EEL5840 Fundamental Machine Learning Homework 2

Hudanyun Sheng

Question 1 Solution:

1.1

$$\Sigma = \frac{1}{4}\begin{bmatrix} 5 & \sqrt{3} \\ \sqrt{3} & 7 \end{bmatrix} = \begin{bmatrix} 5/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 7/4 \end{bmatrix}$$

$$|\Sigma - \lambda I| = \begin{vmatrix} 5/4-\lambda & \sqrt{3}/4 \\ \sqrt{3}/4 & 7/4-\lambda \end{vmatrix} = 0$$

We got $(5/4-\lambda)(7/4-\lambda)(\sqrt{3}/4)^2 = 0$, which can be simplified into $\lambda^2 - 32\lambda + 2 = 0$.

By solving the equation above, we got the eigenvalues are $\lambda_1 = 1$ and $\lambda_2 = 2$.

1.2 $\Sigma \mathbf{v_1} = \frac{1}{4}\begin{bmatrix} 5 & \sqrt{3} \\ \sqrt{3} & 7 \end{bmatrix}\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix} = \begin{bmatrix} 2 \\ 2\sqrt{3} \end{bmatrix} = 2\begin{bmatrix} 1 \\ \sqrt{3} \end{bmatrix} = \lambda_2 \mathbf{v_1}$, thus $\mathbf{v_1}$ is the eigenvector of Σ with the corresponding eigenvalue 2.

$$\Sigma \mathbf{v_2} = \frac{1}{4}\begin{bmatrix} 5 & \sqrt{3} \\ \sqrt{3} & 7 \end{bmatrix}\begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix} = \begin{bmatrix} \sqrt{3} \\ -1 \end{bmatrix} = \lambda_1 \mathbf{v_2}$$, thus, $\mathbf{v_2}$ is the eigenvector of Σ with the corresponding eigenvalue 1.

For convenience of the later discussion, $\lambda_1 = 2$, $\lambda_2 = 1$.

1.3 Based on the spectral theorem, the symmetric matrix Σ can be diagonalized into diagonal matrix D, with the orthogonal matrix $\mathbf{U}$ consisting of the normalized eigenvectors of Σ. Thus, $\mathbf{U} = \begin{bmatrix} \mathbf{v_1}/||\mathbf{v_1}|| \\ \mathbf{v_2}/||\mathbf{v_2}|| \end{bmatrix} = \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$.

D is the diagonal matrix with eigenvalues of Σ, thus $D = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$.

$$\mathbf{UDU}^T = \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix} = \begin{bmatrix} 5/4 & \sqrt{3}/4 \\ \sqrt{3}/4 & 7/4 \end{bmatrix} = \Sigma$$

1.4 Based on the known information, the covariance matrix $\Sigma = \mathbf{x}^T\mathbf{x}$, I got to know $\mathbf{x}$ is a row vector.

And we also define $\mathbf{y}$ to be a row vector. So $\mathbf{y} = \mathbf{x}A^T$, where $A = \begin{bmatrix} \mathbf{v_1}^T \\ \mathbf{v_2}^T \end{bmatrix} = \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$, and $A^T = A = \begin{bmatrix} 1/2 & \sqrt{3}/2 \\ \sqrt{3}/2 & -1/2 \end{bmatrix}$.

1.5 $R_y = E[y^T y] = AR_X A^T = \begin{bmatrix} \mathbf{v_1}^T \\ \mathbf{v_2}^T \end{bmatrix} R_x \begin{bmatrix} \mathbf{v_1} & \mathbf{v_2} \end{bmatrix}$

$\because$ $\mathbf{v_1}$ and $\mathbf{v_2}$ are the eigenvectors of R_x, $\therefore$ $R_x\mathbf{v_1} = \lambda_1\mathbf{v_1}$, $R_x\mathbf{v_2} = \lambda_2\mathbf{v_2}$.

$\therefore R_y = \begin{bmatrix} \mathbf{v_1}^T R_x \mathbf{v_1} & \mathbf{v_1}^T R_x \mathbf{v_2} \\ \mathbf{v_2}^T R_x \mathbf{v_1} & \mathbf{v_2}^T R_x \mathbf{v_2} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$

1.6 Based on the definition of Mahalanobis distance found on wikipedia (https://en.wikipedia.org/wiki/Mahalanobis_distance):

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{\mu})^T S^{-1} (\vec{x} - \vec{\mu})}$$

In our case, $\vec{\mu} = \vec{0}$, $S = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}$, $S^{-1} = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix}$.

Thus, $D_M(\vec{x}) = \sqrt{\begin{bmatrix} x_1' & x_2' \end{bmatrix} \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \end{bmatrix}} = 1$. I got $\frac{1}{2}x_1'^2 + x_2'^2 = 1$, which is $\frac{(x_1' - 0)^2}{\sqrt{2}^2} + \frac{(x_2' - 0)^2}{1} = 1$. It is actually an ellipse, with center at the origin, and semi-major axis on the x_1' axis, with length equal to $\sqrt{2}$, and semi-minor axis on the x_2' axis, with length equal to 1.

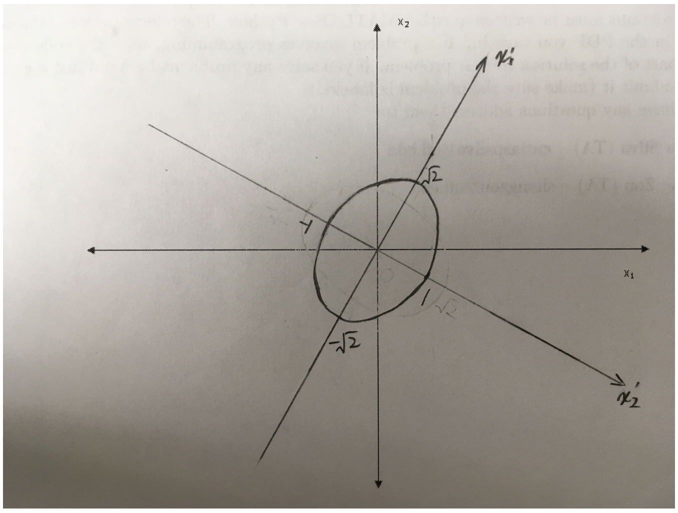


Figure 1: New coordinate system and the curve describing all points with a Mahalanobis distance of 1 from the origin

Question 2

Solution:

If matrix A is eigendecomposible, which means matrix A is a diagonalizable $n \times n$ square matrix with n eigenvalues λ_1, λ_2, ... λ_n and corresponding normalized eigenvectors $\mathbf{e_1}$, $\mathbf{e_2}$, ... $\mathbf{e_n}$. We have

$$\Lambda = PAP^T$$

, where $P^T = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & ... & \mathbf{e_n} \end{bmatrix}$, Λ is a diagonal matrix.

$P^T P = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & ... & \mathbf{e_n} \end{bmatrix} \begin{bmatrix} \mathbf{e_1}^T \\ \mathbf{e_2}^T \\ ... \\ \mathbf{e_n}^T \end{bmatrix} = 1$, and $A = P^{-1}\Lambda(P^T)^{-1}$

Let $Q = P^{-1}$, then $A = Q\Lambda Q^T$.

$$A^k = (Q\Lambda Q^T)^k = Q\Lambda Q^T \times Q\Lambda Q^T \times ... \times Q\Lambda Q^T = Q\Lambda^k Q^T$$

, where $Q = P^{-1}$, P is defined by $P^T = \begin{bmatrix} \mathbf{e_1} & \mathbf{e_2} & ... & \mathbf{e_n} \end{bmatrix}$, $\mathbf{e_i}(i = 1, 2, ..., n)$ are n eigenvectors of matrix A.

Question 3

Solution:

The code I used to import the 3 files of dataset into MATLAB shows as below:

```
addpath('/Users/hudanyun.sheng/Google Drive/Me/201708Fall/EEL5840FundamentalMachineLe
load ('ellipsoids.txt');
load ('spheres.txt');
load ('swissroll.txt');
```

To consider **X** to be any one of the dataset, simply use

```
X = swissroll;
```

or

```
X = spheres;
```

or

```
X = ellipsoids;
```

3.1 The MATLAB code used to calculate the covariance of every data set is shown below:

```
mu = mean(X);
X_std = X - mu;
cov_mat = cov(X_std);
```

"cov_mat" is the desired covariance matrix.

- For the **"Swiss Roll"** data set, the covariance matrix is

$$\begin{bmatrix} 43.2882 & 0.1535 & 4.4555 \\ 0.1535 & 10.5800 & 0.1544 \\ 4.4555 & 0.1544 & 47.1548 \end{bmatrix}$$

- For the **"Spheres"** data set, the covariance matrix is

$$\begin{bmatrix} 10.0368 & 8.9743 & 9.0941 \\ 8.9743 & 9.9246 & 9.0678 \\ 9.0941 & 9.0678 & 10.2626 \end{bmatrix}$$

- For the **"Ellipsoids"** data set, the covariance matrix is

$$\begin{bmatrix} 57.8661 & 14.6934 & 7.3664 \\ 14.6934 & 9.8630 & 4.5264 \\ 7.3664 & 4.5264 & 3.3570 \end{bmatrix}$$

3.2 The MATLAB code used to find the eigenvectors and eigenvalues of the covariance matrix is shown below:

```
[eigenVecs, eigenVals] = eig(cov_mat);
```

"eigenVals" is the diagonal matrix with eigenvalues of covariance matrix at the diagonal, "eigenVecs" a matrix whose columns are the corresponding eigenvectors.

- For the **"Swiss Roll"** data set, the eigenvalues of the covariance matrix are $\lambda_1 = 10.5788$, $\lambda_2 = 40.3647$, $\lambda_3 = 50.0795$, and the corresponding eigenvectors of the covariance matrix are

$$\mathbf{e_1} = \begin{bmatrix} 0.0042 \\ -1.0000 \\ 0.0037 \end{bmatrix}, \mathbf{e_2} = \begin{bmatrix} 0.8361 \\ 0.0015 \\ -0.5486 \end{bmatrix}, \text{ and } \mathbf{e_3} = \begin{bmatrix} 0.5486 \\ 0.0054 \\ 0.8361 \end{bmatrix}.$$

- For the **"Spheres"** data set, the eigenvalues of the covariance matrix are $\lambda_1 = 1.0001$, $\lambda_2 = 1.0571$, $\lambda_3 = 28.1668$, and the corresponding eigenvectors of the covariance matrix are

$$\mathbf{e_1} = \begin{bmatrix} 0.5318 \\ -0.8043 \\ 0.2652 \end{bmatrix}, \mathbf{e_2} = \begin{bmatrix} 0.6207 \\ 0.1572 \\ -0.7681 \end{bmatrix}, \text{ and } \mathbf{e_3} = \begin{bmatrix} 0.5760 \\ 0.5731 \\ 0.5829 \end{bmatrix}.$$

- For the **"Ellipsoids"** data set, the eigenvalues of the covariance matrix are $\lambda_1 = 1.0352$, $\lambda_2 = 6.8856$, $\lambda_3 = 63.1653$, and the corresponding eigenvectors of the covariance matrix are

$$\mathbf{e_1} = \begin{bmatrix} 0.0046 \\ -0.4623 \\ 0.8867 \end{bmatrix}, \mathbf{e_2} = \begin{bmatrix} -0.3068 \\ 0.8433 \\ 0.4413 \end{bmatrix}, \text{ and } \mathbf{e_3} = \begin{bmatrix} -0.9518 \\ -0.2741 \\ -0.1380 \end{bmatrix}.$$

3.3 The MATLAB code used to find and plot the projection of the data points into the 2-D and 1-D principal components is shown below:

```
w2 = eigenVecs(:,2:3);
w1 = eigenVecs(:,3);
X_pca2 = X_std * w2;
X_pca1 = X_std * w1;
figure
scatter(X_pca2(:,1), X_pca2(:,2))
xlabel('pc1', 'FontSize', 12, 'FontWeight', 'bold');
ylabel('pc2', 'FontSize', 12, 'FontWeight', 'bold')
```

```
title('Swiss Roll into 2-D', 'FontSize', 20, 'FontWeight', 'bold');
% title('Spheres into 2-D', 'FontSize', 20, 'FontWeight', 'bold');
% title('Ellipsoids into 2-D', 'FontSize', 20, 'FontWeight', 'bold');
figure
scatter(X_pca1, zeros(size(X_pca1)))
xlabel('pc', 'FontSize', 12, 'FontWeight', 'bold');
title('Swiss Roll into 1-D', 'FontSize', 20, 'FontWeight', 'bold');
% title('Spheres into 2-D', 'FontSize', 20, 'FontWeight', 'bold');
% title('Ellipsoids into 2-D', 'FontSize', 20, 'FontWeight', 'bold');
```

- The **“Swiss Roll”** dataset:

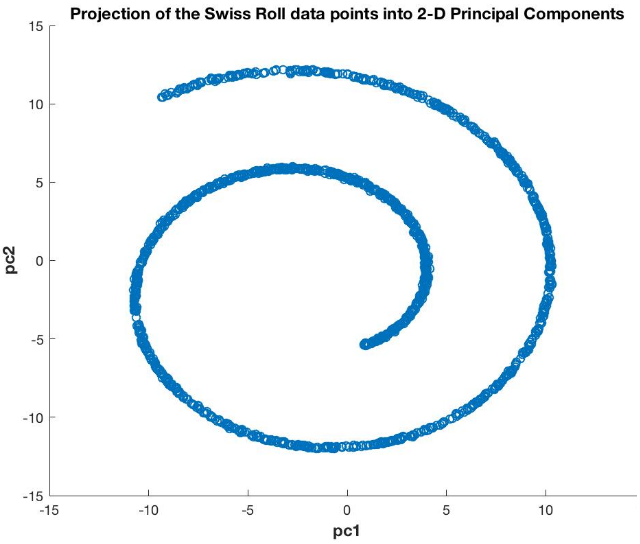


Figure 2: projection of “Swiss Roll” into the 2-D

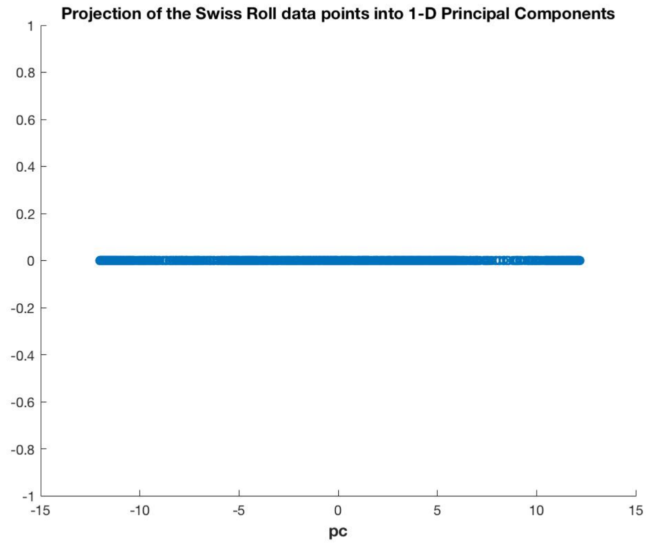


Figure 3: projection of “Swiss Roll” into the 1-D

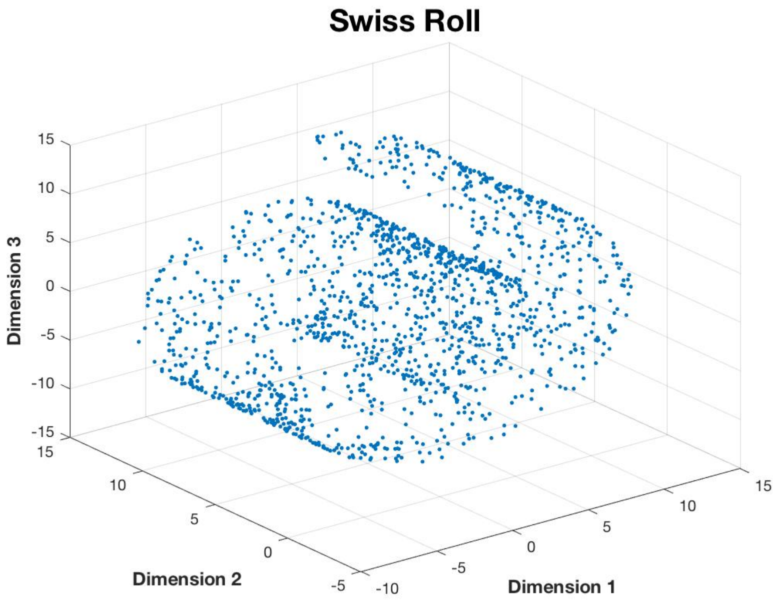


Figure 4: original plot of the “Swiss Roll” dataset

Based on the results, I would say that the 2-D projection preserve the most informative structure of the original data, while the 1-D projection has not the important information enough to represent the original dataset. For this “Swiss Roll” dataset,

- The **"Spheres"** dataset:

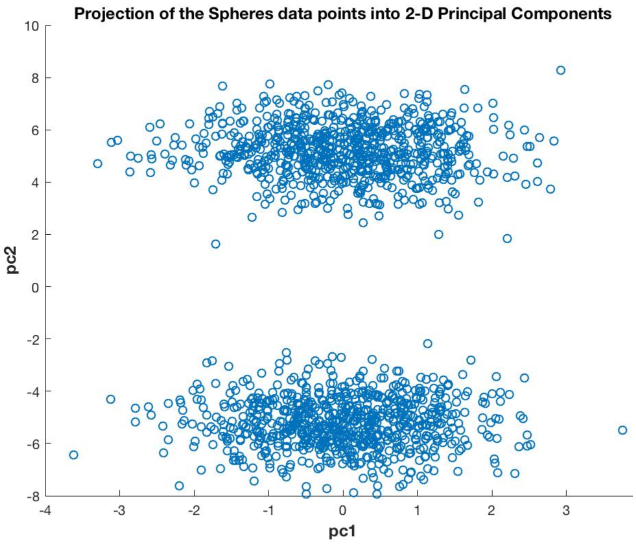


Figure 5: projection of "Spheres" into the 2-D

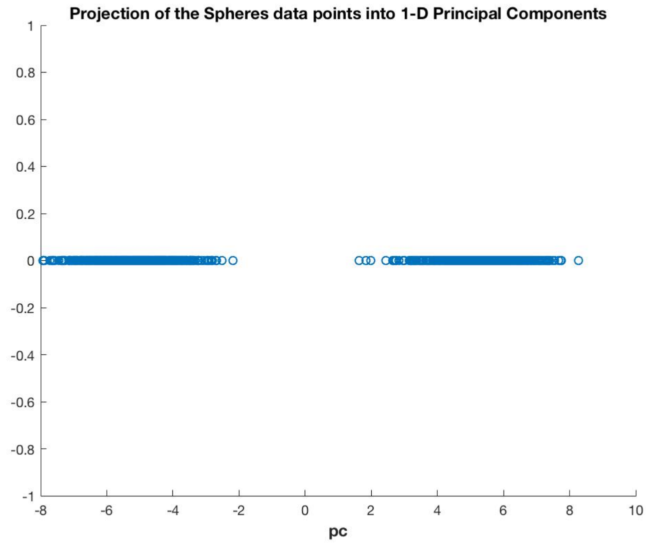


Figure 6: projection of "Spheres" into the 1-D

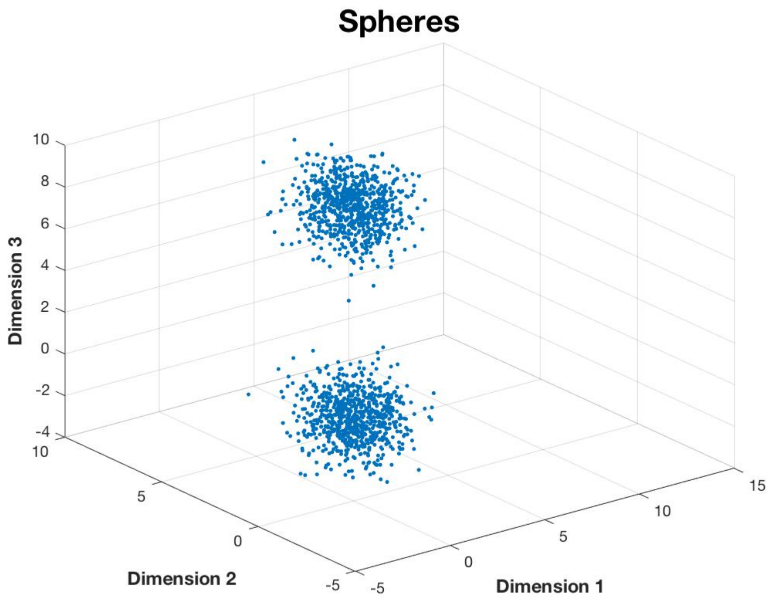
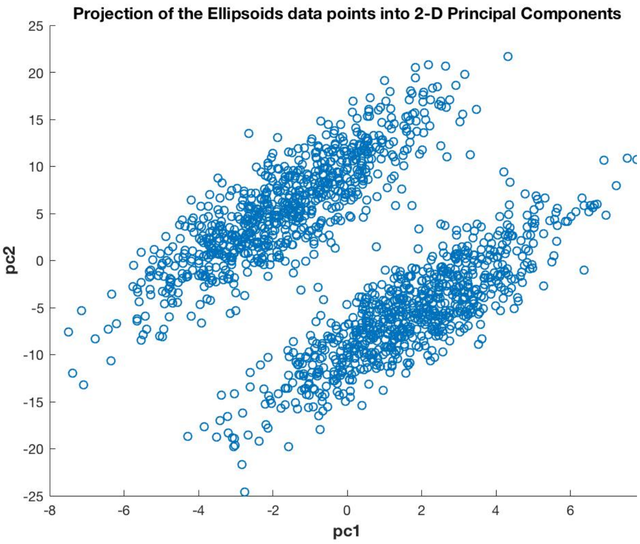


Figure 7: original plot of the "Spheres" dataset

Based on the results, I would say that the 2-D projection preserve the most informative structure of the original data, while the 1-D projection has not the important information enough to represent the original dataset. For this "Spheres" dataset,

- The **"Ellipsoids"** dataset:



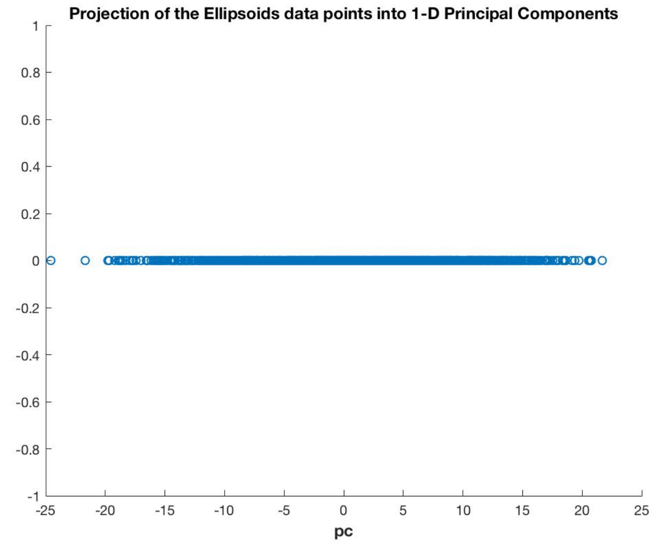


Figure 8: projection of "Ellipsoids" into the 2-D

Figure 9: projection of "Ellipsoids" into the 1-D

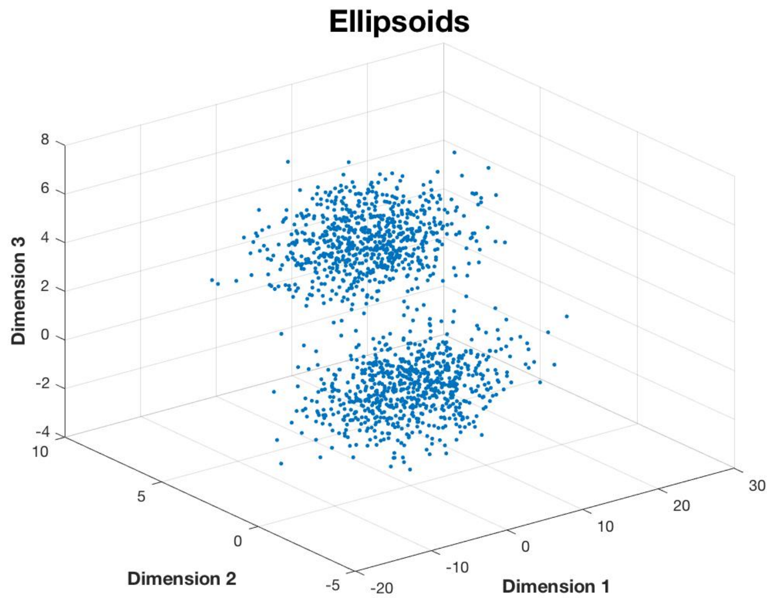


Figure 10: original plot of the "Swiss Roll" dataset

Based on the results, I would say that the 2-D projection preserve the most informative structure of the original data, while the 1-D projection has not the important information enough to represent the original dataset. For this "Ellipsoids" dataset,