Principal Component Analysis

Arkaprava Sinha
Purnendu Ghosh
Rohan Khaitan
Subhasish Basak

CHENNAI MATHEMATICAL INSTITUTE

April 17, 2019

Outline

Motivation

- Machine learning problems are often associated with high dimensional data. In many practical applications it is of interest to reduce the dimensionality of the data
- Data visualization
- Data exploration: For investigating the effective dimensionality of the data
- Model training: Reduction in dimensions also boosts up the model training

Motivation

- Principal Component Analysis (PCA) is a technique that can be used to simplify a dataset
- It is a linear transformation that chooses a new coordinate system for the data set such that greatest variance by any projection of the data set comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on.
- PCA can be used for reducing dimensionality by eliminating the later principal components.

Motivation:Example

- Consider the following 3D points
$$\begin{pmatrix}1\\2\\3\end{pmatrix}, \begin{pmatrix}2\\4\\6\end{pmatrix}, \begin{pmatrix}4\\8\\12\end{pmatrix}, \begin{pmatrix}3\\6\\9\end{pmatrix}, \begin{pmatrix}5\\10\\15\end{pmatrix}, \begin{pmatrix}6\\12\\18\end{pmatrix}$$
- If each component is stored in a byte, we need $18 = 3 \times 6$ bytes

Motivation:Example

- Looking closer, we can see that all the points are related geometrically, in fact they are all the same point, scaled by a factor.
- They can be stored using only 9 bytes (50% savings!). Store one point (3 bytes) + the multiplying constants (6 bytes)

Geometric Interpretation

- View each point in 3D space

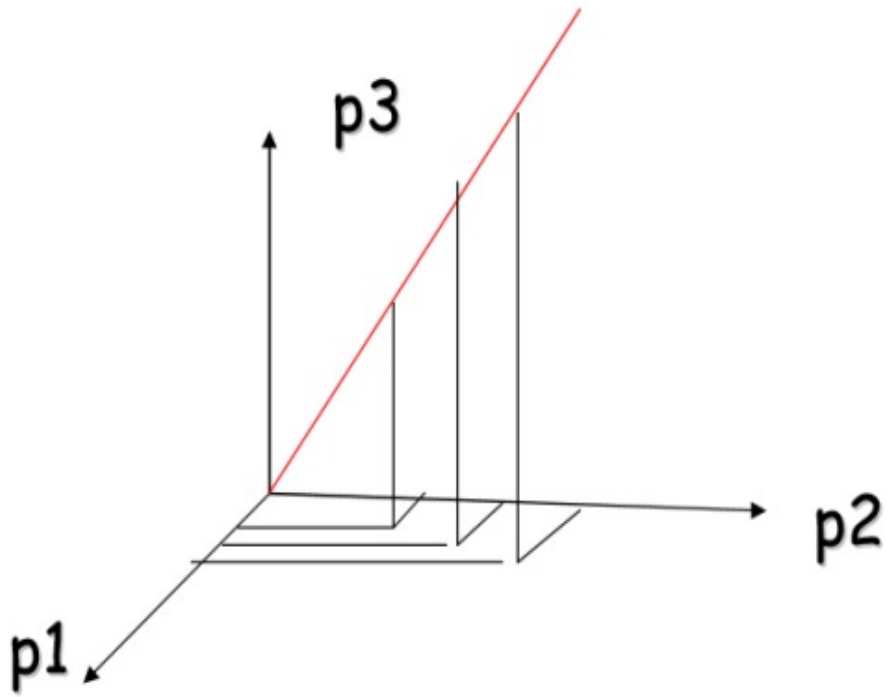


Figure: Using 3 given dimensions

- But in this example, all the points happen to belong to a line: a 1D subspace of the original 3D space.

Geometric Interpretation

- Consider a new coordinate system where one of the axes Consider a new coordinate system where one of the axes is along the direction of the line:

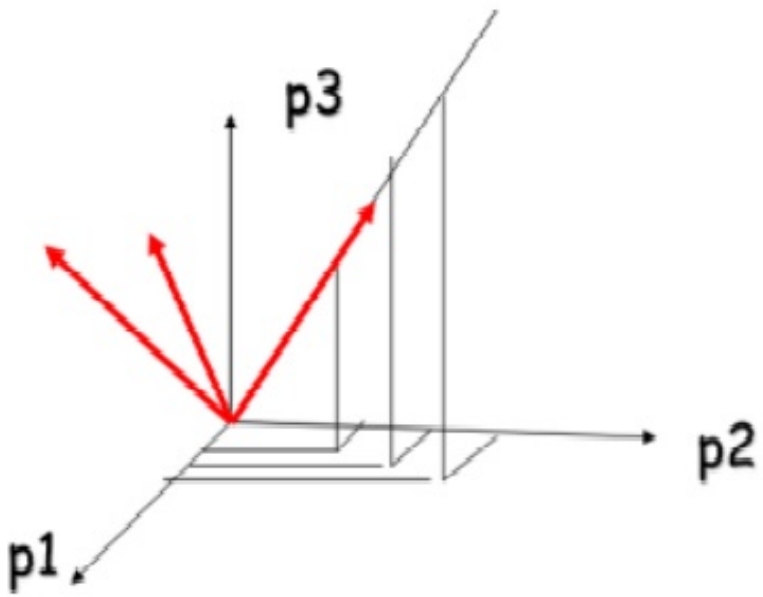


Figure: Using 3 optimized dimensions

Geometric Interpretation

- In this coordinate system, every point has only one non non-zero coordinate, we only only need to store the direction of the line (a 3 bytes image) and the nonzero coordinate for each of the points (6 bytes).

  i.e. $\begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ and $\begin{pmatrix} 2 & 4 & 3 & 5 & 6 \end{pmatrix}$

Principal Component Analysis

- Given a set of points, how do we know if they can be compressed like in the previous example?
- The answer is to look into the correlation between the points
- The tool for doing this is called PCA

A Geometric Approach

- Let X be a d-dimensional random vector and X1,..., Xn be n independent copies of X.
- This problem of dimensionality reduction can be seen as the problem of defining a map

$$M : X = R^D \rightarrow R^k, k << D,$$

  according to some suitable criterion.
- In the following data reconstruction will be our guiding principle.

A Geometric Approach

We recall that, if

- $w \in R^D, ||w|| = 1$

then $(w^{\mathrm{T}}x)w$ is the orthogonal projection of x on w

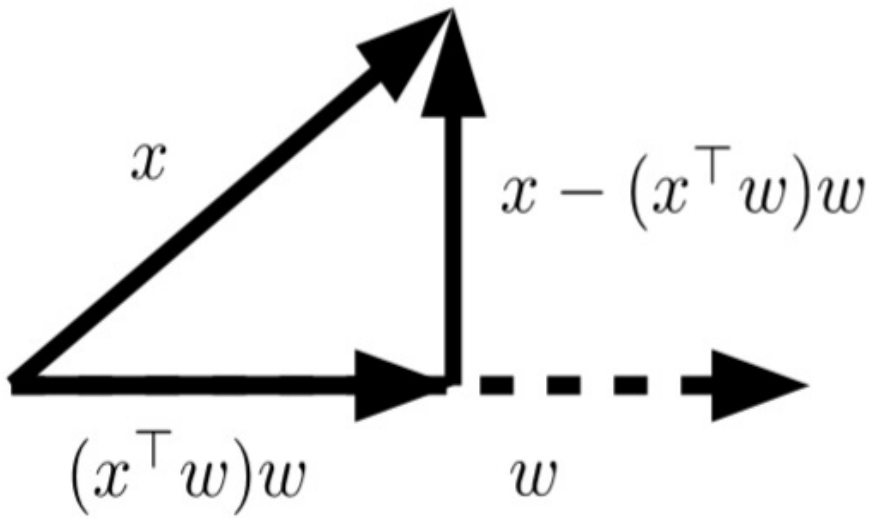


Figure: Vector projection

A Geometric Approach

- First, consider k $= 1$. The associated reconstruction error is $||x - (w^{\mathrm{T}}x)w||^2$ (that is how much we lose by projecting x along the direction w).

  Our problem is to find the direction p allowing the best reconstruction of the training set.

A Geometric Approach

- Let $S^{D-1} = \{w \in R^D \mid ||w|| = 1\}$ is the sphere in D dimensions. Consider the empirical reconstruction minimization problem,

$$\min_{w \in S^{D-1}} \frac{1}{n} \sum_{i=1}^{n} ||x_i - (w^{\mathrm{T}} x_i) w||^2 \tag{1}$$

The solution p to the above problem is called the first principal component (PC1) of the data

A Geometric Approach

A direct computation shows that

$$||x_i - (w^{\mathrm{T}} x_i) w||^2 = ||x_i||^2 - (w^{\mathrm{T}} x_i)^2$$

Then,problem

$$\min_{w \in S^{D-1}} \frac{1}{n} \sum_{i=1}^{n} ||x_i - (w^{\mathrm{T}} x_i) w||^2 \tag{2}$$

is equivalent to

$$\max_{w \in S^{D-1}} \frac{1}{n} \sum_{i=1}^{n} (w^{\mathrm{T}} x_i)^2 \tag{3}$$

To solve this optimization problem we setup the mathematical preliminaries

Statistical keywords & methodologies used

- Let X be a d-dimensional random vector and $X_1, ..., X_n$ be n independent copies of X
- Next write $X_i = (X_i^1, \dots X_i^d)^{\mathrm{T}}$, $i = 1, \dots n$

Denote by $\mathbf{X}$ the random n matrix

$$\mathbf{X} = \begin{bmatrix} \dots X_1^{\mathrm{T}} \dots \\ \vdots \\ \dots X_n^{\mathrm{T}} \dots \end{bmatrix}$$

Statistical keywords & methodologies used

Assume that $\mathbb{E}[||\mathbf{x}||_2{}^2] < \infty$

Mean of $\mathbf{X}$:

$$\mathbb{E}[\mathbf{X}] = (\mathbb{E}[\mathbf{X}^1], \ldots \mathbb{E}[\mathbf{X}^d])^{\mathrm{T}}$$

Covariance matrix of X:the matrix $\mathbf{\Sigma} = (\sigma_{j,k})_{j,k=1,\ldots d}$ where

$$\sigma_{j,k} = \boldsymbol{cov}(\mathbf{X^j}, \mathbf{X^k})$$

It is easy to see that

$$\mathbf{\Sigma} = \mathbb{E}[\mathbf{X}\mathbf{X}^{\mathrm{T}}] - \mathbb{E}[\mathbf{X}][\mathbf{X}]^{\mathrm{T}} = \mathbb{E}[(\mathbf{X} - \mathbb{E}(\mathbf{X})(\mathbf{X} - \mathbb{E}(\mathbf{X})^{\mathrm{T}}]$$

Statistical keywords & methodologies used

Empirical mean of $\mathbf{X}_1, \ldots \mathbf{X}_n$:

$$\bar{\mathbf{X}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i = (\bar{\mathbf{X}}^1, \ldots \bar{\mathbf{X}}^d)^{\mathrm{T}}$$

Empirical covariance of $\mathbf{X}_1, \ldots \mathbf{X}_n$:the matrix

$S = (s_{j,k})_{j,k=1,\ldots d}$ where $s_{j,k}$ is the empirical covariance of $\mathbf{X}_i^j, \mathbf{X}_i^k, i = 1, \ldots n$

It is easy to see that

$$S = \frac{1}{n} \sum_{i=1}^{n} \mathbf{X}_i \mathbf{X}_i^{\mathrm{T}} - \bar{\mathbf{X}} \bar{\mathbf{X}}^{\mathrm{T}} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^{\mathrm{T}}$$

Statistical keywords & methodologies used

Theorem

If $u \in R^D$, then $u^{\mathrm{T}}\mathbf{\Sigma}u$ is the variance of $\mathbf{u}^{\mathrm{T}}\mathbf{X}$;

Theorem

If $u \in R^D$, then $u^{\mathrm{T}}\mathbf{S}u$ is the sample variance of $\mathbf{u}^{\mathrm{T}}\mathbf{X_1}, \ldots \mathbf{u}^{\mathrm{T}}\mathbf{X_n}$;

In particular, $\mathbf{u}^{\mathrm{T}}\mathbf{S}\mathbf{u}$ measures how spread (i.e., diverse) the points are in direction $\mathbf{u}$.

Statistical keywords & methodologies used

- In particular, $\mathbf{\Sigma}$ and $\mathbf{S}$ are symmetric, positive semi-denite. Any real symmetric matrix $\mathbf{A} \in R^{d \times d}$ has the decomposition

$$\mathrm{A}=\mathrm{PDP}^{\mathrm{T}}$$

where: $\mathbf{P}$ is a $d \times d$ diagonal matrix,i.e. $\mathbf{P}\mathbf{P}^{\mathrm{T}} = \mathbf{P}^{\mathrm{T}}\mathbf{P} = \mathbf{I}_d$; $\mathbf{D}$ is diagonal.

The diagonal elements of $\mathbf{D}$ are the eigenvalues of $\mathbf{A}$ and the columns of $\mathbf{P}$ are the corresponding eigenvectors of $\mathbf{A}$.

Principal Component Analysis: Heuristics

Write $\mathbf{S} = \mathbf{P}\mathbf{D}\mathbf{P}^{\mathrm{T}}$,where
$\mathbf{P} = (\mathbf{V}_1, \ldots \mathbf{V}_d)$ is an orthogonal matrix,
such that $||\mathbf{V}_j||_2 = 1$, $\mathbf{V}_j^{\mathrm{T}}\mathbf{V}_k = 0, \forall j \neq k$.

$$\mathbf{D} = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_d \end{bmatrix}$$

with $\lambda_1 \geqslant \cdots \geqslant \lambda_d$

Note that $\mathbf{D}$ is the empirical covariance matrix of the $\mathbf{P}^{\mathrm{T}}\mathbf{X}_i$'$s$, $i = 1, \ldots n$

In particular, $\lambda_\mathbf{1}$ is the empirical variance of the $\mathbf{V}_1^{\mathrm{T}}\mathbf{X}_i$'s; $\lambda_\mathbf{2}$ is the empirical variance of the $\mathbf{V}_2^{\mathrm{T}}\mathbf{X}_i$'$s$, etc ...

Principal Component Analysis: Heuristics

So, each λ_j measures the spread of the cloud in the direction $\mathbf{V}_j$.
In particular, $\mathbf{V}_1$ is the direction of maximal spread.

With this much of mathematical framework we recall our optimization problem:

$$\max_{w \in S^{D-1}} \frac{1}{n} \sum_{i=1}^{n} \left(w^{\mathrm{T}} x_i\right)^2 \tag{4}$$

PCA as an Eigenproblem

A further manipulation shows that PCA corresponds to an eigenvalue problem.

Using the symmetry of the inner product,

$$\frac{1}{n}\sum_{i=1}^{n}(w^{\mathrm{T}}x_i)^2 = \frac{1}{n}\sum_{i=1}^{n} w^{\mathrm{T}}x_i w^{\mathrm{T}}x_i = \frac{1}{n}\sum_{i=1}^{n} w^{\mathrm{T}}x_i x_i^{\mathrm{T}}w = w^{\mathrm{T}}(\frac{1}{n}\sum_{i=1}^{n} x_i x_i^{\mathrm{T}})w \tag{5}$$

Then, we can consider the problem

$$\max_{w\in S^{D-1}} w^{\mathrm{T}}\boldsymbol{C}_n w, \boldsymbol{C}_n = \frac{1}{n}\sum_{i=1}^{n} x_i x_i^{\mathrm{T}} \tag{6}$$

PCA as an Eigenproblem

- We make two observations: The covariance matrix $\mathbf{C}_n$ is symmetric and positive semi-definite.
- The objective function of PCA can be written as

$$\frac{\mathbf{W}^{\mathrm{T}}\mathbf{C}_n\mathbf{W}}{\mathbf{W}^{\mathrm{T}}\mathbf{W}}$$

- the so called Rayleigh quotient.
- Note that, if $\mathbf{C}_n\mathbf{u} = \lambda\mathbf{u}$ then $\mathbf{u}^{\mathrm{T}}\mathbf{C}_n\mathbf{u} = \lambda$, since u is normalized
- Computing the first principal component of the data reduces to computing the biggest eigenvalue of the covariance matrix and the corresponding eigenvector.

PCA as an Eigenproblem

Beyond the First Principal Component:

We discuss how to consider more than one principle component ($\mathbf{k}>1$)

$$M : X = R^D \rightarrow R^k, k << D,$$

The idea is simply to iterate the previous reasoning

PCA as an Eigenproblem

The idea is to consider the one dimensional projection that can best reconstruct the residuals

$$r_i = x_i - (p^{\mathrm{T}} x_i) p_i$$

An associated minimization problem is given by

$$\min_{w \in S^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} ||r_i - (w^{\mathrm{T}} r_i) w||^2$$

(note: the constraint $w \perp p$)

PCA as an Eigenproblem

Note that for all $i = 1,\ldots,n$

$$||r_i - (w^{\mathrm{T}} r_i)w||^2 = ||r_i||^2 - (w^{\mathrm{T}} r_i)^2 = ||r_i||^2 - (w^{\mathrm{T}} x_i)^2$$

since $w \perp p$

Then, we can consider the following equivalent problem

$$\max_{w \in S^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} \left(w^{\mathrm{T}} x_i\right)^2 = w^{\mathrm{T}} \boldsymbol{C}_n w$$

PCA as an Eigenproblem

$$\max_{w \in S^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} \left(w^{\mathrm{T}} x_i\right)^2 = w^{\mathrm{T}} \boldsymbol{C}_n w$$

Again, we have to maximize the Rayleigh quotient of the covariance matrix with the extra constraint w $\perp$ p

PCA as an Eigenproblem

$$\max_{w \in S^{D-1}, w \perp p} \frac{1}{n} \sum_{i=1}^{n} \left(w^{\mathrm{T}} x_i\right)^2 = w^{\mathrm{T}} \boldsymbol{C}_n w$$

Again, we have to maximize the Rayleigh quotient of the covariance matrix with the extra constraint w $\perp$ p

Similarly to before, it can be proved that the solution of the above problem is given by the second eigenvector of C_n, and the corresponding eigenvalue.

PCA Algorithm

- **Input: $x_1, ..., x_1$: cloud of n points in dimension d.**
- **Step 1: Compute the empirical covariance matrix.**
- **Step 2: Compute the decomposition $S = PDP^T$, where $D = diag(\lambda_1, \ldots, \lambda_d)$,with $\lambda_1 \geqslant \cdots \geqslant \lambda_d$ and $P = (V_1, \ldots V_d)$ is an orthogonal matrix.**
- **Step 3: Choose k<d and set $P_k = (V_1, \ldots V_d) \in R^{d \times k}$.**
- **Output:$Y_1, \ldots, Y_n$ where $Y_i = P_k^T X_i \in R^k$, i=1,. . . ,n**

PCA Algorithm

Question: How to choose k ?

Experimental Rule:Take k where there is an inflection point in the sequence: $\lambda_1, \ldots, \lambda_d$ (Scree plot)

Define a criterion: Take k such that

$$\frac{\lambda_1 + \cdots + \lambda_k}{\lambda_1 + \cdots + \lambda_d} \geqslant 1 - \alpha$$

for some $\alpha \in (0, 1)$ that determines the approximation error that the practitioner wants to achieve.

Remark: $\lambda_1 + \cdots + \lambda_k$ s called the variance explained by the PCA and $\lambda_1 + \cdots + \lambda_d = Tr(S)$ is the total variance.

Implementation

PCA Implementation :

On real life data-

- Main Purpose-To identify patterns and finding patterns to reduce the dimensions of the dataset with minimal loss of information.
- Desired Outcome- To project a feature space onto a smaller subspace that represents our data “well”.
- To reduce the computational costs and the error of parameter estimation by reducing the number of dimensions of our feature space by extracting a subspace that describes our data “best”.

PCA Implementation :

- In PCA, we are interested to find the directions (components) that maximize the variance in our dataset.
- Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analysing data

Steps as previously discussed :

What to do with the data!

- Take the whole dataset consisting of d-dimensional samples ignoring the class labels
- Compute the d-dimensional mean vector (i.e., the means for every dimension of the whole dataset)
- Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set
- Compute eigenvectors $(\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_d})$ and corresponding eigenvalues $(\lambda_\mathbf{1}, \lambda_\mathbf{2} \ldots, \lambda_\mathbf{d})$

Steps as previously discussed :

- Sort the eigenvectors by decreasing eigenvalues and choose k eigenvectors with the largest eigenvalues to form a $\mathbf{d} \times k$ dimensional matrix W (where every column represents an eigenvector)
- Use this $\mathbf{d} \times k$ eigenvector matrix to transform the samples onto the new subspace. This can be summarized by the mathematical equation: $\mathbf{y} = \mathbf{W}^{\mathrm{T}} \times x$ (where x is a $\mathbf{d} \times 1$ -dimensional vector representing one sample, and y is the transformed $\mathbf{k} \times 1$-dimensional sample in the new subspace.)

PCA Algorithm:

2-D Data:

Data =

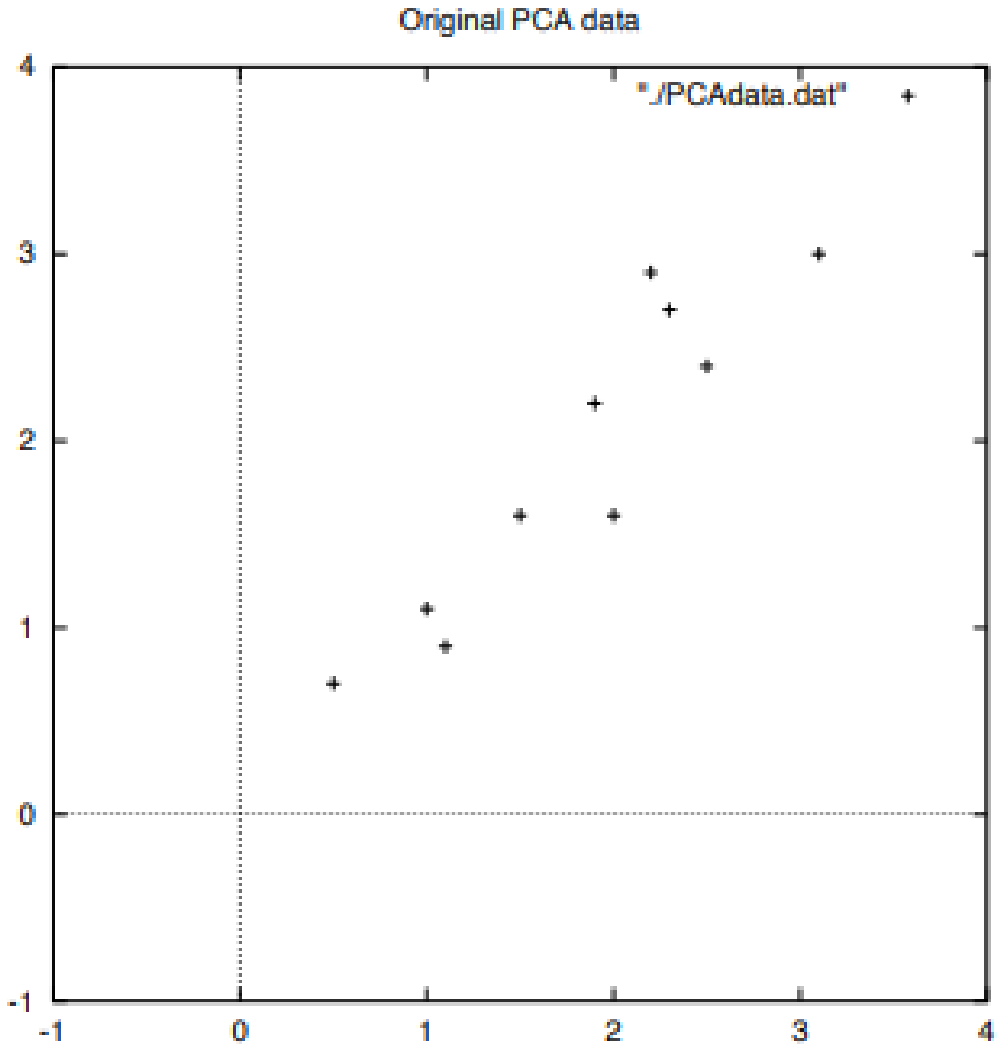
x	y
2.5	2.4
0.5	0.7
2.2	2.9
1.9	2.2
3.1	3.0
2.3	2.7
2	1.6
1	1.1
1.5	1.6
1.1	0.9

DataAdjust =

x	y
.69	.49
-1.31	-1.21
.39	.99
.09	.29
1.29	1.09
.49	.79
.19	-.31
-.81	-.81
-.31	-.31
-.71	-1.01

Data Plot:

We can visualize!



Computation:

We compute the co-variance matrix,eigen-values and eigen-vectors.

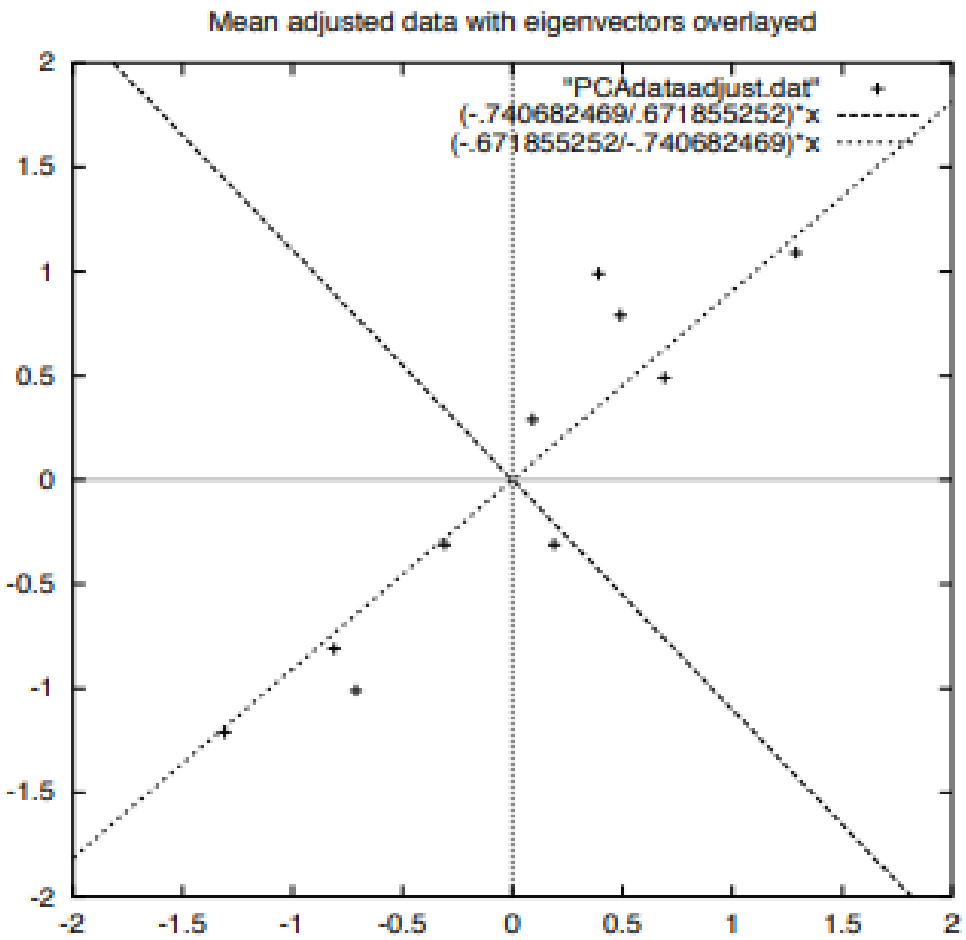
$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

Visualisaton:

A plot of the data with the eigenvectors of the covariance matrix overlayed on top.



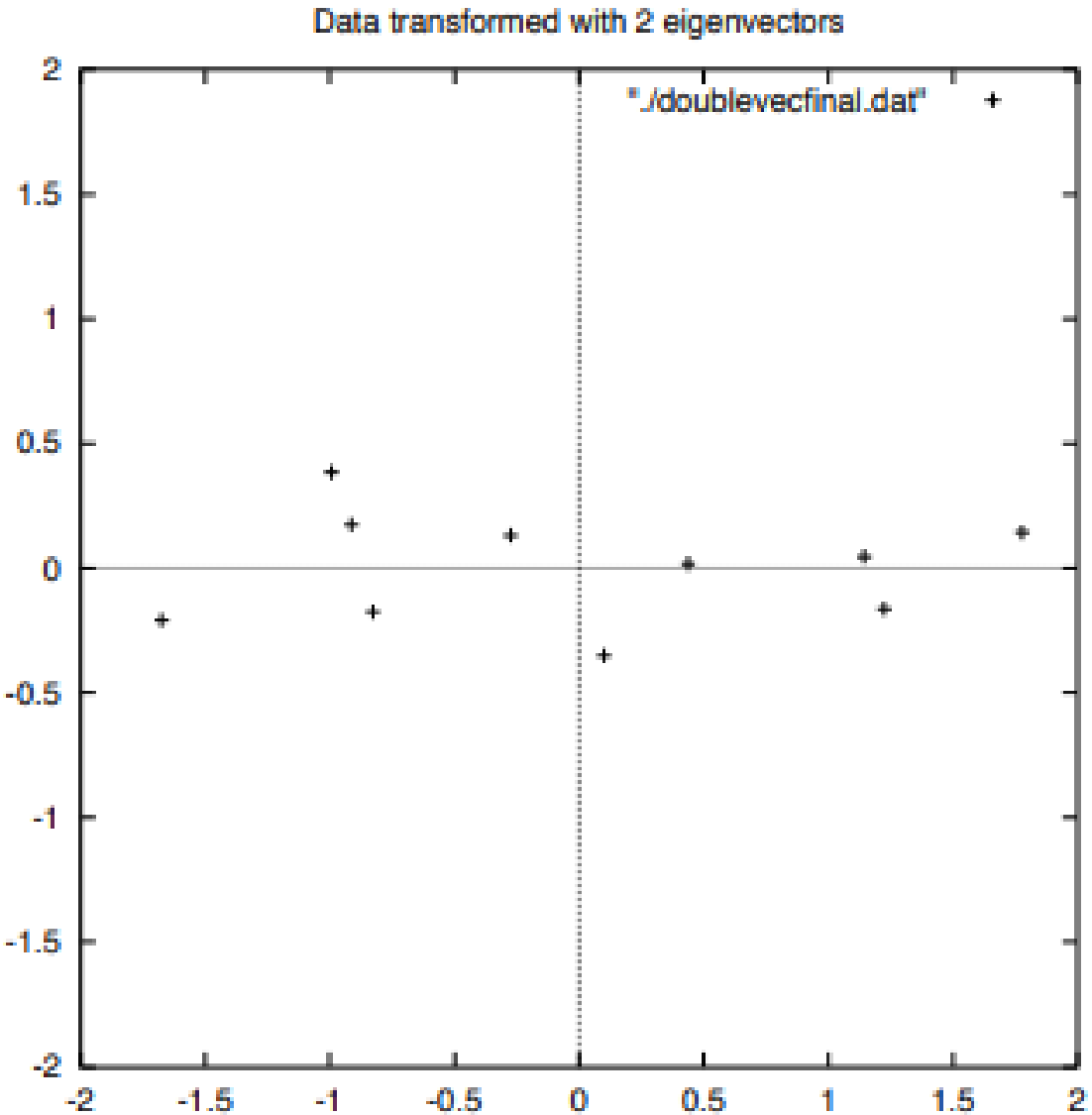
Transformed Data:

The table of data by applying the PCA analysis using both eigenvectors.

Transformed Data=

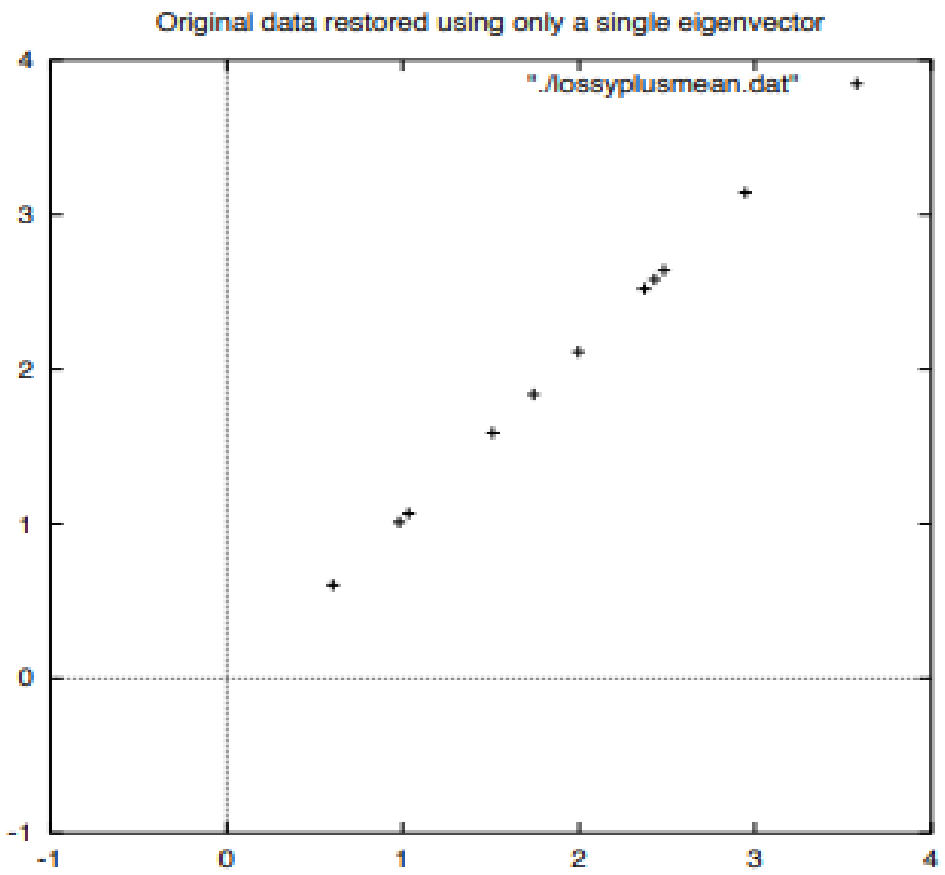
x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287

Transformed Data plot in new axis:



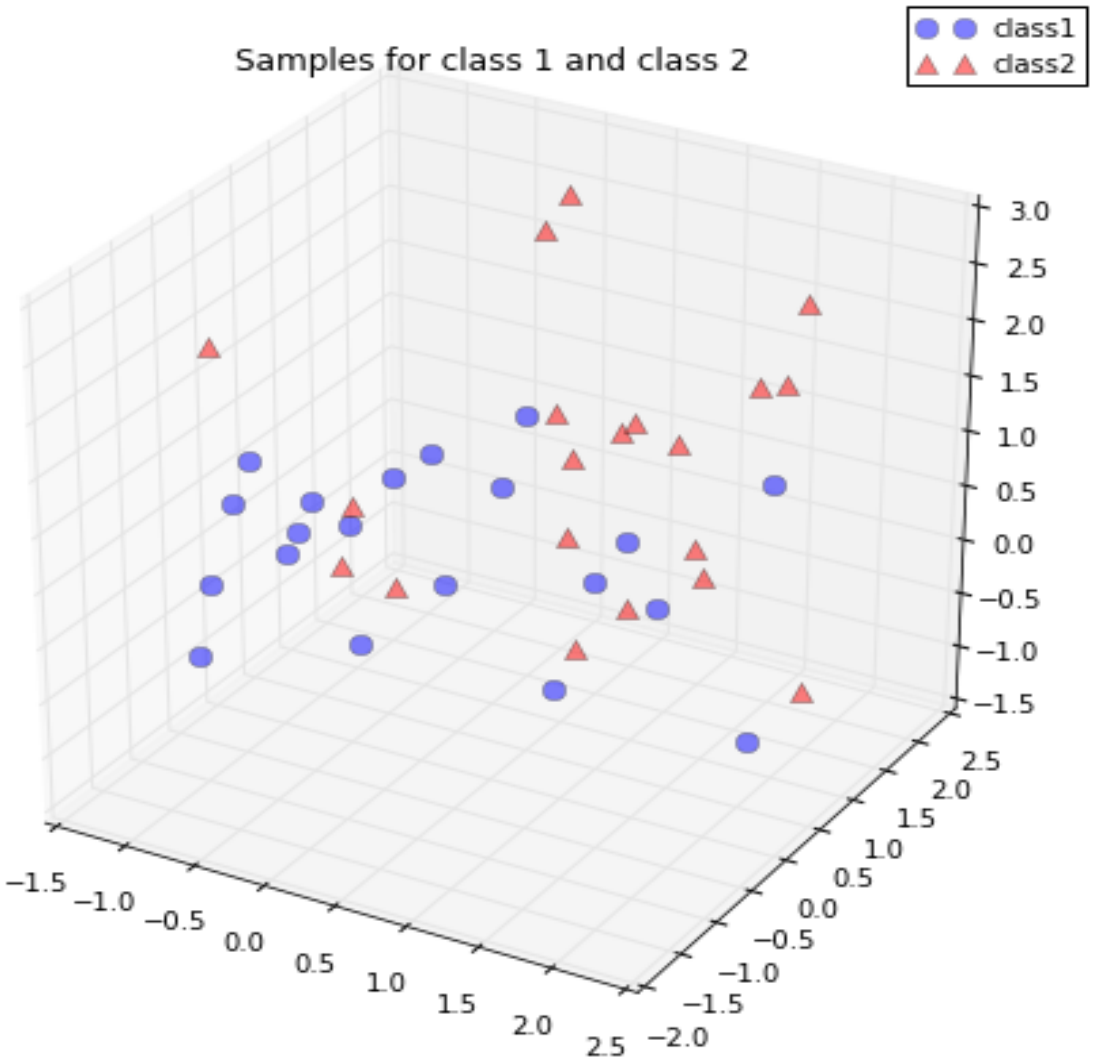
Restored Data using a single eigenvector:

The variation along the principal eigenvector has been kept.



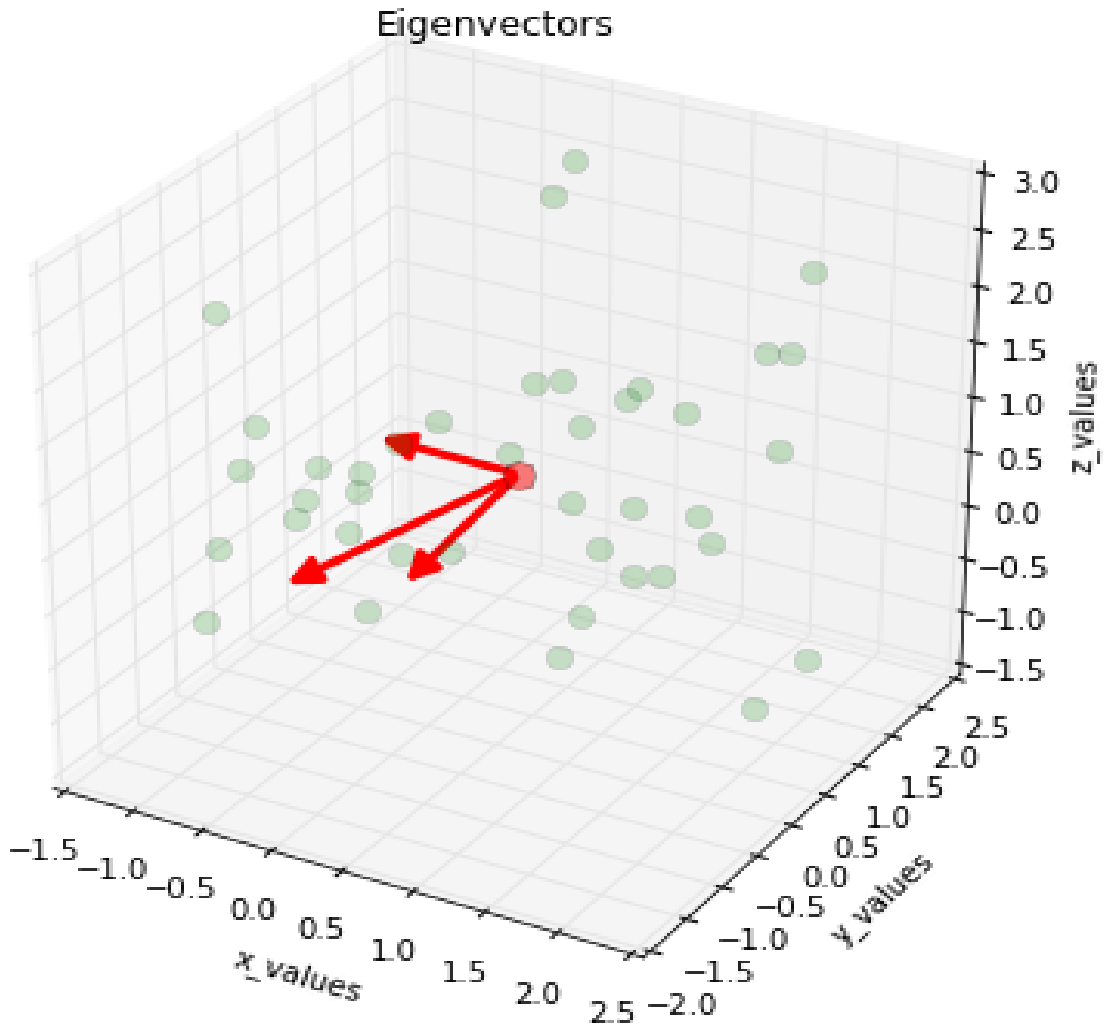
Example:

3D data:



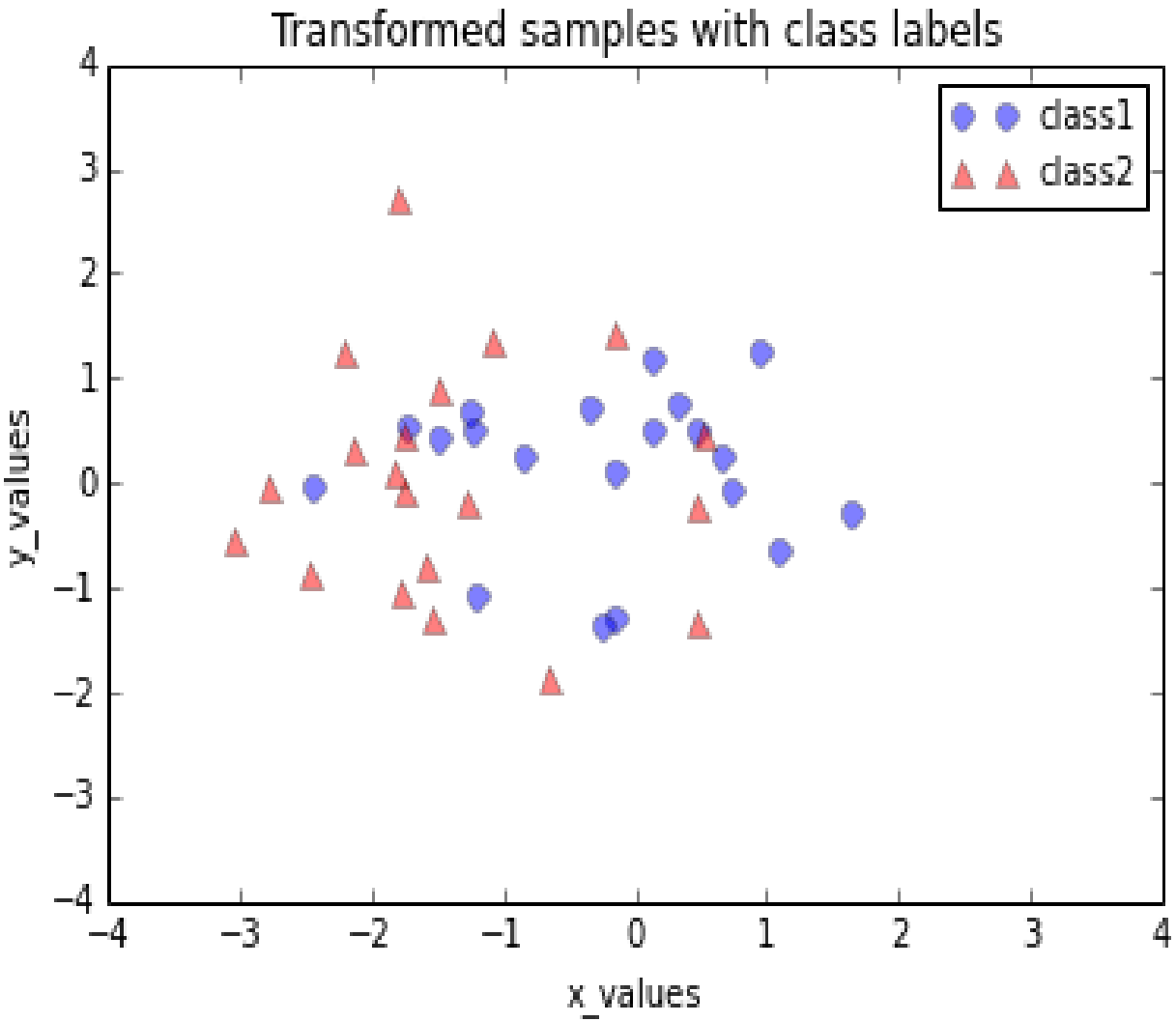
Example:

Eigen Vectors:

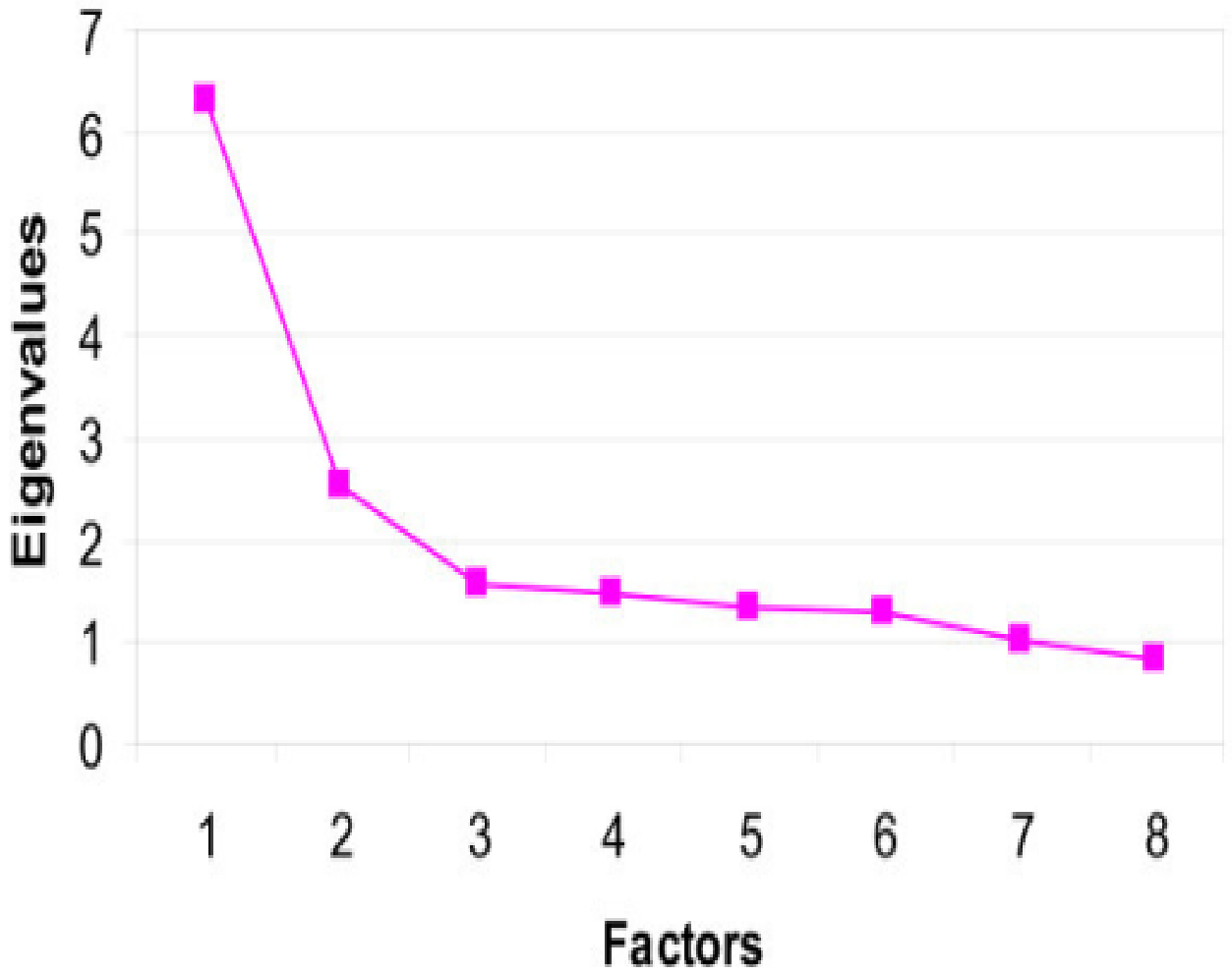


Example:

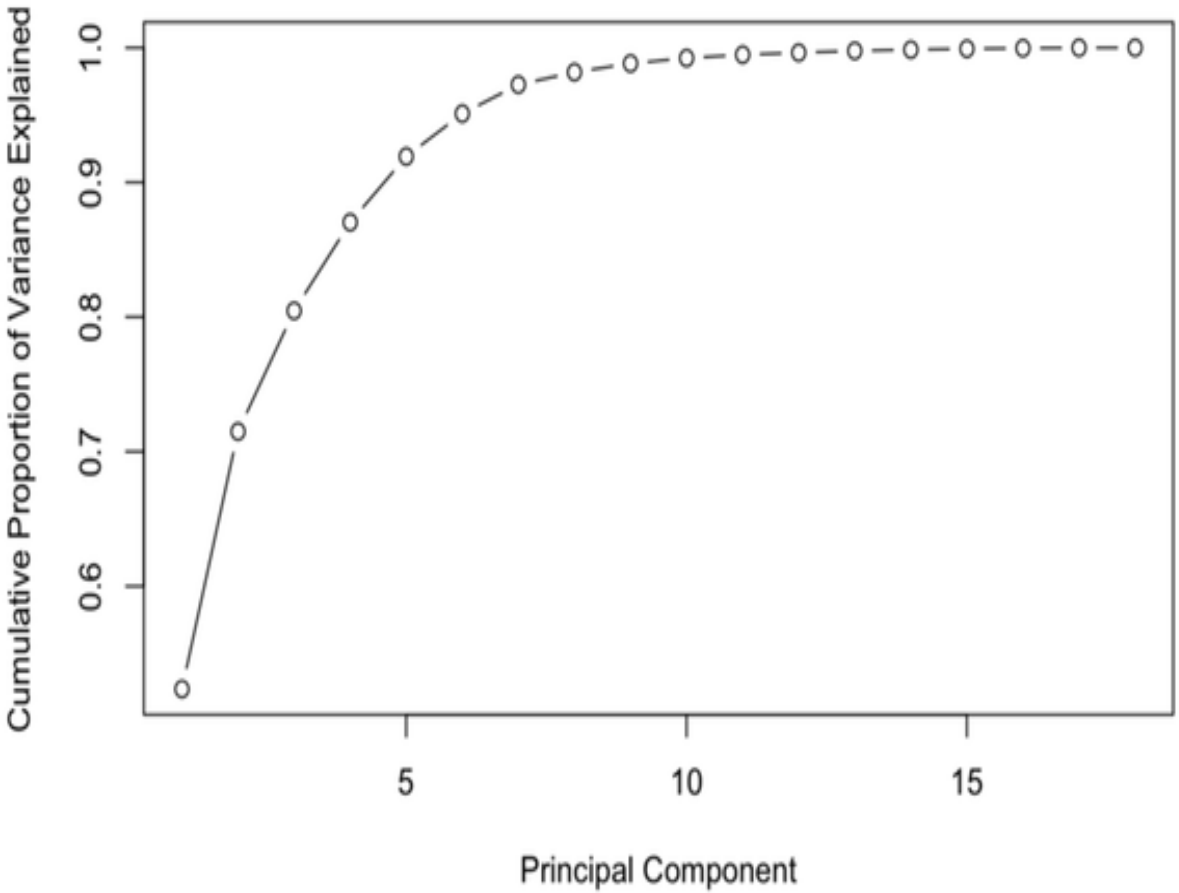
Projected on 2D:



How many dimensions!



How many dimensions!



Implementation in Jupyter Notebook

Summary :

- Dimensionality Reduction is important!
- Visualisation!
- Other methods - LDA,t-SNE

Thank You