COSC 522 HOMEWORK 4

STUDENT NAME: JOSEPH OCHE AGADA

NET ID: JAGADA

PROBLEM 1

(1) I will present details of calculation here while implementation will be in the notebook attached to this submission. I will start from Minimum Euclidean Distance classifier (MD) to Support Vector Machine (SVM) and finally to perceptron.

MD

The steps involve in plotting the decision boundary for MD are:

- Calculate the centroids
- Determine the midpoint of the centroids
- Calculate the boundary line

I will calculate the centroids as follow:

AND gate has the following 4 points

1. (0, 0) with class label of 0
2. (0, 1) with class label of 0
3. (1, 0) with class label of 0
4. (1, 1) with class label of 1

Centroid for class 0

$$x\text{-coordinate} = \frac{0+0+1}{3} = \frac{1}{3}$$

$$y\text{-coordinate} = \frac{0+1+0}{3} = \frac{1}{3}$$

The centroid for class 0 is $(\frac{1}{3}, \frac{1}{3})$

Class 1 has only one point (1, 1) hence the centroid for class 1 is (1, 1)

Next step is to determine the midpoint. Midpoint for the 2 centroid is

$$M = \left(\frac{\frac{1}{3}+1}{2}, \frac{\frac{1}{3}+1}{2}\right) = \left(\frac{2}{3}, \frac{2}{3}\right)$$

The last step is to calculate the decision boundary line.

The decision boundary is a line perpendicular to the line joining the centroids. If the centroids are C_0 and C_1 then we can use the vector $(C_0 C_1)$ to determine the slope and direction of the decision boundary.

If the vector

$$\vec{d} = \left(1-\frac{1}{3}, 1-\frac{1}{3}\right) = \left(\frac{2}{3}, \frac{2}{3}\right)$$

Since the decision boundary is perpendicular to $\vec{d}$, and slope of $\vec{d} = 1$, then the slope of of the decision boundary is -1

Hence the equation of the decision boundary is

$$y - \frac{2}{3} = -1\left(x - \frac{2}{3}\right)$$

The decision boundary is

$$\boxed{y = -x + \frac{4}{3}}$$

The plot for the decision boundary is in the note book attached

SVM

To calculate the decision boundary of SVM, we determine the equation of the hyperplane that maximize the distance between the

hyperplane that maximizes the distance between the hyperplane and the closest Support Vector.

As already known,

for AND gate

class 0 has points (0,0), (0,1), (1,0)

class 1 has point (1,1)

The objective is to find a line that maximizes the distance between the line and the closest points (Support Vectors). Let the line be

$$W_1x_1 + W_2x_2 + b = 0$$

Where $W = [W_1, W_2]$ is the weight and b is the bias of the hyperplane. For SVM, the classes are labeled as -1 and 1 hence we set the constraint as follow to ensure that each datapoint is on the correct side of the boundary. the constraint are:

for class 0 $Wx_i + b \leq -1$

for class 1 $Wx_i + b \geq 1$

Since the samples are linearly seperable, the objective is to minimize $\|W\|^2$ subject to

$$y_i(Wx_i + b) \geq 1 \quad \forall i$$

Where y_i is the class label for each sample ($y_i = -1$ for class 0 and 1 for class 1)

We solve this optimization problem using Langrangien for optimization as follow:

- identify the support vectors
- calculate W and b based on the constraint

Suppose we have a solution $w = [1, 1]$, we solve for b point (1, 1) with label +1

$$wx + b \geq 1 \Rightarrow (1)(1) + (1)(1) + b \geq 1$$
$$\Rightarrow b \geq -1$$

Support vector machine decision boundary in this case becomes

$$x_1 + x_2 - 1 = 0$$

with this boundary, the margin boundaries are

$x_1 + x_2 = 0$ for class 0 (-1 side of the boundary)

$x_1 + x_2 = 2$ for class 1 (+1 side of the boundary)

These are parallel lines equidistant from the decision boundary illustrating the maximum boundaries between classes.

This is the explanation for SVM. Implementation is in the notebook attached.

Perceptron

Given some optimal weights W_1, W_2, W_0, the decision boundary for perception is gotten by setting

$$W_1 x_1 + W_2 x_2 + W_0 = 0$$

To calculate the decision boundary, I will determine the optimal weights for the AND gate using online learning with initial weights of

$W_1 = 0.1$

$W_2 = 0.2$

$W_0 = 0.5$

and learning rate

$n = 1$

Other initialization parameters may give a different decision boundary. this is reason why decision boundary for perceptron is not unique.

Epoch 1

Sample 1 : (0,0) target = 1

$Z = W_1x_1 + W_2x_2 + W_0 = 0.1\times0 + 0.2\times0 + 0.5 = 0.5$

$Z = 0.5 > 0$ hence $\hat{y} = 1$

but $y = 0$, therefore we update weight as follow:

$$\begin{pmatrix} W_1 \\ W_2 \\ W_0 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.5 \end{pmatrix} + 1(0-1)\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.2 \\ -0.5 \end{pmatrix}$$

Sample 2: (0, 1); targe = 0

$Z = W_1x_1 + W_2x_2 + Z = 0.1\times0 + 0.2\times1 - 0.5 = -0.3$

$Z = -0.3 < 0$ hence $\hat{y} = 0$ and $y = 0$

therefore, no weight update

Sample 3: (1,0) target = 0

$Z = W_1x_1 + W_2x_2 + W_0 = 0.1\times1 + 0.2\times0 - 0.5 = -0.4$

$Z = -0.4 < 0$ hence $\hat{y} = 0$ and $y = 0$, therefore no weight update

Sample 4 (1, 1) target = 0

$Z = W_1x_1 + W_2x_2 + W_0 = 0.1\times1 + 0.2\times1 - 0.5 = -0.2$

$Z = -0.2 < 0$ hence $\hat{y} = 0$ and $y = 1$ hence we update weight as follow:

$$\begin{pmatrix} W_1 \\ W_2 \\ W_0 \end{pmatrix} = \begin{pmatrix} 0.1 \\ 0.2 \\ -0.5 \end{pmatrix} + (1-0)\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 1.1 \\ 1.2 \\ 0.5 \end{pmatrix}$$

Epoch 2 will not result in any weight change hence final weights are $W_1 = 1.1$, $W_2 = 1.2$ $W_3 = 0.5$

Hence the decision boundary is

$1.1x_1 + 1.2x_2 + 0.5 = 0$

Reason Why Decision Boundaries for perceptron are Not Unique

Decision boundary for perceptron is not unique for the following reasons

- Dependency on Initial Weights and Order of data presentation — the final weight of a perceptron depend on the initial weights and for different set of weights, there is a different decision boundary. Hence decision boundary largely depend on Initialization strategy/initial weights used.

- Non-Unique Solution in Linearly Seperable Data When sample is linearly seperable, there are infinitly many possible hyperplanes that seperate the classes. the perceptron algorithm stop as soon as it finds a solution, but other hyperplanes could also correctly classify the sample. Thus, depending on the Initialization strategy/Weights and the specific order in which data points are presented, perceptron could converge to different solutions.

- Non Convex Solution Space - For perceptron, the optimization problem is non-convex, which means there are multiple solution that could satisfy some objective. Since perceptron algorithm does not seek to minimize an objective

function globally but only locally, it can find any feasible solution leading to non-unique boundaries (7)

PROBLEM 1

(2) Here, I calculated the projections for FLD and PCA, and applied minimum euclidean to the projections to arrive a new set of decision boundaries for FLD + MD and PCA + MD (Recall that MD = Minimum Euclidean Distance).

Details of the calculations are as follow:

for class 0

$$\mu_0 = \left(\frac{0+0+0}{3}, \frac{0+1+0}{3}\right) = \left(\frac{1}{3}, \frac{1}{3}\right)$$

for class 1

$$\mu_1 = (1, 1)$$

overall mean

$$\mu = \left(\frac{0+0+1+1}{4}, \frac{0+1+0+1}{4}\right) = \left(\frac{1}{2}, \frac{1}{2}\right)$$

Within class scatter matrix (S_W)

for class 0

$$S_W^{(0)} = \sum_{i=1}^{3} (x_i - \mu_0)(x_i - \mu_0)^T$$

for class 1

$$S_W^{(1)} = \sum_{i=1}^{3} (x_i - \mu_1)(x_i - \mu_1)^T$$

Between class scatter matrix

$$S_B = n_0 (\mu_0 - \mu)(\mu_0 - \mu)^T + n_1 (\mu_1 - \mu)(\mu_1 - \mu)^T$$

Next step is eigen decomposition by solving for the eigen values and eigen vectors using $S_W^{-1} S_B$ to find the direction that maximize separation (For FLD). The eigen vector corresponding to the highest eigen value becomes the ~~weight~~ projection weight (W_{FLD})

PCA Projection

For PCA projection, we define

$$X_{centered} = X - \mu$$

We compute the covariance matrix

$$Cov(X) = \frac{1}{n-1} X_{centered}^{T} X_{centered}$$

Next step is to calculate the eigen values and eigen vectors of $Cov(X)$ so that the eigen vector corresponding to the the highest eigen value, become the projection weights for PCA (W_{PCA})

Projections For FLD and PCA:

$$X_{FLD} = X \cdot W_{FLD}$$

$$X_{PCA} = X \cdot W_{PCA}$$

The projection was implemented in the A notebook attached.

PROBLEM 2

a. Causes of High precision Low Recall

First, precision = $\frac{TP}{(TP+FN)}$ Recall = $TP/(TP+FN)$

Here, high precision itself is not a problem but it becomes a problem if it is at the expense of Low precision because it implies that the model is missing or failing to identify many positive classes. possible causes are

→ Imbalanced Training Sample - When training Sample is imbalanced with the positive class being the minority class, recall is expected to ~~beb~~ be low because the model, even though may perform well generally, it will tend to misclassify the positive class leading to low recall.

→ classification threshhold — the model may be using a high ~~treh~~ threshold for classifying instance as positive which reduces false positive, hence high precision but also leads to missing many ~~positive~~ actual positive cases implying low recall.

→ overfitting — the model might have been over fine tune to perform well on positive class leading to predicting positive only when it is very ~~sure thereby~~ Certain thereby missing many actual positive implying low recall.

b. Low precision and High Recall

Low precision means the model has high rate of false positive meaning many samples predicted positive are actually negetive, while high recall indicates that it identifies most of the positive cases correctly. possible causes are:

→ Lax classification Threshhold – The model might employ a low threshhold for classifying instances as positive leading to too many prediction being marked as positive which increase false positive and lower precision.

→ The goal may be to prioritize identifying the positive class (for example in the case of medical diagnosis), this will lead to high recall while precision will be intentionally sacrificed.

→ Complex or Noisy Data – if the data set is noisy or have overlapping classes, the model may find ~~the~~ it easier to classify a larger number of instances as positive.

PROBLEM 3

Hrs (x_k)	1·0	2·0	3·0	4·0	5·0
Prob (P_k)	0·07	0·26	0·61	0·87	0·97
Pass (y_k)	0	0	0	1	1

Q. To build a model that will predict probability of pass, I will model the relationship between x_k and P_k using linear basis function where P_k is treated as the dependent variable while x_k is the independent variable. the general regression equation is

$$P_k(w, x_k) = \sum_{j=0}^{M-1} w_j \phi_j(x_k)$$

where $\phi_j(x_k)$ is a linear basis function, for this particular problem, the regression equation can be reduced to

$$P_k = w_0 + w_1 x_k$$

where w_0 and w_1 can be denoted to be

$$\hat{w}_1 = \frac{\sum_k (x_k - \bar{x})(P_k - \bar{P})}{\sum (x_k - \bar{x})^2}$$

$$\hat{w}_0 = \bar{P} - w_1 \bar{x}$$

$\bar{x} = \frac{1+2+3+4+5}{5} = 3$

$\bar{P} = \frac{0.07+0.26+0.61+0.87+0.97}{5} = 0.556$

To calculate w_1, we first calculate $(x_k - \bar{x})$ and $(P_k - \bar{P})$ for each pair

for $x_k = 1$ $P_k = 0.07$ (ie $k = 1$)

$(x_1 - \bar{x}) = (1-3) = -2$

$(P_1 - \bar{P}) = (0.07 - 0.556) = -0.486$

for $k = 2$

$(x_2 - \bar{x}) = (2-3) = -1$

$(P_2 - \bar{P}) = (0.26 - 0.556) = -0.296$

for $k = 3$

$(x_3 - \bar{x}) = (3-3) = 0$

$(P_3 - \bar{P}) = (0.61 - 0.556) = 0.054$

for $k = 4$

$(x_4 - \bar{x}) = (4-3) = 1$

$(P_4 - \bar{P}) = (0.87 - 0.556) = 0.314$

for $k = 5$

$(x_5 - \bar{x}) = (5-3) = 2$

$(P_5 - \bar{P}) = (0.97 - 0.556) = 0.414$

Hence $\sum_k (x_k - \bar{x})(P_k - \bar{P}) = (-2)(-0.486) + (-1)(-0.296)$

$(0 \times 0.054) + (1)(0.314) + (2)(0.414)$

$= 0.972 + 0.296 + 0 + 0.314 + 0.828$

$= 2.41$

hence $\sum_K (x_K - \bar{x})^2 = (-2)^2 + (-1)^2 + (0)^2 + 1^2 + 2^2$

$= 1^2 + 2^2 + 1^2 + 2^2 = 10$

Hence

$$W_1 = \frac{2.41}{10} = \underline{0.241}$$

$$W_0 = \bar{P} - W_1 \bar{x} = 0.556 - 0.241 \times 3 = -0.167$$

the linear regression model becomes

$$\boxed{P_K = -0.167 + 0.241 x_K}$$

When $x = 3.27$

$$P_K = -0.167 + 0.241(3.27)$$

$$P_K = \underline{0.62}$$

b. The general regression equation is

$$y(W, x) = \sum_{j=0}^{M-1} W_j \phi_j(x)$$

In this particular case

$\phi_j(x)$ is a sigmoid basis function given as

$$\sigma(x_K) = \frac{1}{1 + e^{-x_K}}$$

This regression problem is equivalent to

$$P(y_K | x_K) = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 x_K))}$$

Where β_0 is intercept and

β_1 is slope

β_0 and β_1 are estimated using a combination of maximum likelihood estimation and iterative approach such as Newton-Raphson as

$\hat{\beta}_0 = -3.748$

$\beta_1 = 1.047$

Hence

$$P(y_k|x_k) = \frac{1}{1+\exp[-(-3.748)+1.047x_k]}$$

A student who studied for 3.25 hours has the following estimated probability of passing

$$P(y|x_k=3.25) = \frac{1}{1+\exp[-(-3.748)+1.047(3.25)]}$$

$$= 0.415$$

A student who studies for 3.25 has probability of passing as 0.415