893 \\ 1 & 0 & 1 & 0.988885102793565 & 0.00243156527047495 & -1.00413263434885 \end{bmatrix}$$

Conclusion

- From the Graph-1 we can conclude that the whole data can be represent using 3 principal axes.
- The Principal axes obtained from PCA, PPCA(using MLE) and PPCA(using EM) are almost spanning the same space,

  So, $\boldsymbol{Rowechelon}(W_{PCA}^{T}) = \boldsymbol{Rowechelon}(W_{ML}^{T}) = \boldsymbol{Rowechelon}(W_{EM}^{T})$

References

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. JOURNAL OF THE ROYALSTATISTICAL SOCIETY, SERIES B, 61(3):611–622, 1999.

R. Iyer and J. Magnus Matrix Differential Calculus with Applications in Statistics and Econometrics. ECONOMICS, MATHEMATICS 1991.

Contribution Table

Roll No	**Name**	**Contribution %**
CH20B009	Kaloori Shiva Prasad	33.33%
CH20B013	Koralapati Hemanth	33.33%
CH20B014	Kuruva Likith Kumar	33.33%