2D PCA

Oshin Gansi
KCE074BCT028

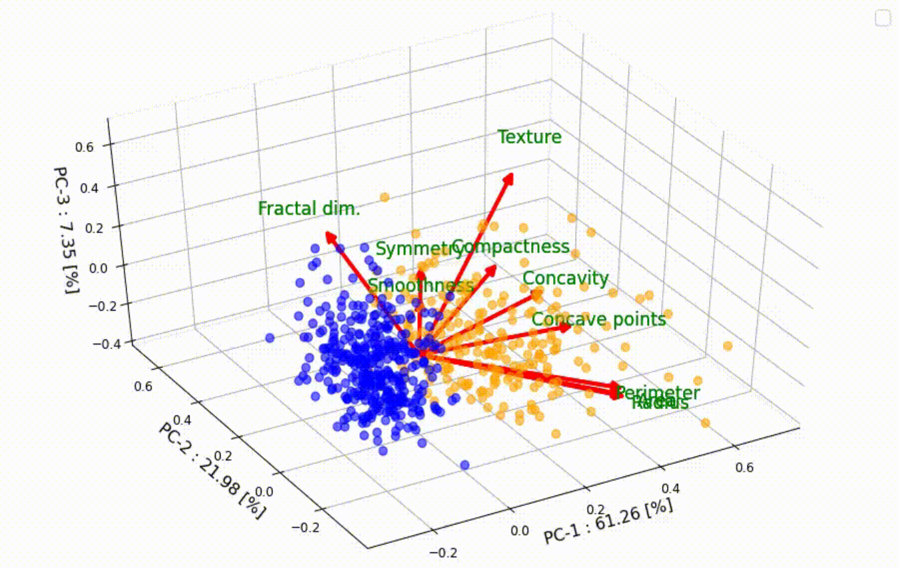
2020-09-22

What is PCA?

- **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 explain the variance-covariance structure of a set of variables through linear combinations
- It is used as dimensionality reduction technique.

Steps

1. Standardization
2. Covariance Matrix Computation
3. Compute Eigenvectors and Eigen values
4. Feature Vector
5. Recast the data along principal component axe



Standardization

- To standardize the range of the continuous initial variables so that each one of them contributes equally to the analysis.
- For example, a variable that ranges between 0 and 100 will dominate over a variable that ranges between 0 and 1), which will lead to biased results.

$$z = \frac{value - mean}{standard\ deviation}$$

Compute Covariance Matrix

- The covariance matrix is a $p \times p$ symmetric matrix (where p is the number of dimensions)

$$\text{Covariance matrix } (\Sigma) = \begin{pmatrix} \text{Cov}(x,x) & \text{Cov}(x,y) \\ \text{Cov}(y,x) & \text{Cov}(y,y) \end{pmatrix}$$

$$= \begin{pmatrix} \text{Var}(x) & \text{Cov}(x,y) \\ \text{Cov}(x,y) & \text{Var}(y) \end{pmatrix}$$

Eigenvectors and Eigenvalues

- Eigen values can be calculated using characteristic polynomial.

Let $\Sigma = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$

Using Characteristic polynomial:

$$\begin{vmatrix} x-a & -b \\ -c & x-d \end{vmatrix}$$ (Eigenvalues)

For Eigen vectors:

$\Sigma \cdot v = \lambda \cdot v$

V = Eigenvector

λ = Eigenvalue

Feature vector

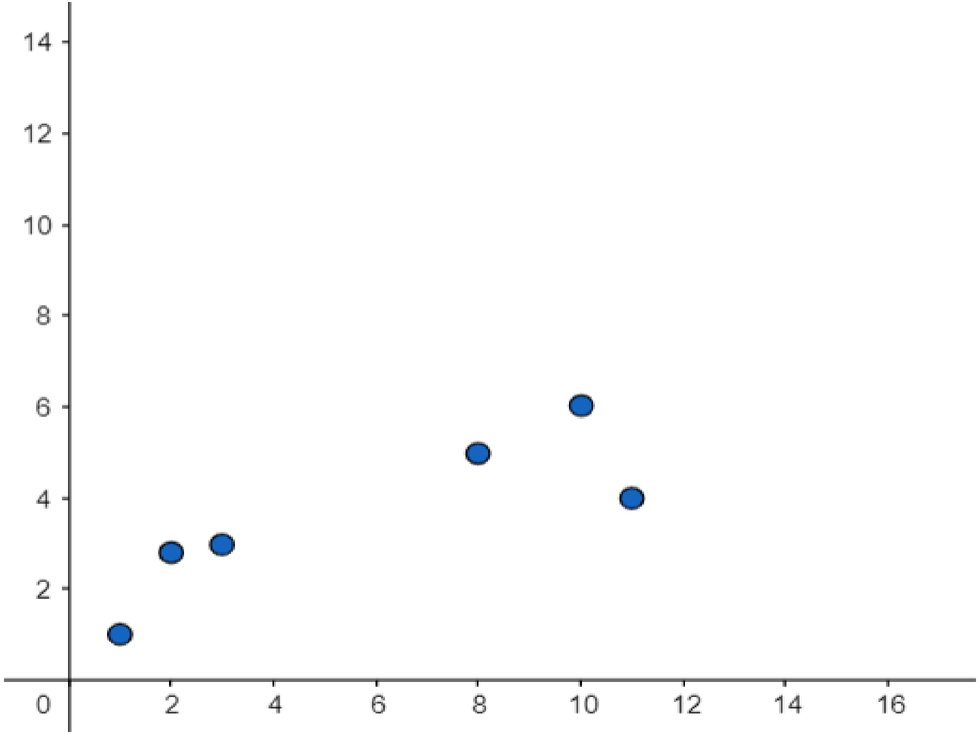
- The feature vector is simply a matrix that has as columns the eigenvectors of the components.

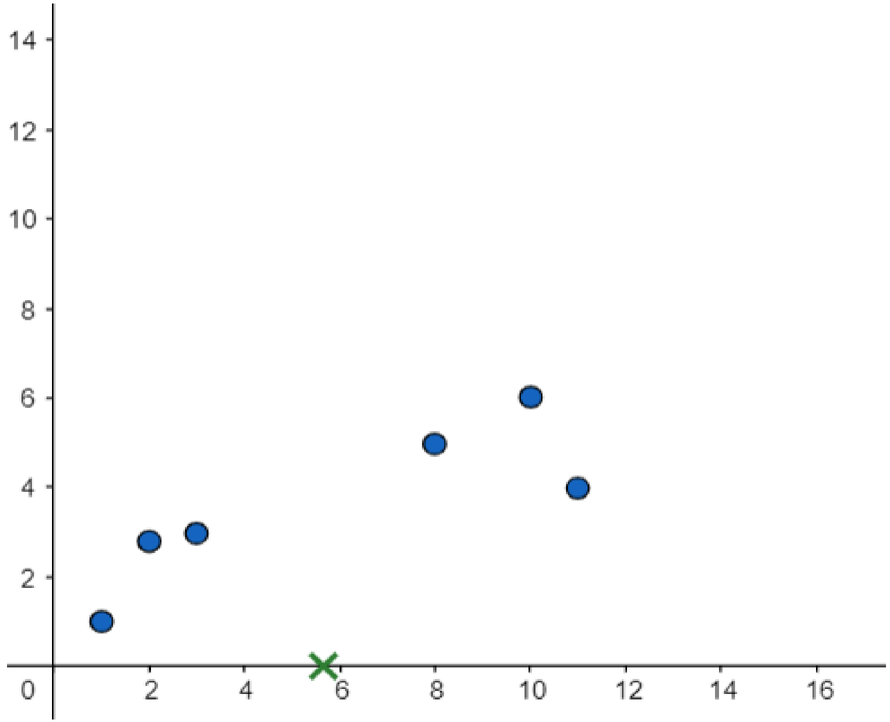
$$\begin{pmatrix} x1 & x2 \\ y1 & y2 \end{pmatrix}$$

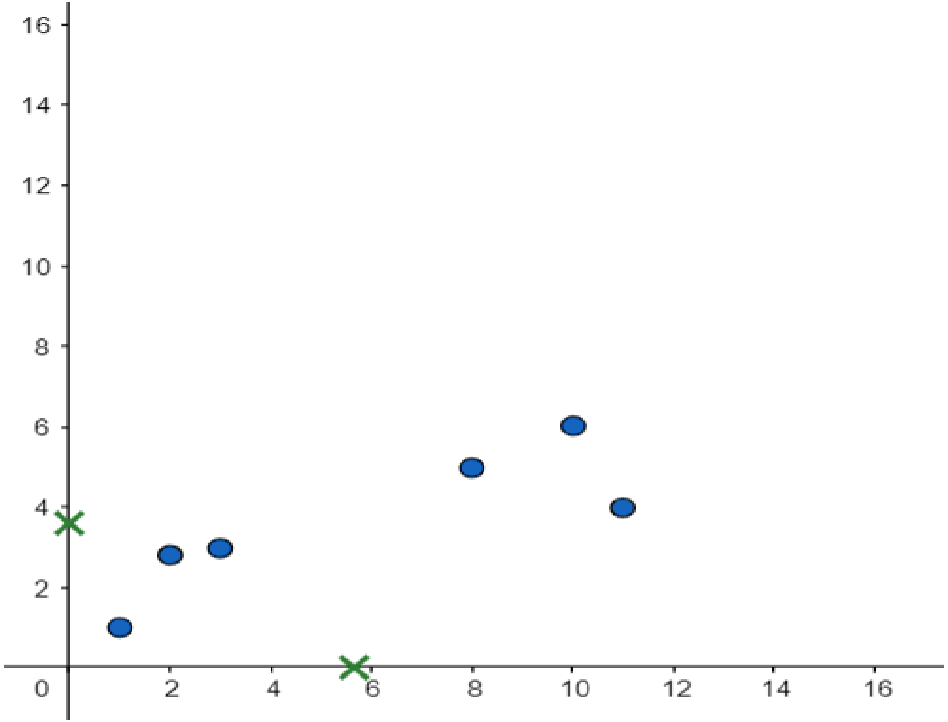
- To choose whether to keep all these components or discard those of lesser significance(of low eigen values).

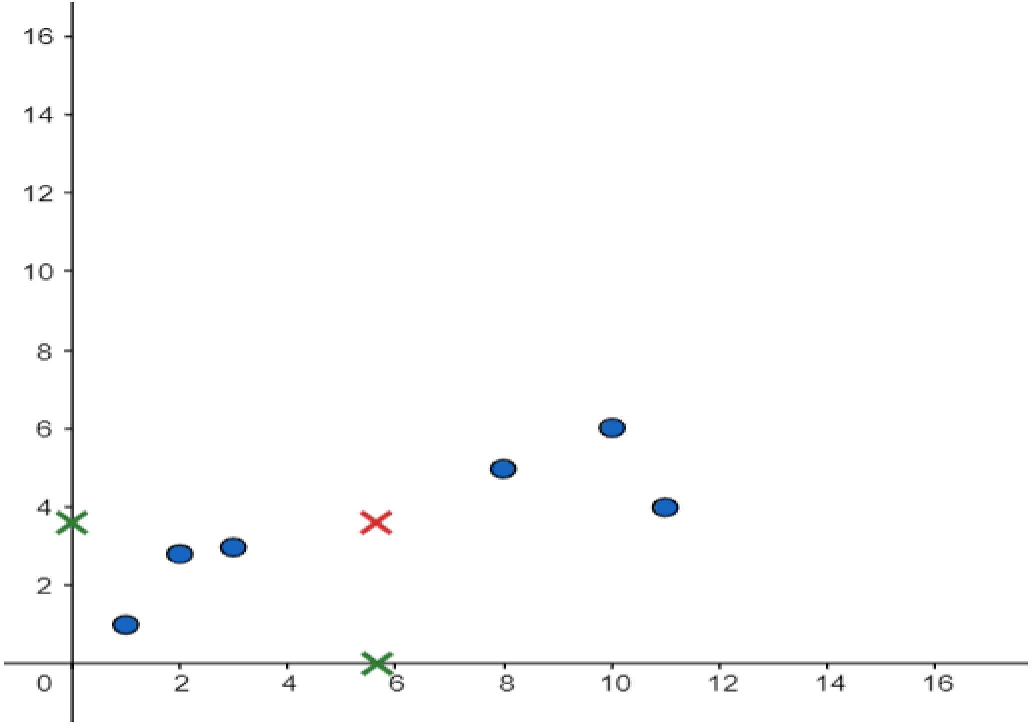
Recast the data

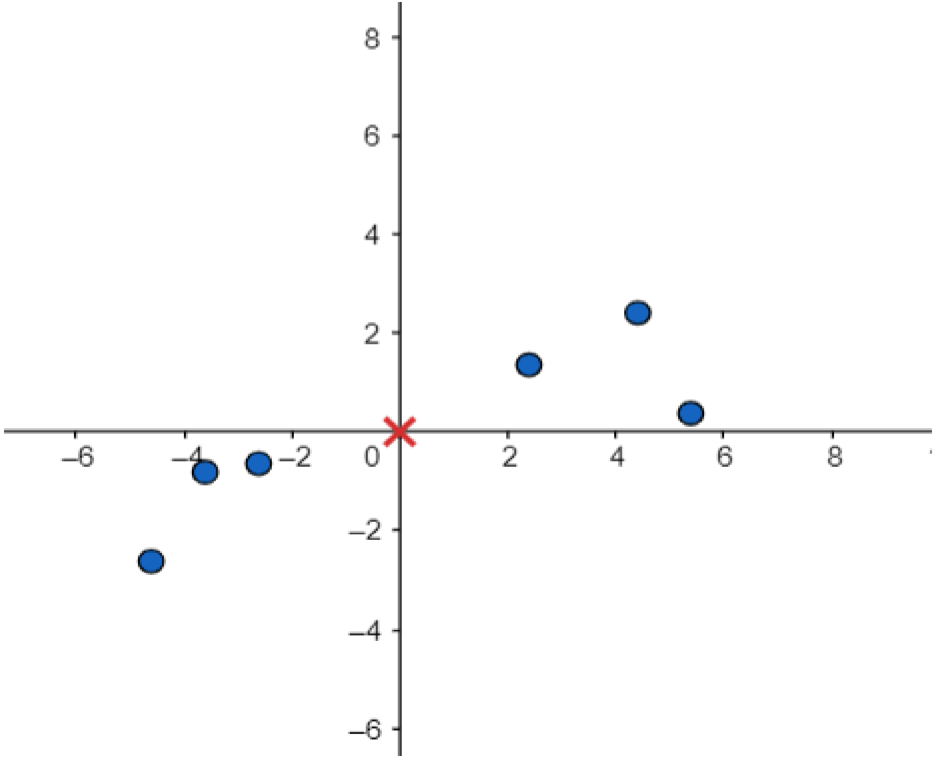
- The aim is to use the feature vector formed using the eigenvectors of the covariance matrix
- To reorient the data from the original axes to the ones represented by the principal components

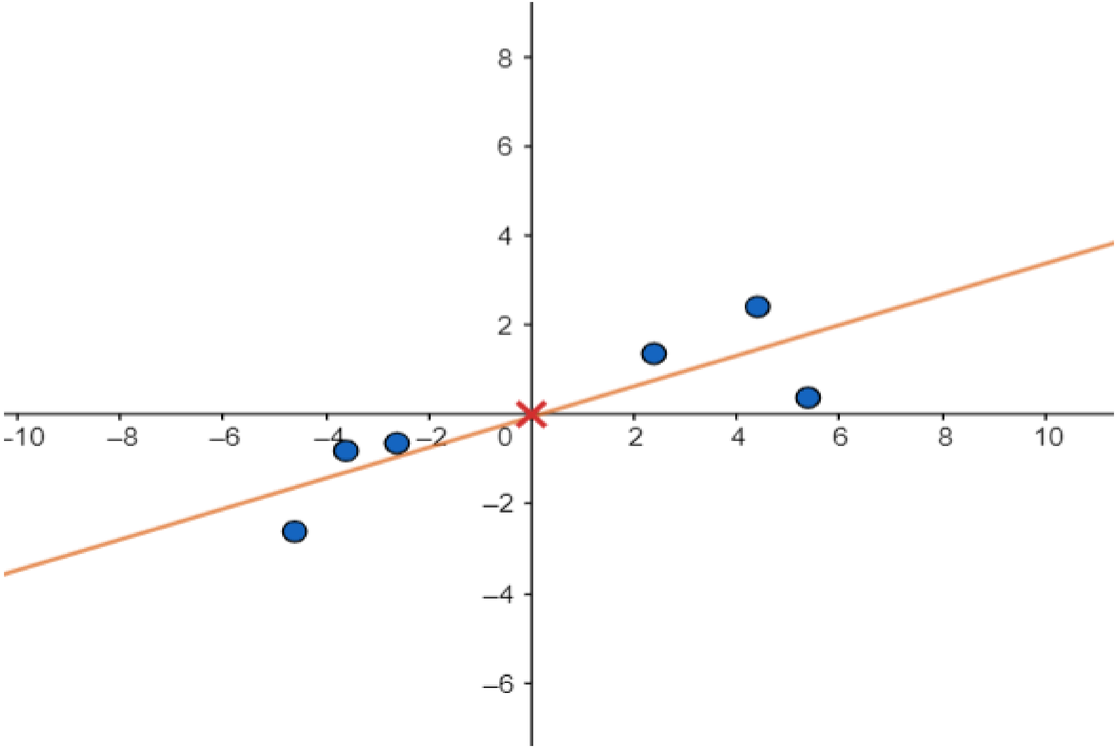

14
12
10
8
6
4
2
0
2
4
6
8
10
12
14
16


14
12
10
8
6
4
2
0
2
4
6
8
10
12
14
16

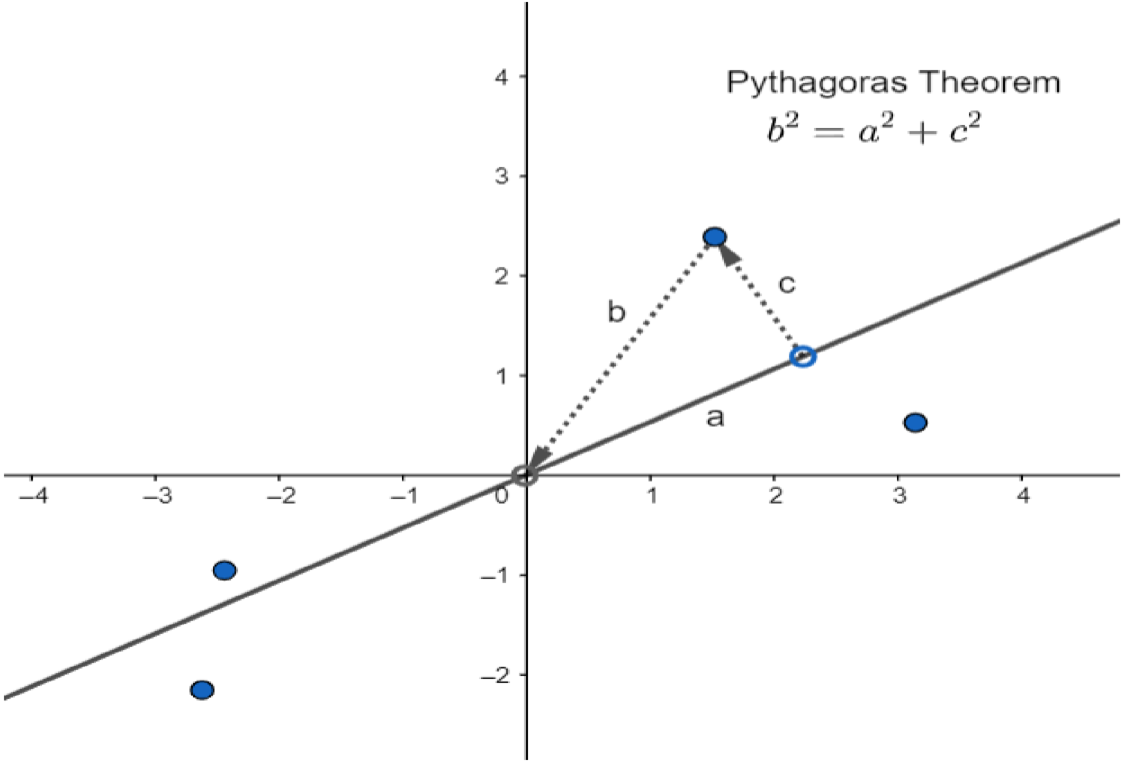

16
14
12
10
8
6
4
2
0
2
4
6
8
10
12
14
16


16
14
12
10
8
6
4
2
0
2
4
6
8
10
12
14
16

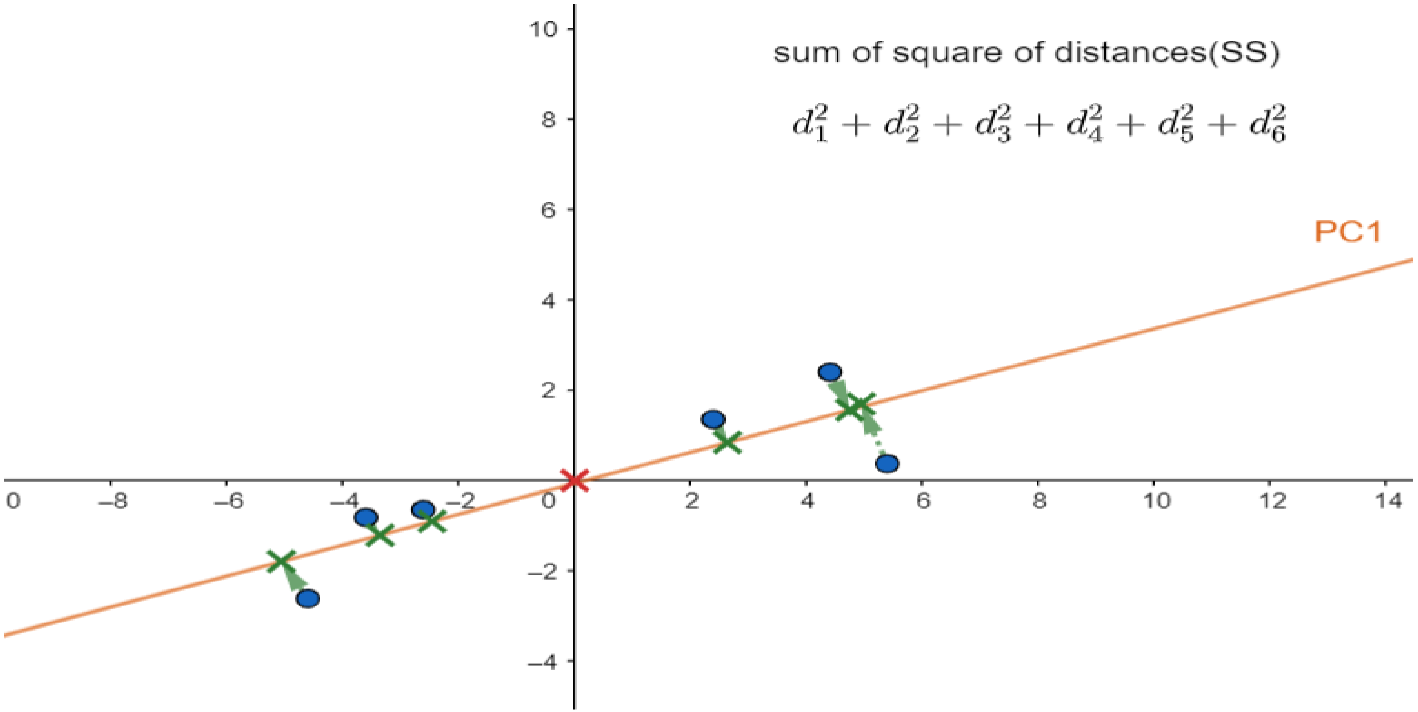

8
6
4
2
0
−2
−4
−6
−6
−4
−2
2
4
6
8

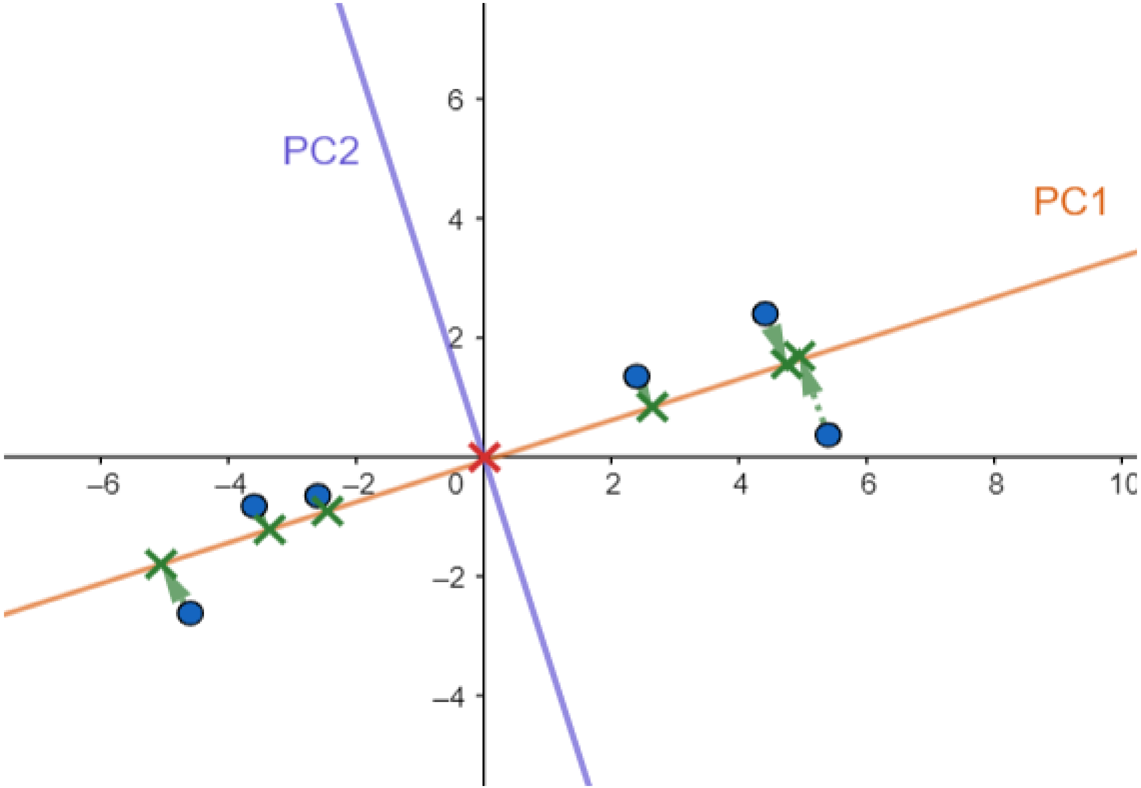


8
6
4
2
0
−2
−4
−6
-10
−8
−6
−4
−2
2
4
6
8
10

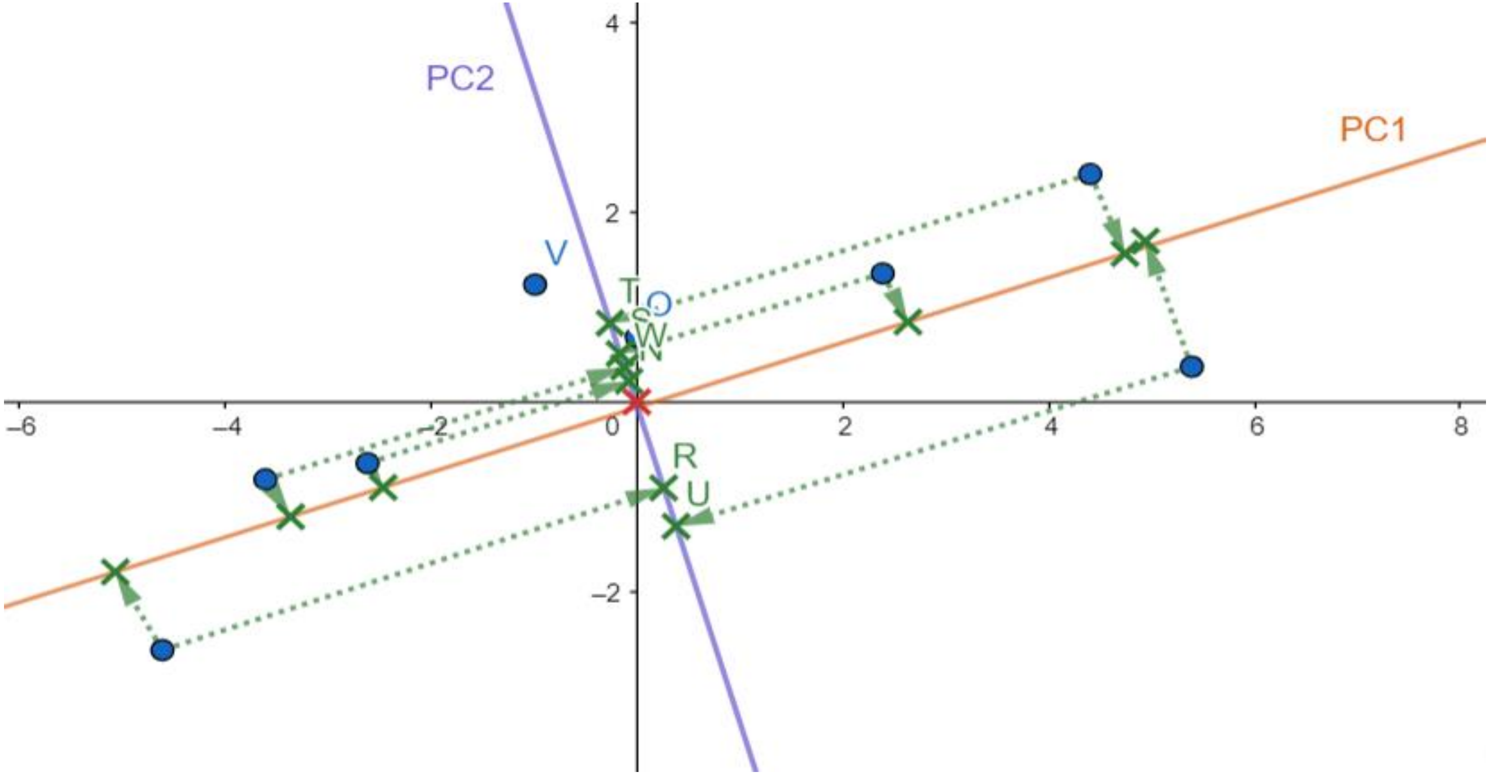


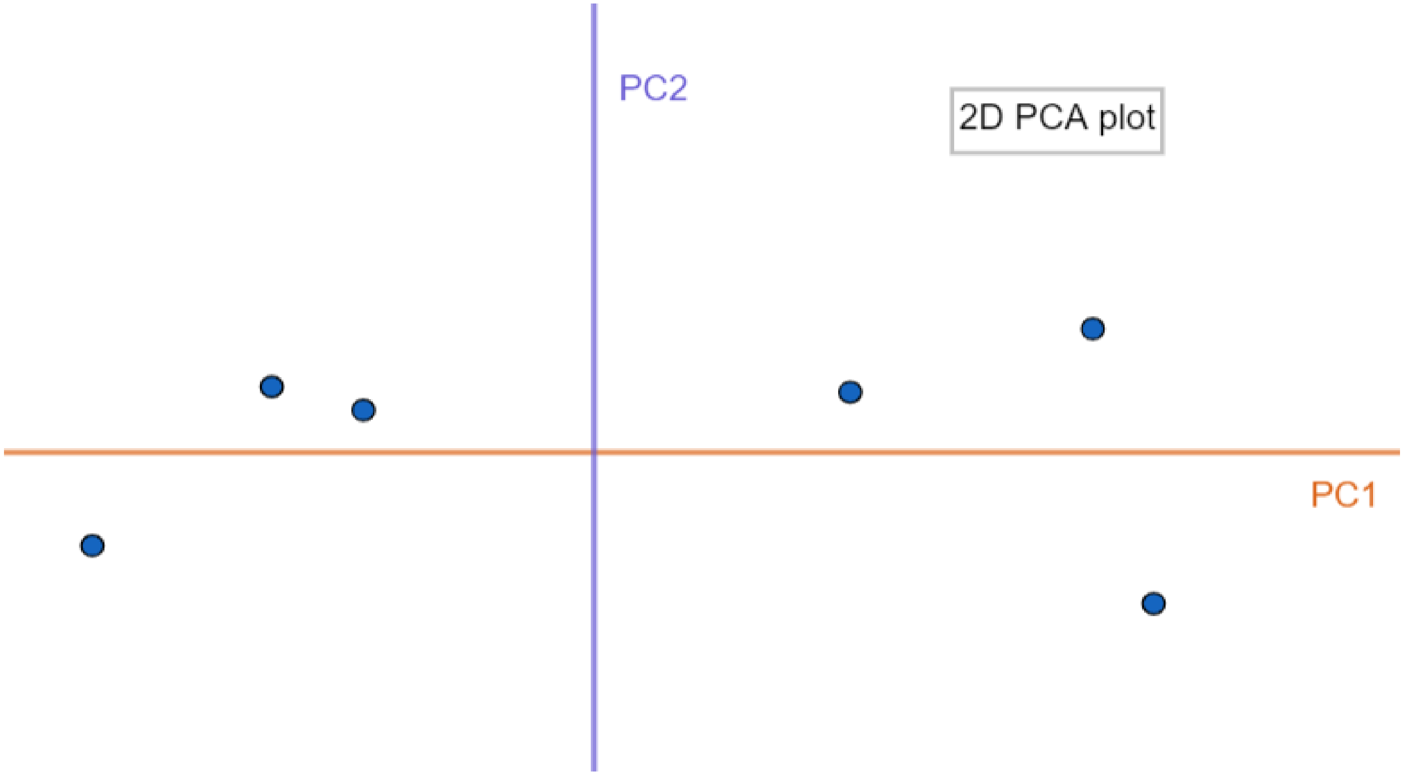
Pythagoras Theorem
$b^2 = a^2 + c^2$
a
b
c

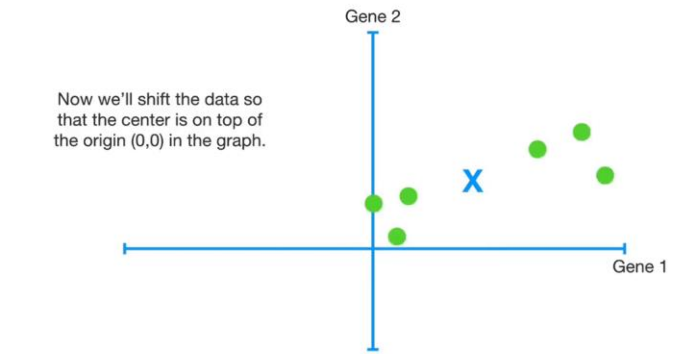

sum of square of distances(SS)
$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$
PC1
10
8
6
4
2
0
−2
−4
−8
−6
−4
−2
2
4
6
8
10
12
14

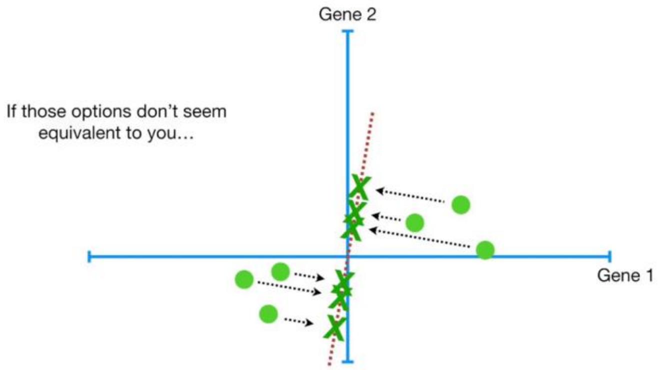


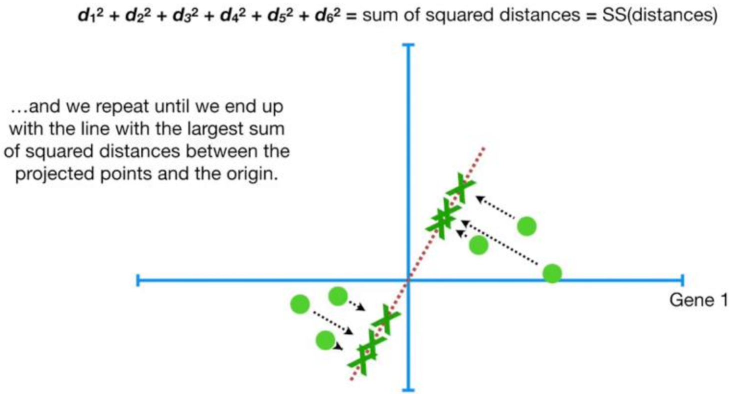
PC2
PC1
6
4
2
0
−2
−4
−6
−4
−2
2
4
6
8
10


PC2
PC1
V
T
O
S
W
N
R
U
4
2
−2
−6
−4
−2
0
2
4
6
8


PC2
2D PCA plot
PC1


Gene 2
Now we'll shift the data so that the center is on top of the origin (0,0) in the graph.
X
Gene 1


Gene 2
If those options don't seem equivalent to you…
Gene 1


$d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2$ = sum of squared distances = SS(distances)
…and we repeat until we end up with the line with the largest sum of squared distances between the projected points and the origin.
Gene 1

Application

- Facial Recognition
- Computer vision
- Image compression
- Finding pattern in data of high dimension
- Data mining

1	2	3
4	5	6
7	8	9

10	11	12
13	14	15
16	17	18

→

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18

PCA

5.5	6.5	7.5
8.5	9.5	10.5
11.5	12.5	13.5

2D PCA

1	2	3
4	5	6
7	8	9

10	11	12
13	14	15
16	17	18

1	2	3
4	5	6
7	8	9

10	11	12
13	14	15
16	17	18

1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18
1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18

PCA

5.5	6.5	7.5
8.5	9.5	10.5
11.5	12.5	13.5

2D PCA

THANK YOU!