HW 3

Bayard Walsh

April 2023

1

1.1

For X, we have

$$x_1 = \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix}, x_2 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix}$$

Therefore, our first orthnormal vector for x_1 is below. As $x_1 v_1 = 0$ for every column except $i = j = 1$, it is orthogonal. Also, $x_1 x_1 = 1$, so the vector is normal. Next we do some calculations to calculate the difference of v_1 from x_2 :

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

$$x_2' = x_2 - v_1 v_1^T x_2 = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} - (\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix})$$

$$x_2' = \begin{bmatrix} 2 \\ 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 3 \\ 4 \end{bmatrix}$$

Now, we have the foundation of v_2 which will be orthogonal to x_2', and we know it is orthogonal to v_1. Therefore we want it to be normal. We achieve this through the Gram–Schmidt process with respect to the desired vector $x'2$. This gives us

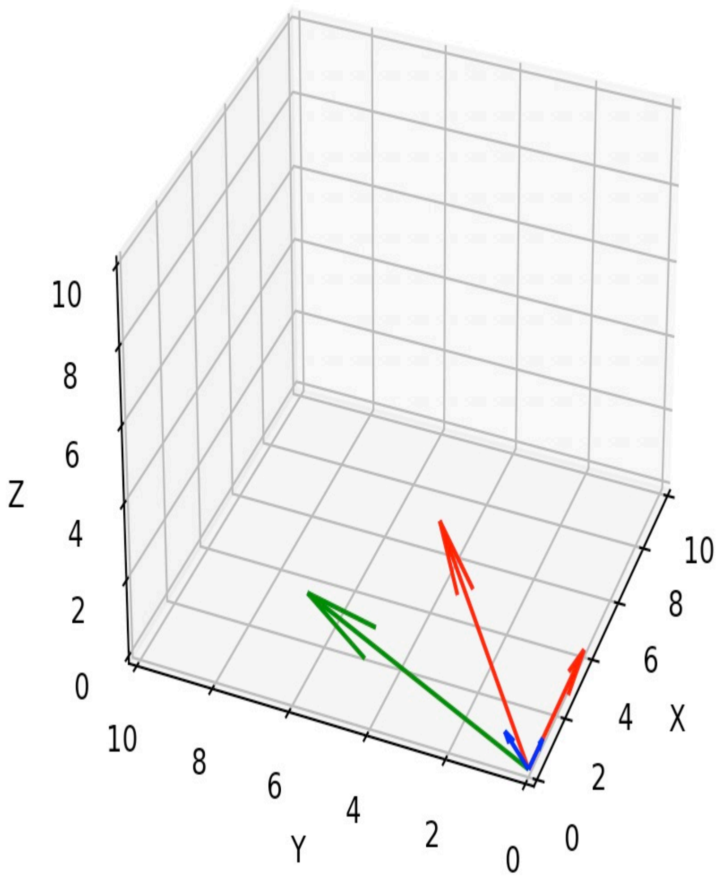
$$v_2 = \begin{bmatrix} 0 \\ 3/(\sqrt{(}3^2 + 4^2)) \\ 4/(\sqrt{(}3^2 + 4^2)) \end{bmatrix} = \begin{bmatrix} 0 \\ 3/5 \\ 4/5 \end{bmatrix}$$

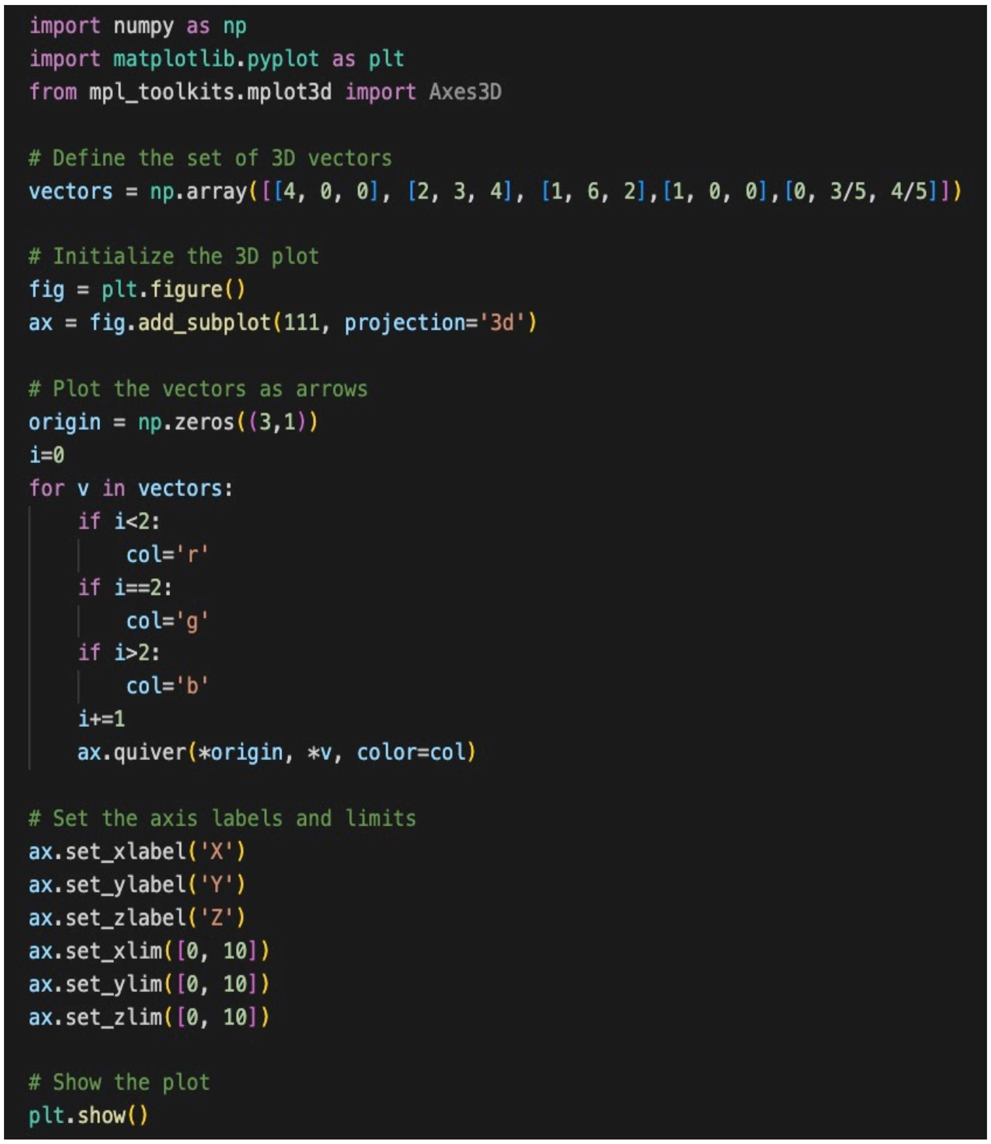
Therefore we have the following two orthonormal vectors that span the plane of the columns of X

$$v_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 3/5 \\ 4/5 \end{bmatrix}$$

1.2

Graph achieved through simple python script (see code below). Note that x vectors are red, y is green, and normalized vectors are blue





```python
import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Define the set of 3D vectors
vectors = np.array([[4, 0, 0], [2, 3, 4], [1, 6, 2],[1, 0, 0],[0, 3/5, 4/5]])

# Initialize the 3D plot
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

# Plot the vectors as arrows
origin = np.zeros((3,1))
i=0
for v in vectors:
    if i<2:
        col='r'
    if i==2:
        col='g'
    if i>2:
        col='b'
    i+=1
    ax.quiver(*origin, *v, color=col)

# Set the axis labels and limits
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_xlim([0, 10])
ax.set_ylim([0, 10])
ax.set_zlim([0, 10])

# Show the plot
plt.show()
```

1.3

From the Orthnormal bases and least squares proof from class, we make the following substitution, where U is an orthnomoral basis for the subspace of X.

$$\hat{y} = X(X^TX)^{-1}X^Ty = UU^Ty$$

Note that we have computed v_1, v_2 as the orthnomoral basis for the subspace of X in $1a$), so we will now substitute and solve

$$\hat{y} = UU^T y = \begin{bmatrix} 1 & 0 \\ 0 & 3/5 \\ 0 & 4/5 \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 & 0 \\ 0 & 3/5 & 4/5 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 6 \\ 2 \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 9/25 & 12/25 \\ 0 & 12/25 & 16/25 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 6 \\ 2 \end{bmatrix}$$

$$\hat{y} = \begin{bmatrix} 1 \\ 78/25 \\ 104/25 \end{bmatrix}$$

2

2.1

First we calculate our basis for the subspace. From $2x_1 - 3x_2 + x_3 = 0$, we can make the following matrix derived by the relation $x_3 = -2x_1 + 3x_2 +$:

$$\begin{bmatrix} x_1 \\ x_2 \\ -2x_1 + 3x_2 \end{bmatrix}$$

From here, we separate into the following two vectors

$$x_1 = \begin{bmatrix} 1 \\ 0 \\ -2 \end{bmatrix} x_2 = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix}$$

First we will compute the first orthnormal vector, which is done through Graham Schmidt algorithm :

$$v_1 = \begin{bmatrix} 1 \\ 0/(\sqrt{(}1^2 + (-2)^2)) \\ -2/(\sqrt{(}1^2 + (-2)^2)) \end{bmatrix} = \begin{bmatrix} 1/\sqrt{5} \\ 0 \\ -2/\sqrt{5} \end{bmatrix}$$

Next, we compute the second orthnormal vector as follows:

$$x_2' = x_2 - v_1 v_1^T x_2 = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} - (\begin{bmatrix} 1/5 & 0 & -2/5 \\ 0 & 0 & 0 \\ -2/5 & 0 & 4/5 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix})$$

$$x_2' = \begin{bmatrix} 0 \\ 1 \\ 3 \end{bmatrix} - \begin{bmatrix} -6/5 \\ 0 \\ 12/5 \end{bmatrix} = \begin{bmatrix} 6/5 \\ 1 \\ 3/5 \end{bmatrix}$$

$$v_2 = \begin{bmatrix} 6/5/(\sqrt{(}36/25 + 1 + 9/25)) \\ 1/(\sqrt{(}36/25 + 1 + 9/25)) \\ 3/5/(\sqrt{(}36/25 + 1 + 9/25)) \end{bmatrix} = \begin{bmatrix} 6/\sqrt{70} \\ 5/\sqrt{70} \\ 3/\sqrt{70} \end{bmatrix}$$

Now we have

$$V = \begin{bmatrix} 1/\sqrt{5} & 6/\sqrt{70} \\ 0 & 5/\sqrt{70} \\ -2/\sqrt{5} & 3/\sqrt{70} \end{bmatrix}$$

As an orthnormal basis for the subspace S. As we have $P_x = VV^T$, we can compute P_x as follows:

$$P_x = VV^T = \begin{bmatrix} 1/\sqrt{5} & 6/\sqrt{70} \\ 0 & 5/\sqrt{70} \\ -2/\sqrt{5} & 3/\sqrt{70} \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{5} & 0 & -2/\sqrt{5} \\ 6/\sqrt{70} & 5/\sqrt{70} & 3/\sqrt{70} \end{bmatrix}$$

$$P_x = VV^T = \begin{bmatrix} \frac{5}{7} & \frac{3}{7} & -\frac{1}{7} \\ \frac{3}{7} & \frac{5}{14} & \frac{3}{14} \\ -\frac{1}{7} & \frac{3}{14} & \frac{13}{14} \end{bmatrix}$$

2.2

As P_x can be represented in a $3 \cdot 2$ matrix and its transpose, we know $\text{rank}(P_x) = 2$

2.3

As we have the projection of vector X onto S gives the closest point in S to X by definition, we can use the following relation to compute distance:

$$D_x = x - (P_x \cdot x) = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} \frac{5}{7} & \frac{3}{7} & -\frac{1}{7} \\ \frac{3}{7} & \frac{5}{14} & \frac{3}{14} \\ -\frac{1}{7} & \frac{3}{14} & \frac{13}{14} \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$D_x = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$D_x = 0$$

Therefore the distance to the subspace is 0

3

3.1

As we want to label flowers across three types, we need to determine bounds for each label. Therefore, a general classification strategy we can implement would be finding three bounds as follows:

assume $a > b$

$$Fw = \begin{cases} \text{plant 1: } 1, & \text{if } Fw \geq a \\ \text{plant 2: } 0, & \text{if } a > Fw \geq b \\ \text{plant 3: } -1, & \text{if } b > Fw \end{cases}$$

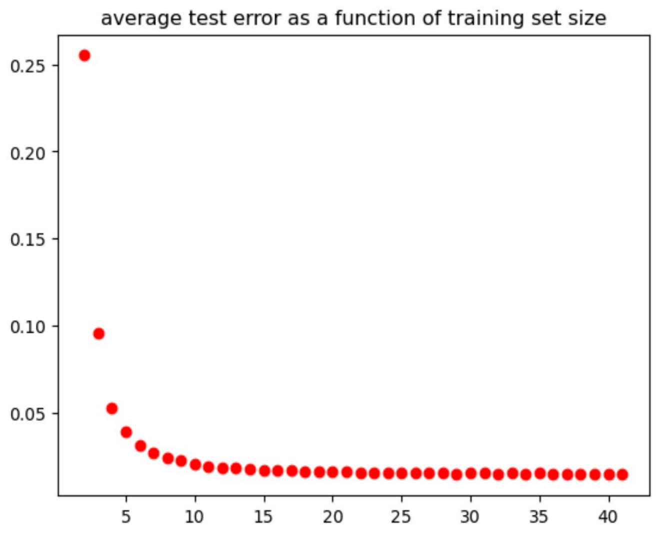
This will solve the least squares problem of given real value prediction because we have expanded our bounds from positive and negative to three discrete categories. Note that in order to determine the best a, b, some trial and error would be needed, but we will start with the bounds $a = .5$ and $b = -.5$. Note that a number is given after each plant name, and which we will use this number to compare predicted labels for our estimation.

3.2

Average test error: 0.014006666666666666

code for solution is included in jupyter notebook on separate page

3.3



code for solution is included in jupyter notebook on separate page

3.4

Average test error: 0.017859999999999997

code for solution is included in jupyter notebook on separate page

3.5

Forming 2- dimensional subspace that the data approximately lie in with given vectors:

$$v_1 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix} v_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

These vectors were picked visually for approximate fit and for being orthnormal.

code for solution is included in jupyter notebook on separate page. In this case graphs of initial data and data with plane on it are included

3.6

Average test error: 0.09783
code for solution is included in jupyter notebook on separate page. Used projection matrix to reduce 3 dimensions of classifications to 2 dimensions.

4

4.1

For $F(w)$ $w = 9.639884$ minimizes the function. See code for script on minimizing with gradient descent. This code returns $w = 9.63988399$ which is essentially the minimum. For this $F(w)$ my function always finds the best w via gradient descent. If I decrease w_0, I may find a false min on the other half of the graph, at roughly -9.619 for example. For this function there are many local min, so random w_0 assignment could be a viable strategy to find the w given in the solution (as mentioned in class). Another byproduct of a different w_0 is overall more steps to reach my end w. For the current w_0 my step size is sufficient. If I increase my step size too much, I may not find the optimal solution because I do not have enough granularity. If I decrease my step size I would take more iterations but eventually reach the same end w.

4.2

We have (as given)
$X = w_0 + w_1x + w_2x^2 + w_3x^3 + w_4x^4 + w_5x^5$
$y = cos(3x)$

$$w = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \end{bmatrix}$$

From these definitions we can reach the least squares estimate definition of $f(w)$ as follows:

$$f(w) = \text{argmin w}||y - Xw||^2$$

As we cannot assume that a columns are linearly independent, we take the long form definition of argmin w.

$$f(w) = \text{argmin w}||y - Xw||^2 = y^T y - y^T Xw - w^T X^T y + w^T X^T Xw$$

This is the estimate for solving the problem with least squares, though we are primarily focused on the gradient for the implementation of gradient descent. This is computed below

$$f(w) = -2X^T Xy + 2X^T Xw$$

code for solution is included in jupyter notebook on separate page.