ECE 146 Midterm Exam

↳ Zack Berger, 705082271

5/4/20

Parameter $d = (1 \% 8) + 1 = 2$ $\boxed{d=2}$

"I, Zack Berger, with UID 705082271, have read and understood the policy on academic dishonesty available on the course website"

\- Zack Berger

Question 1 - Perceptron

a) Perceptron learning rules...

i. x is falsely classified as positive : $w^{t+1} = w^t - \eta x$

ii. x is falsely classified as negative : $w^{t+1} = w^t + \eta x$

Because if x misclassified as positive, y is negative 1 and vice versa, and the update rule is $w^{t+1} = w^t + y\eta x$.

b) consider the training set

Sample #	1	2	3	4
x	[2,2]	[-2,-4]	[-8,-16]	[3,1]
y	+1	+1	-1	-1

- Use Max Iter = 1
- Order as in table
- Initial weight $w = [w_0, w_1, w_2]^T = [0,1,1]^T$.
- use $\eta = 1$
- Show weights at each step of algorithm.

Initial weights: $w = [0,1,1]^T$

Transform data from 2D to 3D: $X_1 = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$, $X_2 = \begin{bmatrix} 1 \\ -2 \\ -4 \end{bmatrix}$, $X_3 = \begin{bmatrix} 1 \\ -8 \\ -16 \end{bmatrix}$, $X_4 = \begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix}$

↳ • Evaluate at x_1 : $y_1 w^T x_1 = (+1)(+4) = 4 > 0$.

So, x_1 is correctly classified.

Do <u>not</u> update w. $w' = [0,1,1]^T$

• Evaluate at x_2 : $y_2 w^T x_2 = (+1)(-2-4) = (+1)(-6) = -6 \leq 0$.

So, x_2 is incorrectly classified. Update w!

$$w^2 = w^1 + y_2 x_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} + (+1)\begin{bmatrix} 1 \\ -2 \\ -4 \end{bmatrix} = \begin{bmatrix} 0+1 \\ 1-2 \\ 1-4 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ -3 \end{bmatrix}$$

$\Rightarrow$ $w^2 = [1 \ -1 \ -3]^T$

⟶

- Evaluate at x_3: $y_3 w^T x_3 = (-1)\begin{bmatrix} 1 & -1 & -3 \end{bmatrix}\begin{bmatrix} 1 \\ -8 \\ -16 \end{bmatrix} = (-1)(57) = -57 \leq 0.$

So, x_3 is incorrectly classified. Update w!

$$w^3 = w^2 + y_3 x_3 = \begin{bmatrix} 1 \\ -1 \\ -3 \end{bmatrix} + (-1)\begin{bmatrix} 1 \\ -8 \\ -16 \end{bmatrix} = \begin{bmatrix} 1-1 \\ -1+8 \\ -3+16 \end{bmatrix} = \begin{bmatrix} 0 \\ 7 \\ 13 \end{bmatrix}$$

$$\Rightarrow \underline{w^3 = \begin{bmatrix} 0 & 7 & 13 \end{bmatrix}^T}$$

- Evaluate at x_4: $y_4 w^T x_4 = (-1)\begin{bmatrix} 0 & 7 & 13 \end{bmatrix}\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = (-1)(34) = -34 \leq 0.$

So, x_4 is incorrectly classified. Update w!

$$w^4 = w^3 + y_4 x_4 = \begin{bmatrix} 0 \\ 7 \\ 13 \end{bmatrix} + (-1)\begin{bmatrix} 1 \\ 3 \\ 1 \end{bmatrix} = \begin{bmatrix} -1 \\ 4 \\ 12 \end{bmatrix}$$

$$\Rightarrow \underline{w^4 = \begin{bmatrix} -1 & 4 & 12 \end{bmatrix}^T}$$

Summary of results

$$\boxed{\begin{array}{ll} w_{initial} = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T & w_3 = \begin{bmatrix} 0 & 7 & 13 \end{bmatrix}^T \\ w_1 = \begin{bmatrix} 0 & 1 & 1 \end{bmatrix}^T & w_4 = \begin{bmatrix} -1 & 4 & 12 \end{bmatrix}^T \\ w_2 = \begin{bmatrix} 1 & -1 & -3 \end{bmatrix}^T & \end{array}}$$

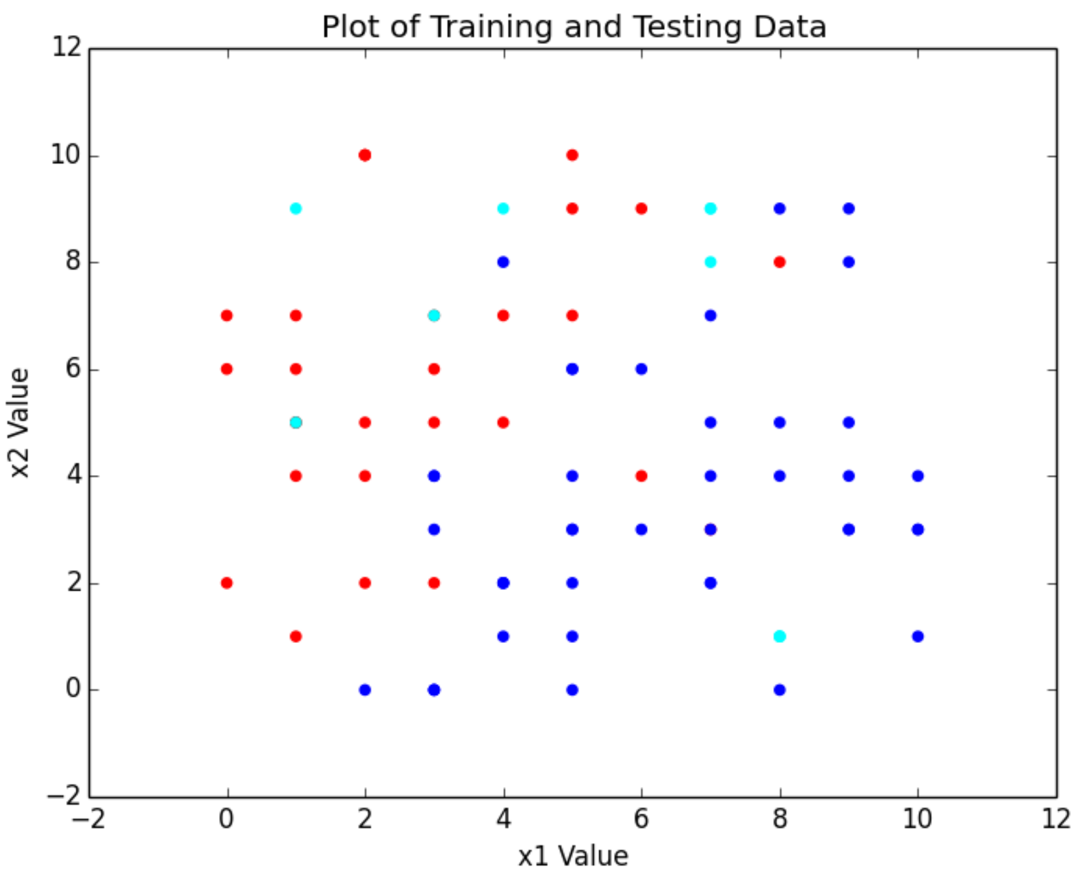
Zack Berger

Question 2

Part A

Plot of training data with red points denorting label 1, blue points denoting label 0. Includes splot of testing data with color cyan. By inspection of the plot, the data is NOT linearly separable.

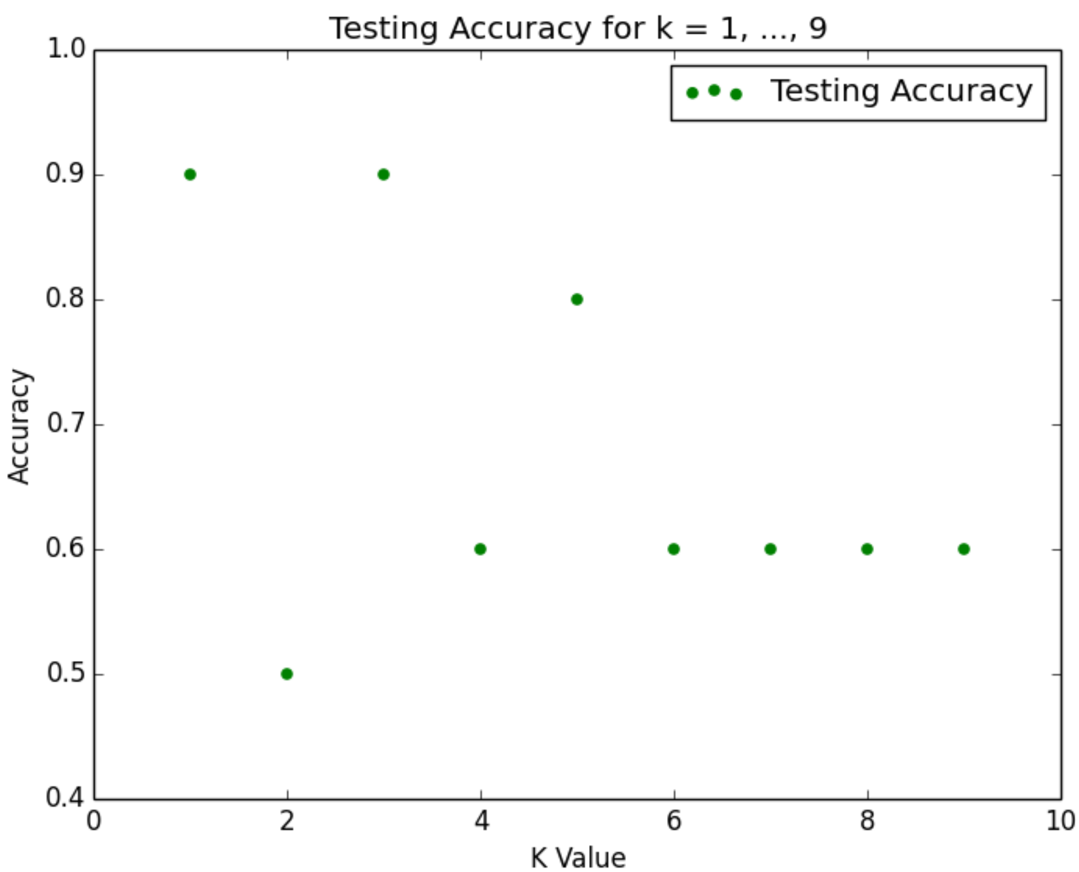
Note that some points overlap, which was allowed by the problem spec:



Part B

The testing accuracies for k = 1, …, 9 are represented in the following table and plot:

K	1	2	3	4	5	6	7	8	9
Testing Accuracy	0.9	0.5	0.9	0.6	0.8	0.6	0.6	0.6	0.6



Scanned printout of code is attached to this problem…

```python
#!/usr/bin/env python
"""Analysis of midterm dataset using K-NN algorithm"""

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

# Define class that implements K-NN algorithm
class Knear:
    def __init__(self, trainingData, trainingLabels):
        self.trainingData = trainingData
        self.trainingLabels = trainingLabels

        # If len(trainingData) != len(trainingLabels), throw error
        self.datamt = len(trainingData)

    # Return an array of predicted labels for testing_data
    def test(self, testing_data, k_value, y_tie):

        labels = []
        for i in range(0, len(testing_data)):
            val = self.testPoint(testing_data[i], k_value, y_tie)
            labels.append(val)

        return labels

    # Test an individual point and return its label
    def testPoint(self, point, k_value, y_tie):

        # Obtain array of l one distances from point to training set
        distances = np.array([])
        for i in range(0, self.datamt):
            dist = l_one_distance(point, self.trainingData[i])
            distances = np.append(distances, dist)

        # Obtain the k lowest indices, which are the k nearest neighbors
        # Use mergesort, which is a stable sorting algorithm
        nearest_neighbors = distances.argsort(kind='stable')[:k_value]

        # Count the 1s and 0s in the k nearest neighbors
        ones = 0
        zeros = 0

        for i in range(0, len(nearest_neighbors)):
            if self.trainingLabels[nearest_neighbors[i]] == 1:
                ones += 1
            else:
                zeros += 1

        # Determine the label of the test point
        if ones > zeros:
```

```
            return 1
        elif zeros > ones:
            return 0
        else:
            return y_tie


def main():

    # Import and parse all training and testing data
    # Because alpha = 2, want 11th to 20th rows as testing, rest as training
    data = np.loadtxt("/Users/zackberger/Desktop/Ml/Exams/Q2data.csv",
     delimiter=',')

    training_info = np.concatenate([ data[0:10], data[20:] ])
    testing_info = data[10:20]

    training_data = training_info[:,:2]
    training_labels = training_info[:,2]

    testing_data = testing_info[:,:2]
    testing_labels = testing_info[:,2]

    # Plot data
    for i in range(0, len(training_data)):
        if training_labels[i] == 1:
            plt.scatter(training_data[i,0], training_data[i,1], color='r')
        else:
            plt.scatter(training_data[i,0], training_data[i,1], color='b')

    plt.scatter(testing_data[:,0], testing_data[:,1], color='cyan')
    plt.title('Plot of Training and Testing Data')
    plt.xlabel("x1 Value")
    plt.ylabel("x2 Value")
    plt.legend(loc="upper right")
    plt.show()

    # Instantiate K-NN framework with training data
    knn = Knear(training_data, training_labels)

    # For even k, if number of points from class 0 is same as number of points
     from class 1, classify
    # this data point as class 0 deterministically by setting the parameter
     tiebreaker = 0
    tiebreaker = 0

    test1 = knn.test(testing_data, 1, tiebreaker)
    acc1 = testing_accuracy(test1, testing_labels)

    test2 = knn.test(testing_data, 2, tiebreaker)
    acc2 = testing_accuracy(test2, testing_labels)
```

```
    test3 = knn.test(testing_data, 3, tiebreaker)
    acc3 = testing_accuracy(test3, testing_labels)

    test4 = knn.test(testing_data, 4, tiebreaker)
    acc4 = testing_accuracy(test4, testing_labels)

    test5 = knn.test(testing_data, 5, tiebreaker)
    acc5 = testing_accuracy(test5, testing_labels)

    test6 = knn.test(testing_data, 6, tiebreaker)
    acc6 = testing_accuracy(test6, testing_labels)

    test7 = knn.test(testing_data, 7, tiebreaker)
    acc7 = testing_accuracy(test7, testing_labels)

    test8 = knn.test(testing_data, 8, tiebreaker)
    acc8 = testing_accuracy(test8, testing_labels)

    test9 = knn.test(testing_data, 9, tiebreaker)
    acc9 = testing_accuracy(test9, testing_labels)

    x = [1, 2, 3, 4, 5, 6, 7, 8, 9]
    accuracies = [acc1, acc2, acc3, acc4, acc5,  acc6, acc7, acc8, acc9]
    print('Testing Accuracies:')
    print(accuracies)

    plt.scatter(x, accuracies, color='g', label='Testing Accuracy')
    plt.title('Testing Accuracy for k = 1, ..., 9')
    plt.xlabel('K Value')
    plt.ylabel('Accuracy')
    plt.legend(loc="upper right")
    plt.show()


# Compute percentage of matches in two equivalently sized arrays
def testing_accuracy(arr1, arr2):
    matches = 0

    for i in range(0, len(arr1)):
        if arr1[i] == arr2[i]:
            matches += 1

    return ( float(matches) / len(arr1) )


# Find the l_one distance between two pointss with two attributes
def l_one_distance(arr1, arr2):

    return ( abs(arr1[0] - arr2[0]) + abs(arr1[1] - arr2[1]) )
```

```
# Name guard
if __name__ == "__main__":
    main()
```

Question 3 - Decision Tree

a) what is the binary entropy of H(ECE146)?

$\hookrightarrow P(ECE146 = 1) = \frac{5}{8} \qquad P(ECE146 = 0) = \frac{3}{8}$

$H(ECE146) = -\frac{5}{8}\log\frac{5}{8} - \frac{3}{8}\log\frac{3}{8} \approx 0.954434$

$$\boxed{H(ECE146) = 0.9544}$$

NOTE: using log base 2 throughout this problem, i.e. $\log := \log_2$.

b) Calculate $H(ECE146|X)$ for $X \in \{ECE102, ECE131, MATH61, MUSIC\}$

i. $H(ECE146 | ECE102)$

$\hookrightarrow P(ECE102 = 1) = \frac{4}{8} \qquad P(ECE102 = 0) = \frac{4}{8}$

- $P(ECE146 = 1 | ECE102 = 1) = \dfrac{P(ECE146 = 1, ECE102 = 1)}{P(ECE102 = 1)} = \dfrac{3/8}{4/8} = \dfrac{3}{4}$
- $P(ECE146 = 0 | ECE102 = 1) = \dfrac{P(ECE146 = 0, ECE102 = 1)}{P(ECE102 = 1)} = \dfrac{1/8}{4/8} = \dfrac{1}{4}$
- Similarly, $P(ECE146 = 1 | ECE102 = 0) = \dfrac{2/8}{4/8} = \dfrac{2}{4} = \dfrac{1}{2}$

  $P(ECE146 = 0 | ECE102 = 0) = \dfrac{2/8}{4/8} = \dfrac{2}{4} = \dfrac{1}{2}$

- $H(ECE146 | ECE102 = 1) = -P(ECE146 = 1 | ECE102 = 1)\log P(ECE146 = 1 | ECE102 = 1)$

  $\qquad - P(ECE146 = 0 | ECE102 = 1)\log P(ECE146 = 0 | ECE102 = 1)$

  $\qquad = -\frac{3}{4}\log\frac{3}{4} - \frac{1}{4}\log\frac{1}{4} \approx 0.811278$

- $H(ECE146 | ECE102 = 0) = -\frac{1}{2}\log\frac{1}{2} - \frac{1}{2}\log\frac{1}{2} = 1$

$\Rightarrow H(ECE146 | ECE102) = P(ECE102 = 0)H(ECE146 | ECE102 = 0) + P(ECE102 = 1)H(ECE146 | ECE102 = 1)$

$\qquad = \left(\frac{4}{8}\right)(1) + \left(\frac{4}{8}\right)(0.8113)$

$\qquad = \underline{0.90565}$

$\longrightarrow$

ii. $H(ECE146 \mid ECE131)$

- $P(ECE131=0) = \frac{4}{8}$ $\quad P(ECE131=1) = \frac{4}{8}$
- $P(ECE146=1 \mid ECE131=1) = \frac{4/8}{4/8} = \frac{4}{4}$ $\quad P(ECE146=0 \mid ECE131=1) = 0$
- $P(ECE146=1 \mid ECE131=0) = \frac{1}{4}$ $\quad P(ECE146=0 \mid ECE131=0) = \frac{3}{4}$
- $H(ECE146 \mid ECE131=1) = -1\log 1 - 0\log 0 = 0$
- $H(ECE146 \mid ECE131=0) = -\frac{1}{4}\log\frac{1}{4} - \frac{3}{4}\log\frac{3}{4} \approx 0.811278$

$\Rightarrow H(ECE146 \mid ECE131) = \frac{4}{8}(0.811278) + \frac{4}{8}(0) = \underline{0.40565}$

iii. $H(ECE146 \mid MATH61)$

- $P(MATH61=0) = \frac{5}{8}$ $\quad P(MATH61=1) = \frac{3}{8}$
- $P(ECE146=1 \mid MATH61=1) = 1$ $\quad P(ECE146=0 \mid MATH61=1) = 0$
- $P(ECE146=1 \mid MATH61=0) = \frac{2}{5}$ $\quad P(ECE146=0 \mid MATH61=0) = \frac{3}{5}$
- $H(ECE146 \mid MATH61=1) = -1\log 1 - 0\log 0 = 0$
- $H(ECE146 \mid MATH61=0) = -\frac{2}{5}\log\frac{2}{5} - \frac{3}{5}\log\frac{3}{5} \approx 0.97095$

$\Rightarrow H(ECE146 \mid MATH61) = \frac{5}{8}(0.97095) + \frac{3}{8}(0) \approx \underline{0.606844}$

iv. $H(ECE146 \mid MUSC15)$

- $P(MUSC15=0) = \frac{3}{8}$ $\quad P(MUSC15=1) = \frac{5}{8}$
- $P(ECE146=1 \mid MUSC15=1) = \frac{3}{5}$ $\quad P(ECE146=0 \mid MUS15=1) = \frac{2}{5}$
- $P(ECE146=1 \mid MUS15=0) = \frac{2}{3}$ $\quad P(ECE146=0 \mid MUSC15=0) = \frac{1}{3}$
- $H(ECE146 \mid MUSC15=1) = -\frac{3}{5}\log\frac{3}{5} - \frac{2}{5}\log\frac{2}{5} \approx 0.97095$
- $H(ECE146 \mid MUSC15=0) = -\frac{2}{3}\log\frac{2}{3} - \frac{1}{3}\log\frac{1}{3} \approx 0.918296$

$\Rightarrow H(ECE146 \mid MUSC15) = \frac{3}{8}(0.918296) + \left(\frac{5}{8}\right)(0.97095) \approx \underline{0.95120}$

$H(ECE146 \mid ECE102) = 0.90565$
$H(ECE146 \mid ECE131) = 0.40565$
$H(ECE146 \mid MATH61) = 0.606844$
$H(ECE146 \mid MUSC15) = 0.95120$

c) Calculate info gain $IG(ECE146, X) = H(ECE146) - H(ECE146 \mid X)$ for each class.

i. $IG(ECE146, ECE102) = 0.9544 - 0.90565 = \underline{0.04875}$

ii. $IG(ECE146, ECE131) = 0.9544 - 0.40565 = \underline{0.54875}$

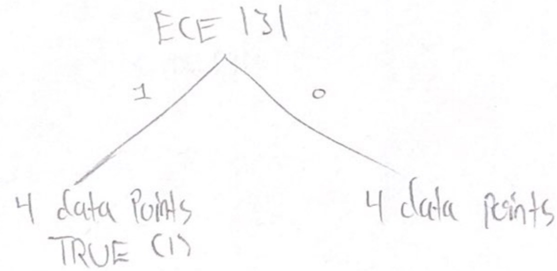
iii. $IG(ECE146, MATH61) = 0.9544 - 0.606844 = \underline{0.347556}$

iv. $IG(ECE146, MUSC15) = 0.9544 - 0.95120 = \underline{0.0032}$

$$\boxed{\begin{array}{l} IG(ECE146, ECE102) = 0.04875 \\ IG(ECE146, ECE131) = 0.54875 \\ IG(ECE146, MATH61) = 0.347556 \\ IG(ECE146, MUSC15) = 0.0032 \end{array}}$$

d) Based on info gain, best feature to split on is <u>ECE131</u>.

e) Initial Tree:



→ Noting that $H(ECE146 \mid ECE131 = 1) = 0$, we can stop splitting on that branch and label its leaf node as TRUE (1).

→ Now, need to split on branch 0 using the following data:

Sample	ECE102	MATH61	MUSC15	ECE146
5	1	0	1	0
6	0	0	0	0
7	1	1	1	1
8	0	0	1	0

By inspection, splitting on MATH61 will give us our max info gain. In fact, it will allow us to terminate splitting, as evidenced by the following computation showing $H(ECE146 \mid MATH61=1) = H(ECE146 \mid MATH61=0) = 0$.

$\hookrightarrow P(MATH61=0) = \frac{3}{4} \qquad P(MATH61=1) = \frac{1}{4}$

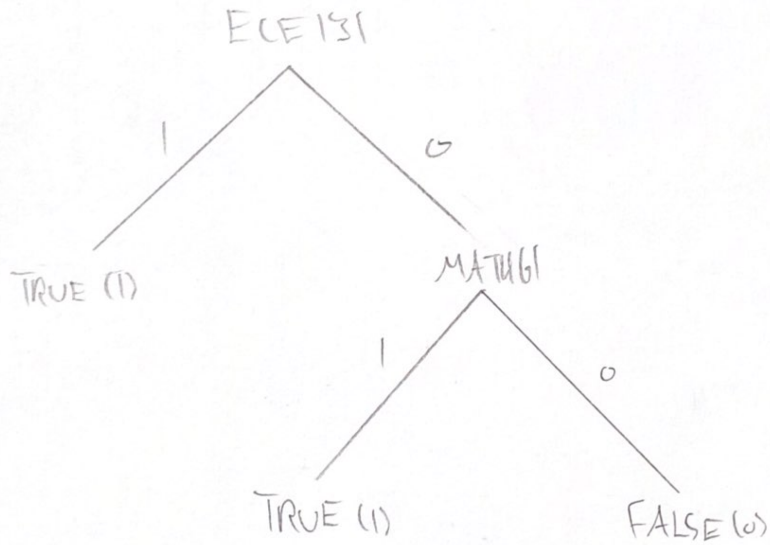
$P(ECE146=1 \mid MATH61=1) = 1 \qquad P(ECE146=0 \mid MATH61=1) = 0$

$P(ECE146=1 \mid MATH61=0) = 0 \qquad P(ECE146=0 \mid MATH61=0) = 1$

$\Rightarrow H(ECE146 \mid MATH61=0) = -1\log 1 - 0\log 0 = 0$

$H(ECE146 \mid MATH61=1) = -1\log 1 - 0\log 0 = 0$

Hence, the full decision tree is



f) Based on the decision tree in e, we know

- Student 9 is good at ECE146 (TRUE)
- Student 10 is bad at ECE146 (FALSE)

Question 4 – Linear Regression

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} 0 \\ 6 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 0 \end{bmatrix}$$

a) Y is a column vector of labels, $Y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \rightarrow Y = \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix}$

X is a matrix of data points, where each row is a data point. If we transform the data to a higher dimension to absorb the intercept term such that $x_i \rightarrow \begin{bmatrix} 1 \\ x_i \end{bmatrix}$ then $X = \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$

b) The optimal w that minimizes the objective function is given by the closed-form solution to linear regression, $w = (X^TX)^{-1}X^TY$.

- $X^T = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix}$

- $X^TX = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 3 & 5 \\ 5 & 13 \end{bmatrix}$ (invertible b/c determinant $\neq 0$)

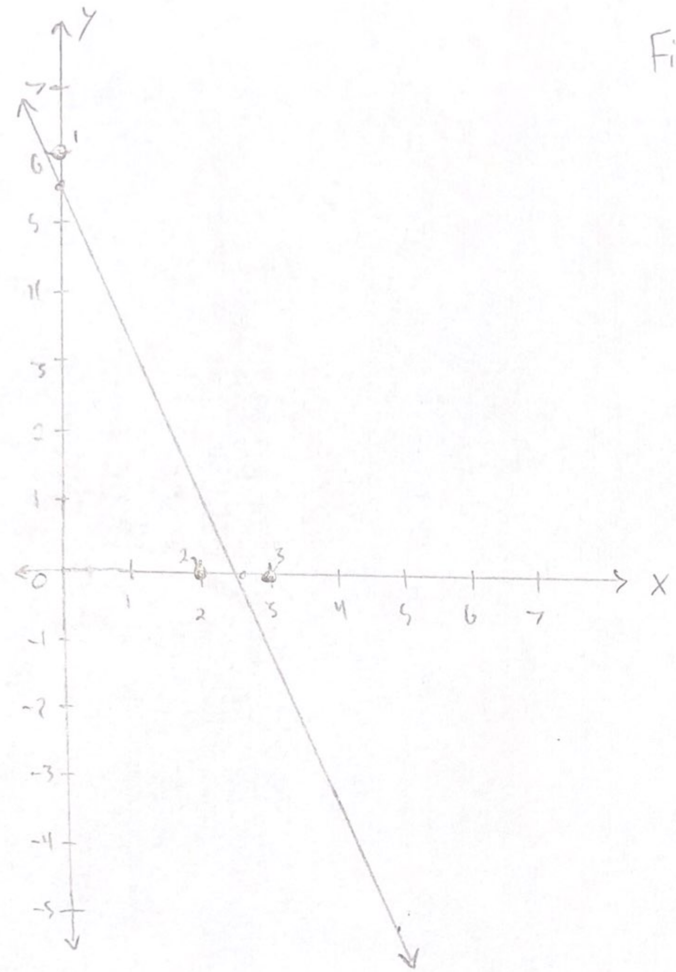
- $(X^TX)^{-1} = \dfrac{1}{3 \cdot 13 - 5 \cdot 5} \begin{bmatrix} 13 & -5 \\ -5 & 3 \end{bmatrix} = \dfrac{1}{14} \begin{bmatrix} 13 & -5 \\ -5 & 3 \end{bmatrix}$

- $w = (X^TX)^{-1}X^TY = \dfrac{1}{14} \begin{bmatrix} 13 & -5 \\ -5 & 3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & 3 \end{bmatrix} \begin{bmatrix} 6 \\ 0 \\ 0 \end{bmatrix} = \dfrac{1}{14} \begin{bmatrix} 13 & -5 \\ -5 & 3 \end{bmatrix} \begin{bmatrix} 6 \\ 0 \end{bmatrix}$

$= \dfrac{1}{14} \begin{bmatrix} 78 \\ -30 \end{bmatrix}$

$$\boxed{w = \frac{1}{14} \begin{bmatrix} 78 \\ -30 \end{bmatrix}}$$

c) Plot of data points and fitted line...



Fitted line: $y = w_1 x + w_0$

$$\updownarrow$$

$$y = -\frac{30}{14}x + \frac{78}{14}$$

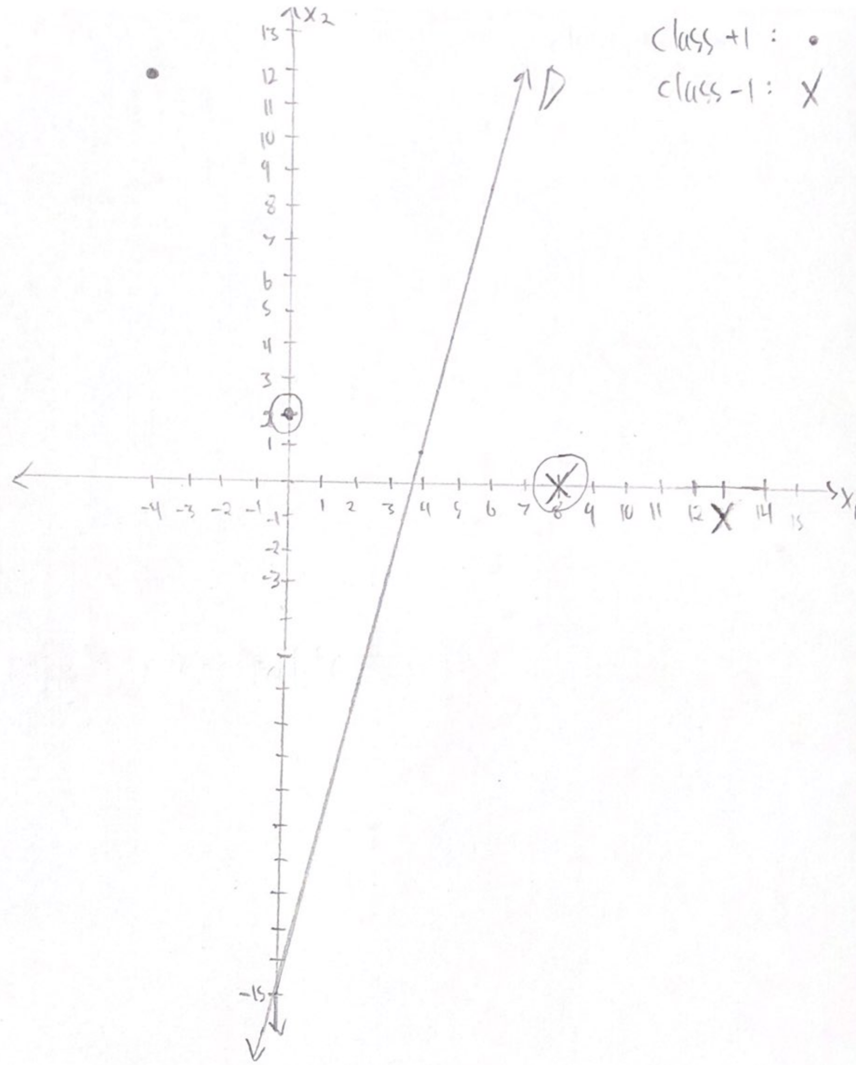
$$\updownarrow$$

$$y = -2.14x + 5.57$$

Question 5 – SVM

a) Plot of the data...

i	$x_1^{(i)}$	$x_2^{(i)}$	y_i
1	−4	12	1
2	0	2	1
3	8	0	−1
4	13	−1	−1



→ By inspection, the data is linearly separable.

b) The primal optimization problem is $\min_{w,b} \frac{1}{2}\|w\|^2$ s.t. $y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad i = 1, \ldots, m.$

Explicitly, this is:

$$\min_{[w_1,w_2]^T, b} \frac{1}{2}\|[w_1,w_2]^T\|^2$$

Subject to $(-4w_1 + 12w_2 + b) \geq 1$

$(0w_1 + 2w_2 + b) \geq 1$

$-(8w_1 + 0w_2 + b) \geq 1$

$-(13w_1 - w_2 + b) \geq 1$

c) Support vectors circled on plot, datapoints 2 and 3.
The maximum margin separating hyperplane satisfies
- intersects the midpoint of the line connecting the two support vectors.
- is perpendicular to the connecting line.

- Slope of connecting line between $X^2 = (0,2)$ and $X^3 = (8,0)$ is
$m = \frac{0-2}{8-0} = \frac{-2}{8} = \underline{-\frac{1}{4}}$. Slope of perpendicular line is $\underline{4}$.

- Midpoint of the support vectors given by $\left(\frac{0+8}{2}, \frac{2+0}{2}\right) = \left(\frac{8}{2}, \frac{2}{2}\right)$
$\Rightarrow$ $\underline{\text{midpoint} = (4,1)}$

- The hyperplane, D, is described by the line passing through $(4,1)$ with slope 4. Using $x_2 = mx_1 + b$...
$1 = 4(4) + b \Rightarrow b = 1 - 16 = -15 \Rightarrow x_2 = 4x_1 - 15$
$\Rightarrow$ $\boxed{D \text{ described by } x_2 = 4x_1 - 15}$

The hyperplane is plotted on the graph in a.

d) Dual form of Problem: $\max_a \left(\sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_i y_j \alpha_i \alpha_j x_i^T x_j \right) = \max_a \mathcal{L}$

subject to $\alpha_i \geq 0$ $(i=1,\dots,m)$, $\sum_{i=1}^{m} \alpha_i y_i = 0$.

Note that there are only 2 support vectors, so only 2 terms with $\alpha_i \neq 0$. Then, the problem can be rewritten as taking the max wrt a of
$\mathcal{L} = (\alpha_2 + \alpha_3) - \frac{1}{2}\left(\alpha_2^2 \|x_2\|^2 + \alpha_3^2\|x_3\|^2 - 2\alpha_2\alpha_3 x_2^T x_3\right)$.

Because $\sum_{n\in S} \alpha_n y_n = \alpha_2 y_2 + \alpha_3 y_3 = 0$ and $\alpha_2, \alpha_3 > 0$, we have $\alpha_2 = \alpha_3$.

Let $\alpha = \alpha_2 = \alpha_3$. Then, $\mathcal{L} = 2\alpha - \frac{1}{2}\left(\alpha^2\|x_2\|^2 + \alpha^2\|x_3\|^2 - 2\alpha^2 x_2^T x_3\right)$.

$\hookrightarrow$ $\|x_2\|^2 = 0^2 + 2^2 = 4$ $\qquad x_2^T x_3 = 0$

$\|x_3\|^2 = 8^2 + 0^2 = 64$

$\Longrightarrow \mathcal{L} = 2\alpha - \frac{1}{2}(\alpha^2 4 + \alpha^2 64) = 2\alpha - 34\alpha^2$.

Taking $\max_a \mathcal{L}$: $\frac{\partial \mathcal{L}}{\partial a} = 2 - 68\alpha = 0 \Longrightarrow \alpha = \frac{1}{34}$.

Noting that $\alpha = \frac{1}{34}$, we can find w and b.

$\hookrightarrow$ $w = \sum_{n\in S} \alpha_n y_n x_n = \left(\frac{1}{34}\right)(1)\begin{bmatrix}0\\2\end{bmatrix} + \left(\frac{1}{34}\right)(-1)\begin{bmatrix}8\\0\end{bmatrix} = \begin{bmatrix}-8/34\\2/34\end{bmatrix}$

$b = \frac{1}{|S|}\sum_{n\in S}\left[y_n - w^T x_n\right] = \frac{1}{2}\left[\left(1 - \begin{bmatrix}-\frac{8}{34} & \frac{2}{34}\end{bmatrix}\begin{bmatrix}0\\2\end{bmatrix}\right) + \left(-1 - \begin{bmatrix}-\frac{8}{34} & \frac{2}{34}\end{bmatrix}\begin{bmatrix}8\\0\end{bmatrix}\right)\right]$

$= \frac{1}{2}\left(1 - \frac{4}{34} - 1 + \frac{64}{34}\right) = \frac{15}{17}$

$$\boxed{\begin{gathered}\alpha_1 = \alpha_4 = 0 \\ \alpha_2 = \alpha_3 = 1/34 \\ w = \begin{bmatrix}-8/34\\2/34\end{bmatrix}, \quad b = \frac{15}{17}\end{gathered}}$$

Note that with these choices of w, b, D is described by

$-\frac{8}{34}x_1 + \frac{2}{34}x_2 + \frac{15}{17} = 0$

$\Updownarrow$

$-8x_1 + 2x_2 + 30 = 0$

$\Updownarrow$

$x_2 = 4x_1 - 15$ as predicted in c.