Harshwardhan
COSC 522
HW-5

Problem 1

(a)

CL

GT

L_1	ω_1	ω_2	ω_3
ω_1	0.66	0.17	0.17
ω_2	0.1	0.8	0.1
ω_3	0	0.3	0.7

CL

GT

L_2	ω_1	ω_2	ω_3
ω_1	0.83	0.07	0.1
ω_2	0.1	0.73	0.17
ω_3	0.17	0.2	0.63

Fused

	$\omega_1\omega_1$	$\omega_1\omega_2$	$\omega_1\omega_3$	$\omega_2\omega_1$	$\omega_2\omega_2$	$\omega_2\omega_3$	$\omega_3\omega_1$	$\omega_3\omega_2$	$\omega_3\omega_3$
ω_1	0.55	0.05	0.07	0.14	0.01	0.02	0.14	0.01	0.02
ω_2	0.01	0.07	0.02	0.08	0.58	0.14	0.01	0.07	0.02
ω_3	0	0	0	0.05	0.06	0.19	0.12	0.14	0.44

L_1, L_2	Fused label
1, 1	1
1, 2	2
1, 3	1
2, 1	1
2, 2	2
2, 3	3
3, 1	1
3, 2	3
3, 3	3

These labels were found using the the probability table above. The highest probability option gets the sample point classification.

(b)

It is not possible to generate lookup table using only the confusion matrices presented above. To use Behaviour-Knowledge-Space (BKS) method, we need the classifier labels as well as samples in the validation (or training) set. The way BKS method works is that it calculates the number of samples from each class for every classifier-label combination.

This is impossible unless we have the validation set (features and labels) and their predicted labels using our algorithm. We need to have a validation set and its labels to know the sample distribution (and then choose based on majority) for each predicted-class combination — which we don't have.

Naive-Bayesian (NB) algorithm assumes that each classifier is an independent machine; thus, predicted labels from each machine is independent from the other machine. This assumption is rarely true in real life as all the classifiers are trained on the same data and trying to measure the same activities and phenomenon.

BKS does not assume this independence of classification machines. By using validation set and the predicted labels, and counting the labels belonging to each category, they avoid the questionable assumption.

Problem 2

* Gini impurity: $i(N) = 1 - \sum_i P^2(\omega_i)$
* $\Delta i(N) = i(N) - P_L\, i(N_L) - (1-P_L)\, i(N_R)$

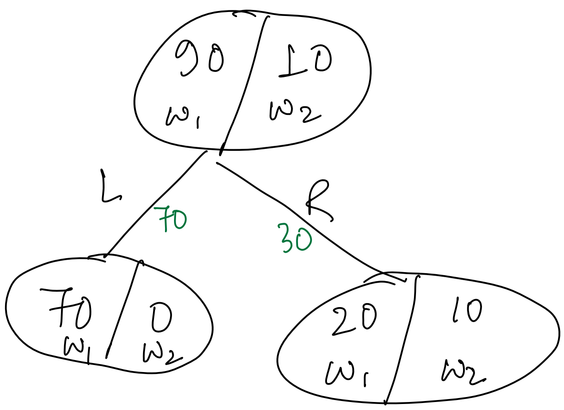


Fig Option 1

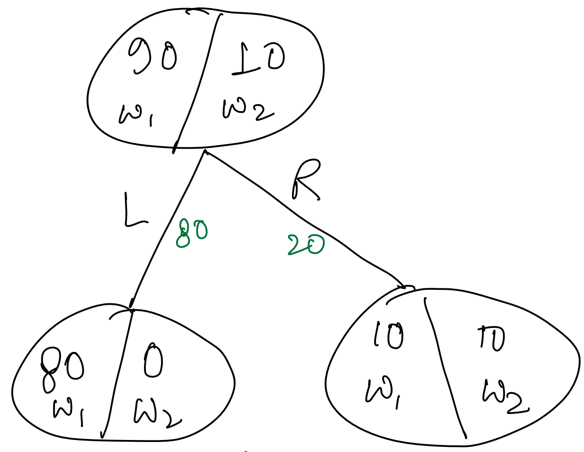


Fig Option 2

$$i(N) = 1 - \left[\left(\frac{90}{100}\right)^2 + \left(\frac{10}{100}\right)^2\right]$$

$$= 1 - (0.81 + 0.01) = 0.18$$

Option 1

$$i(N_L) = 1 - \left[\left(\frac{70}{70}\right)^2 + \left(\frac{0}{70}\right)^2\right] = 0$$

$$i(N_R) = 1 - \left[\left(\frac{20}{30}\right)^2 + \left(\frac{10}{30}\right)^2\right] = 0.44$$

$$\Delta i = 0.18 - \left[\frac{70}{100} \times 0 + \frac{30}{100} \times 0.44\right]$$

$$= 0.18 - 0.132$$

$$= 0.048$$

Option 2

$$i(N_L) = 1 - \left[\left(\frac{80}{80}\right)^2 + \left(\frac{0}{80}\right)^2\right] = 0$$

$$i(N_R) = 1 - \left[\left(\frac{10}{20}\right)^2 + \left(\frac{10}{20}\right)^2\right] = 1 - [0.25 + 0.25]$$

$$= 0.5$$

$$\Delta i = 0.18 - \left[\frac{80}{100} \times 0 + \frac{20}{100} \times 0.5\right]$$

$$= 0.18 - 0.1$$

$$= 0.08$$

Because the drop in impurity is highest for option 2, I will choose option 2 over option 1.

Problem 3

The four training samples are [[-1, -1], [-1, 1], [1, - 1], [1, 1]], with the corresponding label being [-1, 1, 1, -1].

(a) Suppose the kernel function is a 2nd-degree polynomial, i.e., K(x, y) = (x1y1 + x2y2 + C)2, where x = [x1 x2]T, y = [y1 y2]T. Derive the basis functions f(x), that is, K(x, y) = f(x)'.f(y). (b) (10 pts) Using the derived basis function, what is the higher-dimensional space that the 2- d sample should be mapped to? Provide the higher-dimensional counterpart to the four 2- d samples.

$$K(x,y) = f(x)' f(y)$$
$$= (x'y + c)^2$$

$$(x_1y_1 + x_2y_2 + c)^2 = x_1^2y_1^2 + x_2^2y_2^2 + c^2 + 2x_1x_2y_1y_2 + 2cx_1y_1 + 2cx_2y_2$$

$$\begin{pmatrix} x_2^2 \\ x_1^2 \\ \sqrt{2}x_2x_1 \\ \sqrt{2c}\,x_2 \\ \sqrt{2c}\,x_1 \\ c \end{pmatrix}' \begin{pmatrix} y_2^2 \\ y_1^2 \\ \sqrt{2}y_2y_1 \\ \sqrt{2c}\,y_2 \\ \sqrt{2c}\,y_1 \\ c \end{pmatrix}$$

$$= x_2^2y_2^2 + x_1^2y_1^2 + 2x_1x_2y_1y_2 + 2cx_2y_2 + 2cx_1y_1 + c^2.$$

Thus, the kernal fn's basis is:

$$\phi(x_1, x_2) = \begin{pmatrix} x_2^2 \\ x_1^2 \\ \sqrt{2}\, x_2 x_1 \\ \sqrt{2c}\, x_2 \\ \sqrt{2c}\, x_1 \\ c \end{pmatrix}$$

(b) Using the derived basis function, what is the higher-dimensional space that the 2- d sample should be mapped to? Provide the higher-dimensional counterpart to the four 2- d samples.

The four samples are:

X_1	X_2	Y
−1	−1	−1
−1	1	1
1	−1	1
1	1	−1

$\phi(x_1, x_2) = \langle 1, 1, \sqrt{2}, -\sqrt{2}c, -\sqrt{2}c, c \rangle$
$(-1,-1)$

$\phi(-1,1) = \langle 1, 1, -\sqrt{2}, \sqrt{2c}, -\sqrt{2c}, c \rangle$

$\phi(1,-1) = \langle 1, 1, -\sqrt{2}, -\sqrt{2c}, \sqrt{2c}, c \rangle$

$\phi(1,1) = \langle 1, 1, \sqrt{2}, \sqrt{2c}, \sqrt{2c}, c \rangle$

They have been mapped from 2-d space to 6-d space.

(c) Apply Perceptron on the higher-dimensional samples and see if you can find a linear decision boundary using the higher-dimensional samples. Output the weights learned. Project the learned hyperplane onto 2-D space and show the decision boundary.

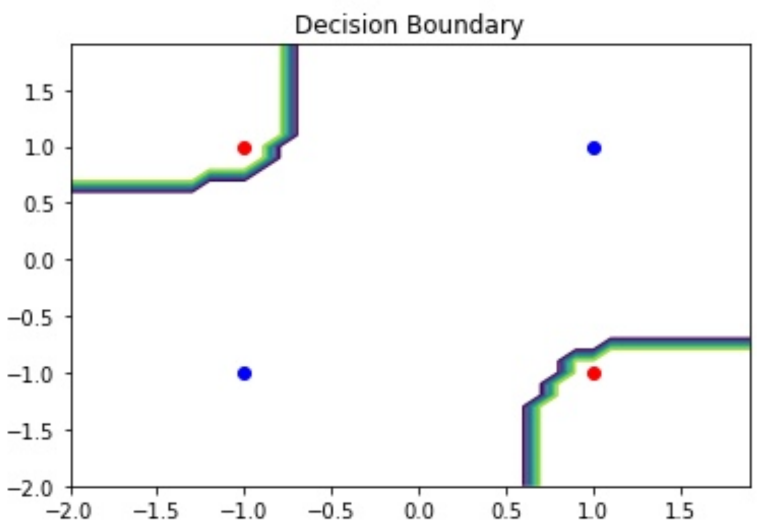
Using the higher dimensional samples, the perceptron algorithm converged in 9 epochs. We did obtain a linear decision boundary: a hyperplane. The weights are the following.

w = <-13.59, -13.92, -42.41, 0.29, 0.017, -13.04>

The bias is 14.69.

The decision boundary equation is:
-13.59x1 - 13.92x2 - 42.41x3 + 0.29x4 + 0.017x5 - 13.04x6 = 14.69

Here is the projected contour plot for the hyperplane. The Python codes to generate it are appended with this submission.



(d) What would be the support vectors in the higher-dimensional space? Manually find the decision boundary if using SVM. Provide parameters that determine the decision boundary. Project the decision boundary onto 2-D space.

From previous questions, we know that the basis function for the kernel is the following.

$$\phi(x) = \begin{pmatrix} 1 \\ x_1^2 \\ \sqrt{2}\, x_1 x_2 \\ x_2^2 \\ \sqrt{2}\, x_1 \\ \sqrt{2}\, x_2 \end{pmatrix} \qquad [\text{assuming } c = 1]$$

It can be verified that

$$K(x_1, x_2) = \phi(x_1)' \phi(x_2)$$

Using this and the values for XOR,

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

Thus the objective function is:

$$= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i x_j$$

$$= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} \left(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3 \right.$$

$+ 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4$
$+ 9\alpha_4^2)$

Optimising this with respect to lagrange multipliers $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ gives the following:

$$9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \quad - (1)$$
$$-\alpha_1 + 9\alpha_2 + \alpha_3 - \alpha_4 = 1 \quad - (2)$$
$$-\alpha_1 + \alpha_2 + 9\alpha_3 - \alpha_4 = 1 \quad - (3)$$
$$\alpha_1 - \alpha_2 - \alpha_3 + 9\alpha_4 = 1 \quad - (4)$$

This is a system of linear equations. with four equations and four unknowns.

Eq^n (1) + (2) gives

$$8(\alpha_1 + \alpha_2) = 2$$
$$\Rightarrow \alpha_1 = \frac{1}{4} - \alpha_2 \quad - (5)$$

Eq^{n}s (1) − (4) gives:

$$8\alpha_1 - 8\alpha_4 = 0$$
$$\Rightarrow \alpha_1 = \alpha_4 \rightarrow (6) \quad \Rightarrow \alpha_4 = \frac{1}{4} - \alpha_2 \quad - (6)$$

Eq^{n}s (2) + (4) gives:

$$8\alpha_2 + 8\alpha_4 = 2$$

$$8\alpha_2 + 8\left(\frac{1}{4} - \alpha_2\right) = 2$$

$$\Rightarrow 8\alpha_2 + 2 - 8\alpha_2 = 2$$

Eq^{n} (2) − (3) gives:

$$8\alpha_2 - 8\alpha_3 = 0$$

$$\Rightarrow \alpha_2 = \alpha_3 . \quad \text{———} \quad (7)$$

Putting (5), (6) & (7) in (1) we get,

$$9\alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1$$

$$\Rightarrow 9\alpha_1 - \left(\frac{1}{4} - \alpha_1\right) - \left(\frac{1}{4} - \alpha_1\right) + \alpha_1 = 1$$

$$\Rightarrow 10\alpha_1 + \alpha_1 + \alpha_1 = 1 + \frac{1}{4} + \frac{1}{4}$$

$$\Rightarrow 4\,\cancel{12}\,\alpha_1 = \frac{\cancel{3}}{2} \qquad \Rightarrow \alpha_1 = \frac{1}{8} .$$

$$\alpha_2 = \frac{1}{4} - \alpha_1 = \frac{1}{8}$$

$$\alpha_3 = \alpha_2 = \frac{1}{8}$$

$$\alpha_4 = \alpha_1 = \frac{1}{8}$$

Thus,

$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = \frac{1}{8}.$$

Thus, all the four points of the XOR input vector are support vectors.

$Q(\alpha)$

$$= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\Big(9\alpha_1^2 - 2\alpha_1\alpha_2 - 2\alpha_1\alpha_3$$
$$+ 2\alpha_1\alpha_4 + 9\alpha_2^2 + 2\alpha_2\alpha_3 - 2\alpha_2\alpha_4 + 9\alpha_3^2 - 2\alpha_3\alpha_4$$
$$+ 9\alpha_4^2\Big)$$

$$= \frac{1}{4}.$$

For the weight vector:

$$\frac{1}{2}\|w_0\|_2^2 = \frac{1}{4}$$

$$\Rightarrow \|w_0\|_2 = \frac{1}{\sqrt{2}}.$$

The optimum weight vector is the following:

$$= -\alpha_1 x_1 + \alpha_2 x_2 + \alpha_3 x_3 - \alpha_4 x_4$$

$$= \frac{1}{8}\left[-\begin{pmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ -\sqrt{2} \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \end{pmatrix} - \begin{pmatrix} 1 \\ 1 \\ \sqrt{2} \\ 1 \\ \sqrt{2} \\ \sqrt{2} \end{pmatrix} \right]$$

$$= \frac{1}{8}\begin{bmatrix} 0 \\ 0 \\ -4\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

The first element of $\underset{\sim}{w}$ tells us that the bias is zero.

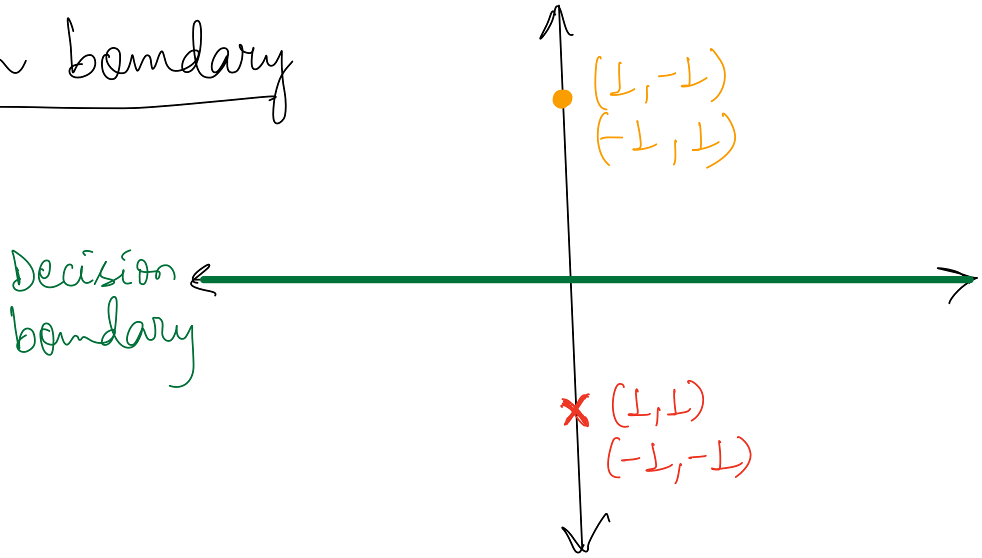
Thus, the optimal hyperplane separating the XOR points is:

$$w_0^T \phi(x) = 0$$

$$\Rightarrow \begin{bmatrix} 0 \\ 0 \\ -1/\sqrt{2} \\ 0 \\ 0 \\ 0 \end{bmatrix}^T \begin{bmatrix} 1 \\ x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \\ \sqrt{2}x_1 \\ \sqrt{2}x_2 \end{bmatrix} = 0$$

$$\Rightarrow -x_1x_2 = 0 \Rightarrow x_1x_2 = 0$$

Decision boundary



Support Vector Machine Table

X_1	X_2	Y
-1	-1	-1
1	1	-1
-1	1	1
1	-1	1

The decision boundary has to pass through the midpoint of the distance between the margins. Thus, the green line is the decision boundary.