Homework Set 3, CPSC 8420, Spring 2022

Last Name, First Name

Due 03/31/2022, Thursday, 11:59PM EST

Problem 1

Given data-points $\{\{1,3\},\{2,5\},\{3,4\},\{4,3\},\{5,2\},\{5,1\}\}$.

1. Please scatter-plot each data point within one figure (you can use Matlab, Python or any other programming language).

   The scatter plot of the original data points in Fig. 1 is made with Mathematica. For the convenience of PCA, the data points centered at origin are also plotted in Fig. 1.2.

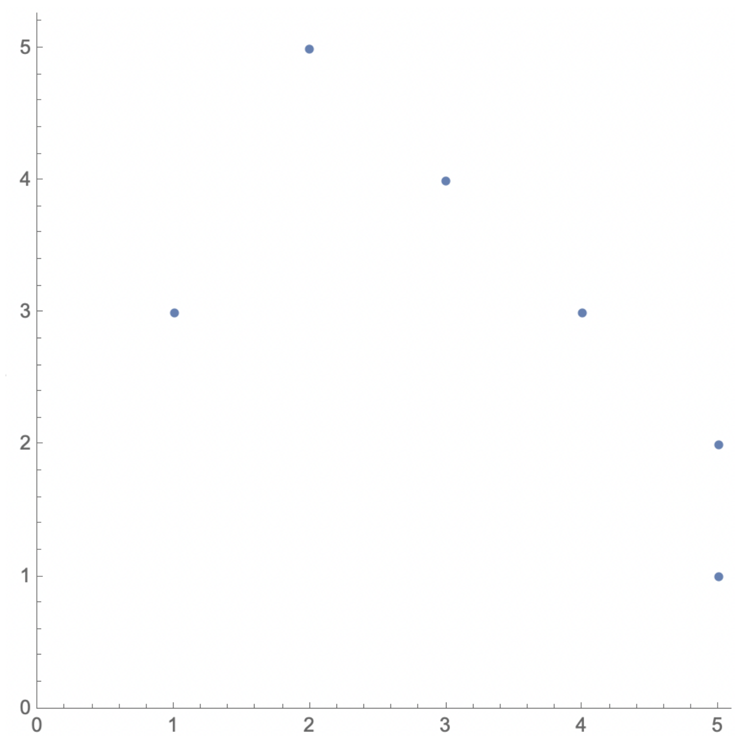

Figure 1: Scatter plot with original data points.

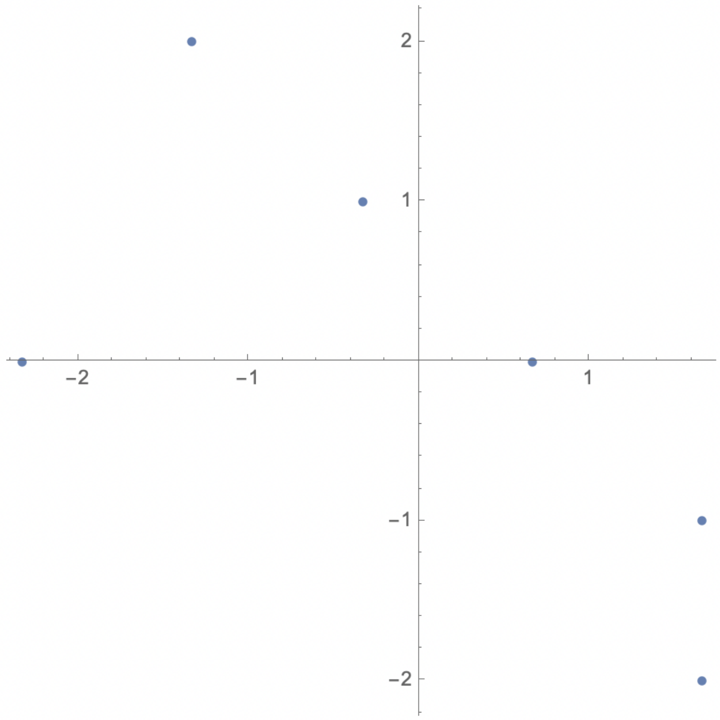

Figure 2: Scatter plot with centered data points.

Codes (Mathematica):

```
X = {{1, 3}, {2, 5}, {3, 4}, {4, 3}, {5, 2}, {5, 1}}
ListPlot[X, PlotRange -> {{0, Automatic}, {0, Automatic}},
AspectRatio -> 1]
Xcent = # - Mean[X] & /@ X
ListPlot[Xcent, PlotRange -> {{0, Automatic}, {0, Automatic}},
AspectRatio -> 1]
```

2. Now if we want to reduce the dimension from 2 to 1 by PCA, please determine the projection line which crosses the origin (please plot the line based on the scatter-plot figure above).

   Since the data points $\mathbf{X} = 1, 3, 2, 5, 3, 4, 4, 3, 5, 2, 5, 1$ is not a square matrix, we will perform PCA by $\mathbf{u}, vs., \mathbf{v} = SVD(\mathbf{X}^T\mathbf{X})$, and then obtain the loadings of the first principal component by taking the first column of $\mathbf{u}$. The data points and the projection line obtained with PCA are shown in Fig. 3.

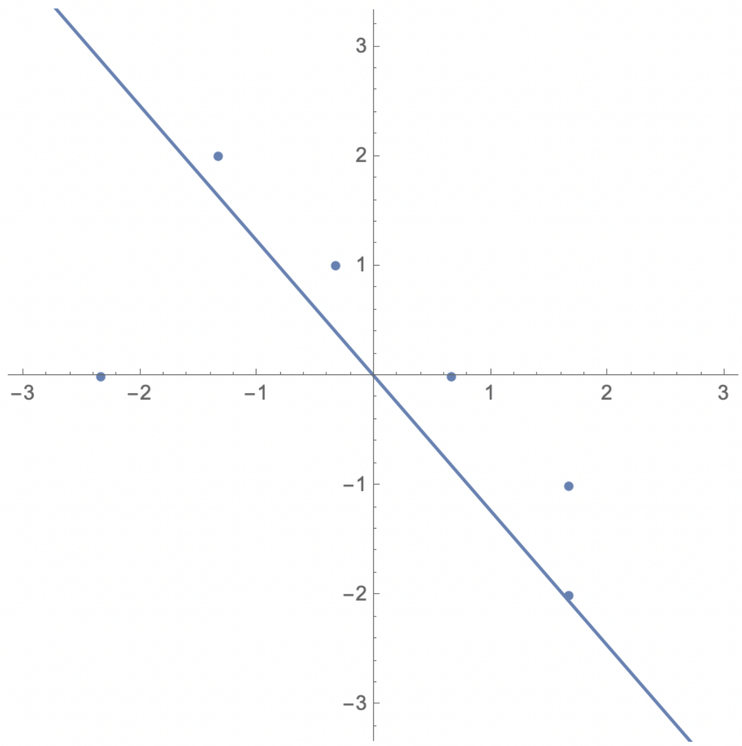

Figure 3: Scatter plot of data points and projection line obtained with PCA.

Codes (Mathematica):

```
{u, s, v} = N@SingularValueDecomposition[Transpose[X].X]
Show[Plot[u[[1, 1]]/u[[2, 1]] x, {x, 0, 5}], ListPlot[X],
PlotRange -> {{0, Automatic}, {0, Automatic}}, AspectRatio -> 1]
```

3. Assume the first 4 data points belong to one class, while the rest 2 belong to the other. Now if we want to reduce the dimension from 2 to 1 by LDA, please determine the projection line which crosses the origin (you are expected to plot the line based on the scatter-plot figure).

   The directional vector of the projection line in LDA is given by: $\mathbf{w} = \mathbf{S}^{-1}(\mu_0 - \mu_1)$, where $\mathbf{S}_w = \sum_{x \in \mathbf{X}_0}(\mathbf{x} - \mu_0)(\mathbf{x} - \mu_0)^T + \sum_{x \in \mathbf{X}_1}(\mathbf{x} - \mu_1)(\mathbf{x} - \mu_1)^T$, for the case with two classes $\mathbf{X}_0$ and $\mathbf{X}_1$. LDA is performed on the data points aftering centering at the origin. The projection line and the centered data points are shown in Fig. 4.

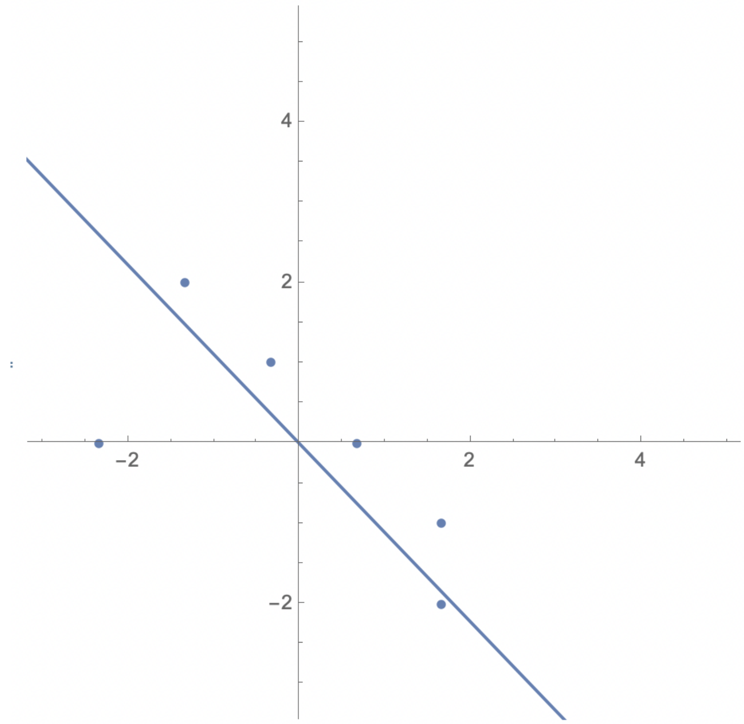

Figure 4: Scatter plot of data points and projection line obtained with LDA.

```
X1 = Xcent[[1 ;; 4]]
X2 = Xcent[[5 ;; 6]]
m1 = Mean[X1]; m2 = Mean[X2];
m1 = Mean[X1]; m2 = Mean[X2];
Sw = Plus @@ ((# - m1).Transpose[{# - m1}] & /@ X1) +
Plus @@ ((# - m2).Transpose[{# - m2}] & /@ X2);
w = 1/N @@ Sw*(m1 - m2)
Show[Plot[w[[1]]/w[[2]] x, {x, -5, 5}], ListPlot[Xcent],
PlotRange -> {{-3, 5}, {-3, 5}}, AspectRatio -> 1]
```

Problem 2

Given positive data-set $\{\{1,1\},\{2,2\},\{2,3\}\}$, as well as negative data-set $\{\{3,2\},\{3,3\},\{4,4\}\}$, please determine the decision boundary when leveraging k-NN where $k=1$ and $k=3$ respectively.

To determine the decision boundary, we can generate dense and evenly distributed grids in the 2D plain and classify them with k-NN method. The boundary between different classes of these densely distributed points then indicates the decision boundary of k-NN.
The k-NN method is implemented with the**Classify** function in Mathematica with the specification of classification method: **Method -¿ "NearestNeighbors"**. The points in different classes are colored in red and blue respectively, and the boundary between the red and blue regions indicates

the decision boundary.

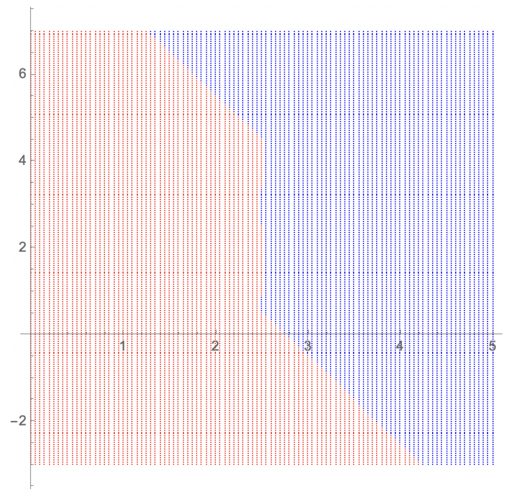

Figure 5: Decision boundary of k-NN when k=1.

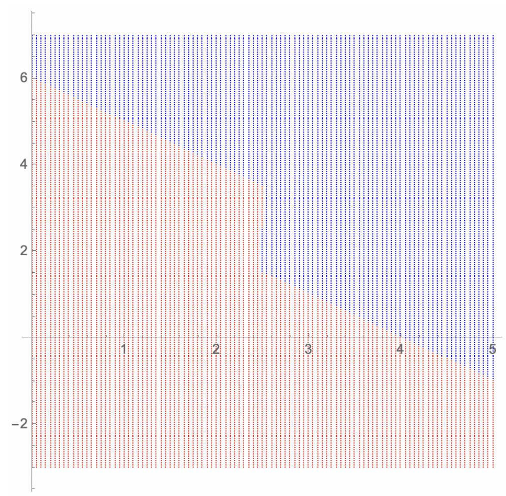

Figure 6: Decision boundary of k-NN when k=3.

With the increase of k from 1 to 3, we can see that the decision boundary is shifted closer to the positive and negative data sets when the boundaries are far away from these data sets.

Note: Ideally, the decision boundary would be completely straight and smooth at $x = 2.5$. However we may observe some inflections in the results shown in Figs. 5 and 6. This is mainly due to the insufficient density of the testing data points due to the limited computational power of my personal laptop.

Codes (Mathematica):

```
(*k=1*)
(*Generate testing data points*)
test = Flatten[
Table[{i, j}, {i, Range[0, 5, 0.05]}, {j, Range[-3, 7, 0.05]}],
1];
(*Nearest neighbor classifier with k=1*)
classifier =
Classify[Join[# -> 1 & /@ C1, # -> 2 & /@ C2],
Method -> {"NearestNeighbors", "NeighborsNumber" -> 1}];
clustering = AssociationThread[test, classifier[test]];
ListPlot[ParallelTable[
Style[x, If[clustering[x] == 1, Red,
If[clustering[x] == 2, Blue]]], {x, test}], AspectRatio -> 1]

(*k=3*)
(*Generate testing data points*)
test = Flatten[
Table[{i, j}, {i, Range[0, 5, 0.05]}, {j, Range[-3, 7, 0.05]}],
1];
(*Nearest neighbor classifier with k=3*)
classifier =
Classify[Join[# -> 1 & /@ C1, # -> 2 & /@ C2],
Method -> {"NearestNeighbors", "NeighborsNumber" -> 1}];
clustering = AssociationThread[test, classifier[test]];
ListPlot[ParallelTable[
Style[x, If[clustering[x] == 1, Red,
If[clustering[x] == 2, Blue]]], {x, test}], AspectRatio -> 1]
```

Problem 3

Given X, Y, Z, now please follow the idea/method used in LDA/PCA to find the best solution to:

$$\begin{gathered} \underbrace{arg\ max}_{a,b}\ a^T Z b \\ s.t.\ \ a^T X a = 1,\ b^T Y b = 1 \end{gathered} \tag{1}$$

The objective $\underbrace{arg\ max}_{a,b}\ a^T Z b$ is equivalent to $\underbrace{arg\ min}_{a,b}\ -a^T Z b$. Using two Lagrangian multipliers, λ_1 and λ_2 for the two constraints $a^T X a = 1$ and $b^T Y b = 1$, respectively, we can reformulate the objective:

$$\underbrace{arg\ min}_{a,b}\ -a^T Z b + \lambda_1(a^T X a - 1) + \lambda_2(b^T Y b - 1) = \underbrace{arg\ min}_{a,b}\ J \tag{2}$$

, where $J = -a^T Zb + \lambda_1(a^T Xa - 1) + \lambda_2(b^T Yb - 1)$ is the objective function. Taking the derivative of J w.r.t. a and b, we get:

$$\frac{\partial J}{\partial a} = -b^T Z^T + \lambda_1 a^T (X + X^T) = 0 \tag{3}$$

$$\frac{\partial J}{\partial b} = -a^T Z + \lambda_2 b^T (Y + Y^T) = 0 \tag{4}$$

Since $a^T Xa = a^T X^T a = 1$, by right-multiplation of a on both sides of eqn. (3), we get:

$$\begin{aligned} &- b^T Z^T a + \lambda_1 a^T (X + X^T)a = 0 \\ &b^T Z^T a = (a^T Zb)^T = 2\lambda_1 \end{aligned} \tag{5}$$

By right-multiplation of b on both sides of eqn. (4), we get:

$$\begin{aligned} &- a^T Zb + \lambda_2 b^T (Y + Y^T)b = 0 \\ &a^T Zb = 2\lambda_2 \end{aligned} \tag{6}$$

Combining Eqn. (5) and (6), we have: $(a^T Zb)^T a^T Zb = 4\lambda_1\lambda_2$, which means that to maximize $a^T Zb$ is equivalent to maximizing $\lambda_1\lambda_2$. Take the transpose on both sides of Eqn. (3) and (4), we get:

$$\begin{aligned} &Zb = \lambda_1(X + X^T)a \\ &b = \lambda_1 Z^{-1}(X + X^T)a \end{aligned} \tag{7}$$

and

$$Z^T a = \lambda_2(Y + Y^T)b \quad a = \lambda_2(Z^T)^{-1}(Y + Y^T)b \tag{8}$$

Plug Eqn. (7) into Eqn. (8), we get:

$$\begin{aligned} &Z^T a = \lambda_1\lambda_2(Y + Y^T)Z^{-1}(X + X^T)a \\ &[(X + X^T)^{-1}Z(Y + Y^T)^{-1}Z^T]a = \lambda_1\lambda_2 a \end{aligned} \tag{9}$$

Define $A = (X + X^T)^{-1}Z(Y + Y^T)^{-1}Z^T$, and Eqn. (9) can be seen as an eigenvalue problem:

$$Aa = \lambda_1\lambda_2 a \tag{10}$$

Similarly, plug Eqn. (8) into (7), we can get:

$$[(Y + Y^T)^{-1}Z^T(X + X^T)^{-1}Z]b = \lambda_1\lambda_2 b \tag{11}$$

Define $B = (Y + Y^T)^{-1}Z^T(X + X^T)^{-1}Z$, we can similarly obtain:

$$Bb = \lambda_1\lambda_2 b \tag{12}$$

The eigenvalue decomposition is cummulative w.r.t. matrix multiplication, i.e. the eigenvalues and eigenvectors of $(MN)v = \sigma v$ is the same as $(MN)v = \sigma v$. Note that if we define $M = (X+X^T)^{-1}Z$ and $N = (Y + Y^T)^{-1}Z^T$, then $A = MN$, $B = NM$, and therefore A and B have the same set of eigenvalues $\{\sigma_i\}$ and corresponding eigenvectors $\{v_i\}$ $(i = 1, 2, ..., n)$, which are sorted in decreasing order of σ_i. To maximize $(a^T Zb)^T$, or equivalently $\lambda_1\lambda_2$, we can set:

$$a = b = v_1 \tag{13}$$

, which is the eigenvector of A and B corresponding to the largest eigenvalue $\lambda_1\lambda_2 = \sigma_1$.