Assignment 1 Solutions, Foundations: 50 points

Due September 6, 2024, 11:59 PM CST

Instructions

Your homework submission must cite any references used (including articles, books, code, websites, and personal communications). All solutions must be written in your own words, and you must program the algorithms yourself. Submit your solutions as a PDF to Canvas: `https://canvas.tamu.edu/`.

Your programs must be written in Python. The relevant code to the problem should be in the PDF you turn in or submitted as a separate file (*e.g.*, Python file (.py), Jupter/Google Colab Notebook (.ipynb)). If a problem involves programming, then the code should be shown as part of the solution to that problem. If you solve any problems by hand just digitize that page and submit it (make sure the problem is clearly labeled and legible).

If you have any questions, please reach out to Dr. Peeples. Python code solutions can be found here (need to be logged into your TAMU Google account).

1 Linear Algebra- 10 points

For each of the following problems, state whether or not the the operation is defined (i.e., valid and can be computed) and, if it is defined, what is the size of the resulting answer. For all of the following problems let $\mathbf{X}$ be a $M \times N$ matrix, $\mathbf{Y}$ be a $N \times N$ matrix, $\mathbf{a}$ be a M × 1 vector, $\mathbf{b}$ be a $N \times 1$ vector and s be a scalar.

1. $\mathbf{XY}$ Defined. The result is a matrix of size $M \times N$.

2. $\mathbf{YX}$ Not defined.

3. $\mathbf{YX^T}$ Defined. The result is a matrix of size $N \times M$.

4. $\mathbf{aX}$ Not defined.

5. $\mathbf{a^TX}$ Defined. The result is a vector of size $1 \times N$.

6. $\mathbf{aX^T}$ Not defined.

7. $\mathbf{a^Tb}$ Not defined.

8. $\mathbf{b^T b}$ Defined. The result is a scalar. (This is also called the inner product).

9. $\mathbf{b b^T}$ Defined. The result is a matrix of size $N \times N$. (This is also called the outer product).

10. $s\mathbf{X} + \mathbf{Y}$ Not defined.

2 Probability- 5 points

The following frequency table is generated from an experiment of rolling a fair dice. You can do this problem by hand or create a python script.

	1	2	3	4	5	6
Counts	200	100	300	50	150	200

1. Compute and plot the empirical probability mass function.
   To compute P(x), need to divide the counts divided by the total number of rolls/trials (1000):

x	1	2	3	4	5	6
P(x)	0.20	0.10	0.30	0.05	0.15	0.20

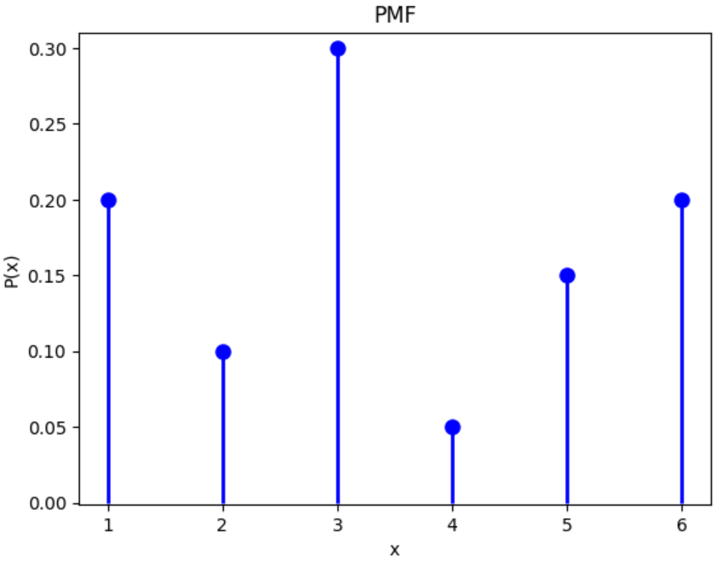


Figure 1: PMF for experimental results of rolling dice for 1000 trials.

2. Compute and plot the empirical cumulative distribution.
   To compute P(X $\leq$ x), need compute cumulative sum (*i.e.*, integrate) of PMF:

x	1	2	3	4	5	6
P(X $\leq$ x)	0.20	0.30	0.60	0.65	0.80	1.0

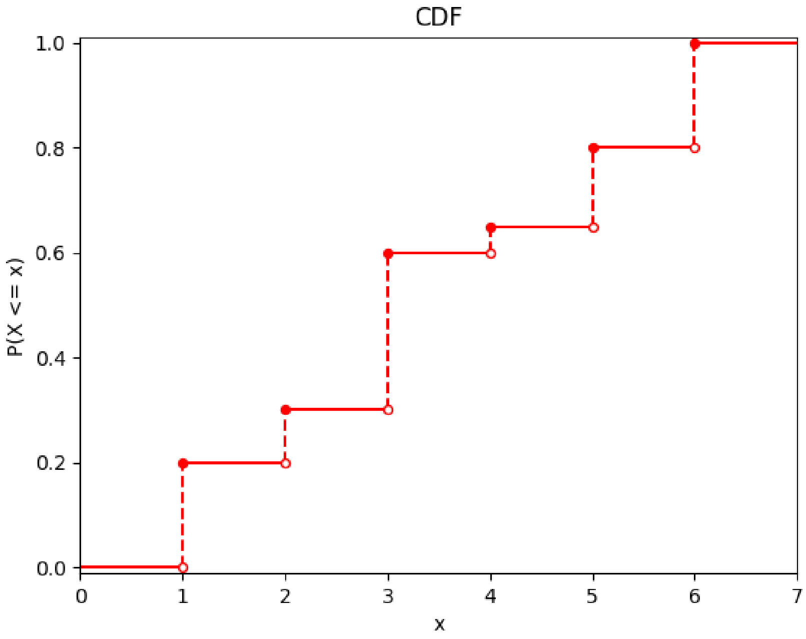


Figure 2: CDF for experimental results of rolling dice for 1000 trials.

3 Statistics- 10 points

Given two datasets, **A** : $[4, 16, 4, 5, 1]^T$ and **B** : $[8, 3, 4, 8, 7]^T$, compute the following statistical (sample) measures by hand for each dataset:

1. Mean

   (a) **A**:

$$\hat{\mu_{\mathbf{A}}} = \frac{1}{n}\sum_{n=1}^{n} x_i = \frac{1}{5}\sum_{n=1}^{5} x_i = \frac{1}{5}(4+16+4+5+1) = \frac{1}{5}(30) = 6 \quad (1)$$

(b) **B**:

$$\hat{\mu_{\mathbf{B}}} = \frac{1}{n}\sum_{n=1}^{n} x_i = \frac{1}{5}\sum_{n=1}^{5} x_i = \frac{1}{5}(8+3+4+8+7) = \frac{1}{5}(30) = 6 \quad (2)$$

2. Median
   Order dataset and find middle-most value:

   (a) **A**:

$$\mathbf{A}_{ordered} : [1,4,4,5,16]^T$$
$$\hat{m_{\mathbf{A}}} = \hat{F}_{\mathbf{A}}^{-1}(.5) = 4 \quad (3)$$

   (b) **B**:

$$\mathbf{B}_{ordered} : [3,4,7,8,8]^T$$
$$\hat{m_{\mathbf{B}}} = \hat{F}_{\mathbf{B}}^{-1}(.5) = 7 \quad (4)$$

3. Range

   (a) **A**:

$$\hat{r_{\mathbf{A}}} = \max_{i=1}^{n} x_i - \min_{i=1}^{n} x_i = \max_{i=1}^{5} x_i - \min_{i=1}^{5} x_i = 16 - 1 = 15 \quad (5)$$

   (b) **B**:

$$\hat{r_{\mathbf{B}}} = \max_{i=1}^{n} x_i - \min_{i=1}^{n} x_i = \max_{i=1}^{5} x_i - \min_{i=1}^{5} x_i = 8 - 3 = 5 \quad (6)$$

4. Variance
   Note: The problem asked for the sample variance, so the **biased** estimator should be used:

$$\hat{\sigma_{\mathbf{A}}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu_{\mathbf{A}}}) = \frac{1}{5}\sum_{i=1}^{n}(x_i - 6) \quad (7)$$

$$= \frac{1}{5}((4-6)^2 + (16-6)^2 + (4-6)^2 + (5-6)^2 + (1-6)^2)$$

$$= \frac{1}{5}((-2)^2 + (10)^2 + (-2)^2 + (-1)^2 + (-5)^2) = \frac{1}{5}(134) = \frac{134}{5} = 26.8$$

$$\hat{\sigma_{\mathbf{B}}} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{\mu_{\mathbf{B}}}) = \frac{1}{5}\sum_{i=1}^{n}(x_i - 6) \quad (8)$$

$$= \frac{1}{5}((8-6)^2 + (3-6)^2 + (4-6)^2 + (8-6)^2 + (7-6)^2)$$

$$= \frac{1}{5}((2)^2 + (-3)^2 + (-2)^2 + (2)^2 + (1)^2) = \frac{1}{5}(22) = \frac{22}{5} = 4.4$$

5. For which dataset (**A** or **B**), would a new value of 18 be more likely to be an outlier and why?

Each dataset, **A** or **B**, have the same mean, but different variance. There are several ways to identity an outlier such as using the a) Z-Score, b) interquartile range (IQR), and c) visualization (*e.g.*, box plots, scatter plots, histograms). The Z-Score can be computed using the following equation:

$$Z = \frac{x - \hat{\mu}}{\hat{\sigma}}. \tag{9}$$

Generally, data points are considered outliers if the values are $\pm 3\sigma$ from the mean (*i.e.*, Z score has a magnitude greater than 3). The Z-Score for datasets **A** or **B** respectively are the following:

$$Z_{\mathbf{A}} = \frac{x - \hat{\mu}_{\mathbf{A}}}{\hat{\sigma}_{\mathbf{A}}} = \frac{18 - 6}{\sqrt{26.8}} = \frac{12}{\sqrt{26.8}} \approx 2.32 \tag{10}$$

$$Z_{\mathbf{B}} = \frac{x - \hat{\mu}_{\mathbf{B}}}{\hat{\sigma}_{\mathbf{B}}} = \frac{18 - 6}{\sqrt{4.4}} = \frac{12}{\sqrt{4.4}} \approx 5.72. \tag{11}$$

Given the resulting Z-score dataset **B**, a new value of 18 would be an outlier.

If IQR is used, any value that is 1.5 times IQR below quartile 1 (Q1) or 1.5 times above quartile 3 (Q3) is considered an outlier. The following steps detail how to use IQR for outlier detection:

1. Order datasets:

$$\mathbf{A}_{ordered} : [1, 4, 4, 5, 16]^T$$

$$\mathbf{B}_{ordered} : [3, 4, 7, 8, 8]^T$$

2. Compute Q1, Q2 (median), Q3, and IQR:

   (a) **A**:

   i. Q1 (median of lower half of data): Q1 $= \frac{1+4}{2} = \frac{5}{2} = 2.5$

   ii. Q2 (median of data): Q2 $= 4$

   iii. Q3 (median of upper half of data): Q3 $= \frac{16+5}{2} = \frac{21}{2} = 10.5$

   iv. IQR: IQR $= \hat{F}_{\mathbf{A}}^{-1}(.75) - \hat{F}_{\mathbf{A}}^{-1}(.25) = Q3 - Q1 = 10.5 - 2.5 = 8$

   (b) **B**:

   i. Q1 (median of lower half of data): Q1 $= \frac{3+4}{2} = \frac{7}{2} = 3.5$

   ii. Q2 (median of data): Q2 $= 7$

   iii. Q3 (median of upper half of data): Q3 $= \frac{8+8}{2} = \frac{16}{2} = 8$

        iv. IQR: IQR $= \hat{F}_{\mathbf{A}}^{-1}(.75) - \hat{F}_{\mathbf{A}}^{-1}(.25) = Q3 - Q1 = 8 - 3.5 = 4.5$

3. Compute lower and upper bounds:

    (a) **A**:

        i. Lower bound: Q1 - 1.5IQR $= 2.5 - 1.5 \times 8 = 2.5 - 12 = -9.5$

        ii. Upper bound: Q3 + 1.5IQR $= 10.5 + 1.5 \times 8 = 10.5 + 12 = 22.5$

    (b) **B**:

        i. Lower bound: Q1 - 1.5IQR $= 3.5 - 1.5 \times 4.5 = 2.5 - 6.75 = -4.25$

        ii. Upper bound: Q3 + 1.5IQR $= 3.5 + 1.5 \times 4.5 = 3.5 + 6.75 = 10.25$

    (c) Values outside of bounds are outliers:

        i. **A**: 18 is within the upper and lower limits of **A** (not an outlier)

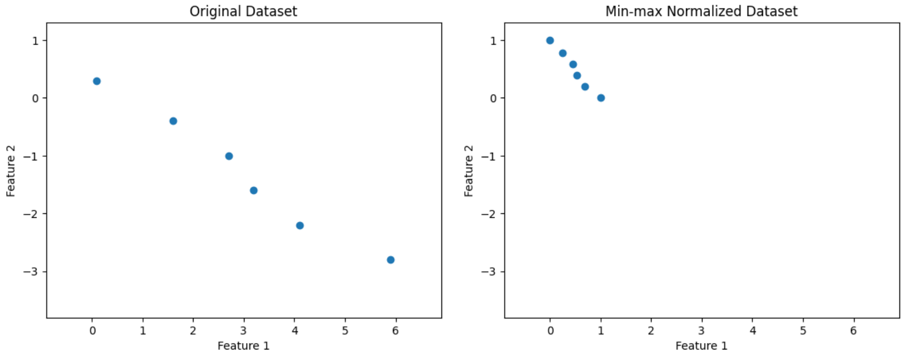
        ii. **B**: 18 is outside of the upper and lower limits of **B** (outlier)

4 Data Attributes- 10 points

4.1 Numerical

Given the following data matrix, $\mathbf{D} = \begin{pmatrix} 0.1 & 1.6 & 2.7 & 3.2 & 4.1 & 5.9 \\ 0.3 & -0.4 & -1 & -1.6 & -2.2 & -2.8 \end{pmatrix}^T$, transform and plot the new data using (can either compute and plot by hand or creating python script):

1. Min-max normalization

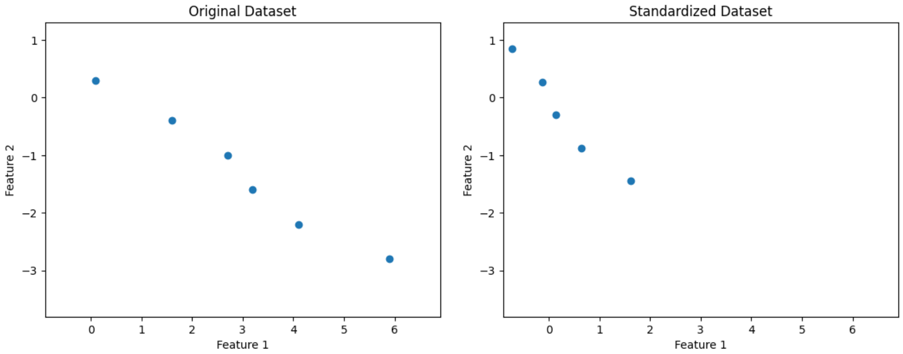


(a) Original Dataset, **D**

(b) Normalized Dataset, $\mathbf{D}_{normalized}$

Figure 3: Original dataset and min-max (*i.e.*, range) normalization are shown in Figures 4a and 4b respectively. Best practice is to make sure your plots have the same axes to compare the original and transformed datasets.

2. Standard score normalization (*i.e.*, standardization)



(a) Original Dataset, $\mathbf{D}$

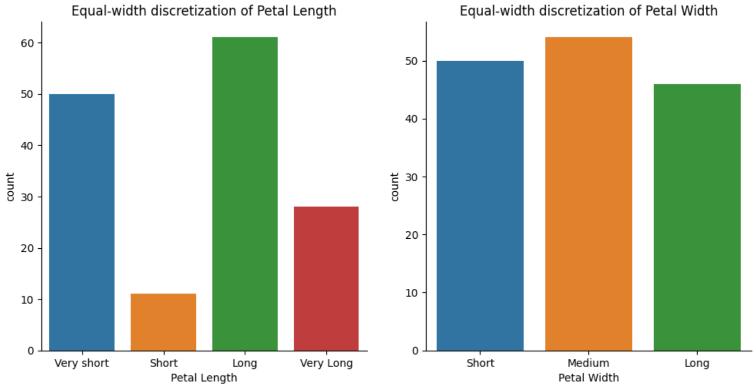
(b) Standardized Dataset, $\mathbf{D}_{standardized}$

Figure 4: Original dataset and standard score normalization (*i.e.*, standardization) are shown in Figures 4a and 4b respectively. Best practice is to make sure your plots have the same axes to compare the original and transformed datasets.

4.2 Categorical

Please access the Iris Flower Dataset via Sklearn and select the following two attributes: petal length and petal width.

1. Discretize the petal length into four groups (very short, short, long, very long) and petal width into three groups (short, medium, and long) using equal-width intervals.



(a) Petal Length

(b) Petal Width

Figure 5: Equal-width discretization for petal length and width are shown in Figures 5a and 5b respectively.

2. Create contingency table for the two attributes.

Petal Length (X_1)		Petal Width (X_2)			
		Short (a_{21})	Medium (a_{22})	Long (a_{23})	Row Counts
	Very Short (a_{11})	50	0	0	$n_1^1 = 50$
	Short (a_{12})	0	11	0	$n_2^1 = 11$
	Long (a_{13})	0	41	20	$n_1^3 = 61$
	Very Long (a_{14})	0	2	26	$n_1^4 = 28$
	Column Counts	$n_1^2 = 50$	$n_2^2 = 54$	$n_3^2 = 50$	$n = 150$

3. Are these two features independent? Why or why not? (hint: use Chi-squared test with $\alpha = 0.01$)

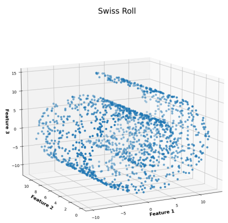
   The p value is less than α based on the Chi-squared independence test. Therefore, we reject the null hypothesis and these two features are not independent.

5 Dimensionality Reduction- 15 points

For this question, please download all three datasets provided along with this assignment: *swissroll.txt*, *spheres.txt*, and *ellipsoid.txt*. Import these files into your programming software. All datasets have 1500 samples and 3 features/dimensions. Now consider **D** to be a dataset (*i.e.*, size 1500 × 3 or $D \in \mathbb{R}^{1500\times3}$). For example, in Python, you can use a script to generate a plot of the dataset D:

```
import matplotlib.pyplot as plt
import numpy as np

D = np.loadtxt(“swissroll.txt”)
fig = plt.figure()
ax = plt.axes(projection=‘3d’)
ax.scatter3D(D[:,0], D[:,1], D[:,2])
ax.set_xlabel(‘Feature 1’, fontsize=12, =dict(weight=‘bold’))
ax.set_ylabel(‘Feature 2’, fontsize=12, fontdict=dict(weight=‘bold’))
ax.set_zlabel(‘Feature 3’, fontsize=12,fontdict=dict(weight=‘bold’))
ax.set_title(‘Swiss Roll’, fontsize = 20)
```



For each dataset:

1. Find the covariance matrix.

   (a) **Swiss Roll:** The covariance matrix of the "swiss roll" dataset is:

$$\mathbf{\Sigma}_{swissroll} = \begin{bmatrix} 43.2882 & 0.1535 & 4.4555 \\ 0.1535 & 10.5800 & 0.1544 \\ 4.4555 & 0.1544 & 47.1548 \end{bmatrix} \tag{12}$$

   (b) **Spheres:** The covariance matrix of the "spheres" dataset is:

$$\mathbf{\Sigma}_{spheres} = \begin{bmatrix} 10.0368 & 8.9743 & 9.0941 \\ 8.9743 & 9.9246 & 9.0678 \\ 9.0941 & 9.0678 & 10.2626 \end{bmatrix} \tag{13}$$

   (c) **Ellipsoids:** The covariance matrix of the "ellipsoids" dataset is:

$$\mathbf{\Sigma}_{ellipsoids} = \begin{bmatrix} 57.8661 & 14.6934 & 7.3664 \\ 14.6934 & 9.8630 & 4.5264 \\ 7.3664 & 4.5264 & 3.3570 \end{bmatrix} \tag{14}$$

2. Find the eigenvectors and eigenvalues of the covariance matrix.

   (a) **Swiss Roll:** The eigenvalues and orthonormal eigenvectors of the covariance of the "swiss roll" dataset is:

$$\begin{gathered} \lambda_1 = 50.0795, \mathbf{u}_1 = \begin{bmatrix} 0.5486 \\ 0.0054 \\ 0.8361 \end{bmatrix} \\ \lambda_2 = 40.3647, \mathbf{u}_2 = \begin{bmatrix} 0.8361 \\ 0.0015 \\ -0.5486 \end{bmatrix} \\ \lambda_3 = 10.5788, \mathbf{u}_3 = \begin{bmatrix} 0.0042 \\ -1.0000 \\ 0.0037 \end{bmatrix} \end{gathered} \tag{15}$$

   (b) **Spheres:** The eigenvalues and orthonormal eigenvectors of the covariance of the "spheres" dataset are:

$$\begin{gathered} \lambda_1 = 28.1668, \mathbf{u}_1 = \begin{bmatrix} -0.5760 \\ -0.5731 \\ -0.5829 \end{bmatrix} \\ \lambda_2 = 1.0571, \mathbf{u}_2 = \begin{bmatrix} -0.6207 \\ -0.1572 \\ 0.7681 \end{bmatrix} \\ \lambda_3 = 1.0001, \mathbf{u}_3 = \begin{bmatrix} 0.5318 \\ -0.8043 \\ 0.2652 \end{bmatrix} \end{gathered} \tag{16}$$

(c) **Ellipsoids:** The eigenvalues and orthonormal eigenvectors of the covariance of the "ellipsoids" dataset are:

$$\lambda_1 = 63.1653, \mathbf{u}_1 = \begin{bmatrix} 0.9518 \\ 0.2741 \\ 0.1380 \end{bmatrix}$$
$$\lambda_2 = 6.8856, \mathbf{u}_2 = \begin{bmatrix} 0.3068 \\ -0.8433 \\ -0.4413 \end{bmatrix} \tag{17}$$
$$\lambda_3 = 1.0352, \mathbf{u}_3 = \begin{bmatrix} 0.0046 \\ -0.4623 \\ 0.8867 \end{bmatrix}$$

3. Find (and plot) the projection of the data points into the 2-D and 1-D principal components (hint: use PCA and **do not** normalize data before PCA). After projecting the data into 2-D and 1-D, provide a short discussion (2-3 sentences) of the results for each dataset that answers the following question: Does the projection preserve the "important" or "most informative" structure for the original data? Why or why not? (hint: analyze quantitative and qualitative observations)

Discussion: Principal Components (PC) are orthogonal directions that capture most of the variance in the data. In PCA, lower-dimensional projections (1) are linear, and (2) only preserve the Euclidean distances between sample points. However, some data sets may contain non-linear structures that linear projections cannot preserve.

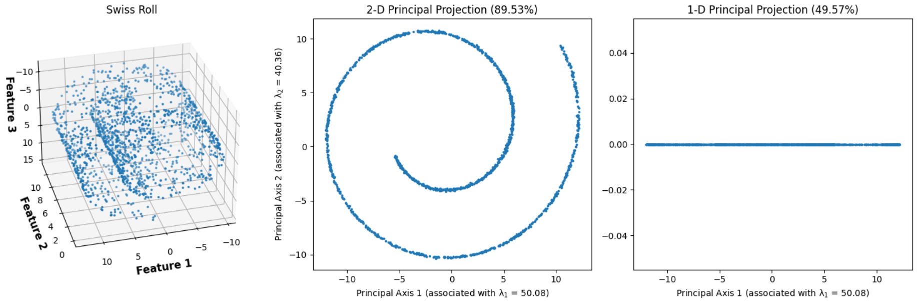


Figure 7: Visualization of "Swiss Roll" dataset and 2D/1D projections.

(a) **Swiss Roll:** Based on the eigenvalues of the covariance matrix, we can see that the 2-D projection preserves approximately

$$\frac{50.0795 + 40.3647}{50.0795 + 40.3647 + 10.5788} \approx 0.8953 \Rightarrow 89.53\%$$

of the variance of the original data. Similarly, the $1 - D$ projection preserves approximately

$$\frac{50.0795}{50.0795 + 40.3647 + 10.5788} \approx 0.4957 \Rightarrow 49.57\%$$

of the variance of the data. We see that the 1-D projection does not preserve the original variance of the data. At the same time, it fails to preserve the original underlying shape of the data because it condenses everything roughly into one connected line; the 2-D projection is better at preserving the variance and the most important structure.

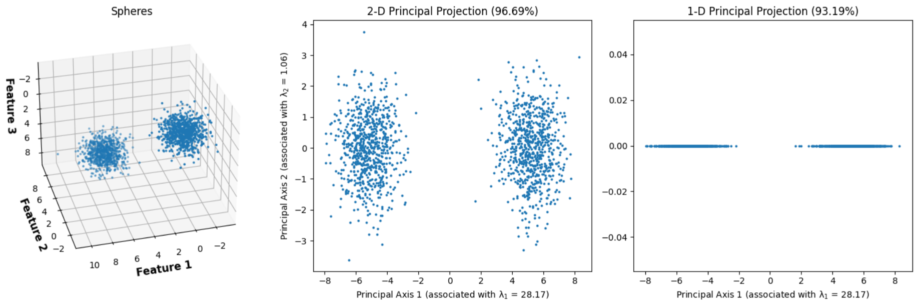


Figure 8: Visualization of "Spheres" dataset and 2D/1D projections.

(b) **Spheres:** Based on the eigenvalues of the covariance matrix, we can see that the 2-D projection preserves approximately

$$\frac{28.1668 + 1.057}{28.1668 + 1.057 + 1.0001} \approx 0.9669 \Rightarrow 96.69\%$$

of the variance of the original data. Similarly, the 1-D projection preserves approximately

$$\frac{28.1668}{28.1668 + 1.057 + 1.0001} \approx 0.9319 \Rightarrow 93.19\%$$

of the variance of the data. Visually, we see that the 2-D and 1-D projections preserve a lot of the original variance of the data while keeping the separation of the two spherical clusters; therefore, one can conclude that the 2-D and 1-D projections preserve the most informative structure of the data.

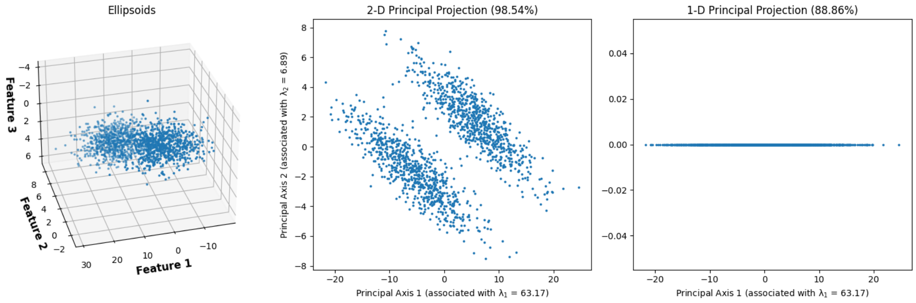


Figure 9: Visualization of "Ellipsoids" dataset and 2D/1D projections.

(c) **Ellipsoids:** Based on the eigenvalues of the covariance matrix, we can see that the 2-D projection preserves approximately

$$\frac{63.1653 + 6.8856}{63.1653 + 6.8856 + 1.0352} \approx 0.9854 \Rightarrow 98.54\%$$

of the variance of the original data. Similarly, the $1 - D$ projection preserves approximately

$$\frac{63.1653}{63.1653 + 6.8856 + 1.0352} \approx 0.8886 \Rightarrow 88.86\%$$

of the variance of the data. Both the 2-D and 1-D projections preserve a good amount of the original variance; however, it should be noted that in the 1-D projection, one can no longer distinguish between the two ellipsoids because the data points are all clumped together. In the 2-D representation, one can still see a separation between the two ellipsoids (thereby, seeing some structure that is not preserved in the 1-D case).