For XOR data set

α_1	α_2	Y	
0	0	0	-1
0	1	1	1
1	0	1	1
1	1	0	-1

Here we are using the Kernel function

for $x = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$ $x' = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$

$$K(x, x') = \left(x^T x' + 1 \right)^2$$

$$\left(v_1 v_1 + v_2 v_2 \right) + 1 \big)^2$$

$$= 1 + (v_1 v_1 + 2 v_2 v_2)^2 + 2(v_1 v_1 + v_2 v_2)$$

$$= 1 + v_1^2 v_1^2 + v_2^2 v_2^2 + 2 v_1 v_1 v_2 v_2 + 2 v_1 v_1 + 2 v_2 v_2)$$

so the equivalent transformation

$$x = \left[1, \sqrt{2} x_1, \sqrt{2} x_2, x_1^2, x_2^2, \sqrt{2} x_1 x_2 \right]^T$$

for $(x_1, x_2) = (0,0)$

$$\phi \cdot [0,0] = \left[1, 0, 0, 0, 0, 0 \right]^T$$

$$\phi \cdot [0,1] = \left[1, 0, \sqrt{2}, 0, 1, 0 \right]^T$$

$$\phi \cdot [1,0] = \left[1, \sqrt{2}, 0, 1, 0, 0 \right]^T$$

$$\phi \cdot [1,1] = \left[1, \sqrt{2}, \sqrt{2}, 1, 1, \sqrt{2} \right]^T$$

$$L(\alpha) = \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j K_{ij}$$

$$S.T \quad \alpha_i y_i = 0 \Rightarrow \alpha_1 + \alpha_4 = \alpha_2 + \alpha_3$$

$$\alpha_i \geq 0$$

inner product = K $\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 4 \\ 1 & 1 & 4 & 4 \\ 1 & 4 & 4 & 89 \end{pmatrix}$

Our lagrange equation:

$$q_1 + q_2 + q_3 + q_4 - \frac{1}{2}\Big(q_1^2 + 4q_2^2 + 4q_3^2 + 9q_4^2 +$$
$$- q_1 q_2 + q_1 q_3 + q_1 q_4$$
$$- q_1 q_2 - q_1 q_3 + q_1 q_4$$
$$+ q_2 q_3 + q_2 q_3 - 4q_2 q_4 - 4q_4 q_2 \quad -4q_3 q_4 - 4 q_4 q_3 \Big)$$

$$= q_1 + q_2 + q_3 + q_4 + \frac{q_1^2}{2} + \frac{2 q_2^2}{2} + 2q_3^2 + 9q_4^2 - q_1 q_2$$
$$- q_1 q_3 + q_1 q_4 + q_2 q_3 \quad + \frac{4}{2} q_2 q_4 - 4 q_3 q_4$$

$$\frac{\partial L}{\partial q_1} = 0 \Rightarrow q_1 - q_2 + q_3 + q_4 = 1 \quad (1)$$

$$\frac{\partial L}{\partial q_2} = 0 \Rightarrow 4q_2 - q_1 + q_3 - 4q_4 = 1 \quad (11)$$

$$\frac{\partial L}{\partial q_3} = 0 \Rightarrow 4q_3 - q_1 + q_2 - 4q_4 = 1 \quad (111)$$

$$\frac{\partial L}{\partial q_4} = 0 \Rightarrow 9 q_4 + q_1 - 4q_2 - 4q_3 = 1 \quad (1V)$$

from the following equations

$$q_1 = \frac{13}{3} \qquad q_2 = \frac{8}{3} \qquad q_3 = \frac{8}{3} \qquad q_4 = 2$$

as we know

$$\omega = \sum_{i=1}^{4} \alpha_i y_i \, \phi(x_i)$$

$$-\frac{13}{3}\begin{bmatrix}①1\\0\\0\\0\\0\end{bmatrix}+\frac{8}{3}\begin{bmatrix}1\\0\\\sqrt{2}\\⓪\\1\\0\end{bmatrix}+\frac{8}{3}\begin{bmatrix}1\\\sqrt{2}\\0\\1\\0\\0\end{bmatrix}-2\begin{bmatrix}1\\\sqrt{2}\\\sqrt{2}\\1\\1\\\sqrt{2}\end{bmatrix}$$

$$=\begin{bmatrix}-1\\\dfrac{2\sqrt{2}}{3}\\\dfrac{2\sqrt{2}}{3}\\-2\sqrt{2}\\\dfrac{2}{3}\\\dfrac{2}{3}\end{bmatrix}$$

$$g(x, cc) = \omega^T \Phi(x)$$

$$=\begin{bmatrix}-1 & \dfrac{2\sqrt{2}}{3} & \dfrac{2\sqrt{2}}{3} & \dfrac{2\sqrt{2}}{3} & \dfrac{2}{3} & \dfrac{2}{3} & -2\sqrt{2}\end{bmatrix}\begin{bmatrix}1\\\sqrt{2}x_1\\\sqrt{2}x_2\\x_1^2\\x_2^2\\\sqrt{2}x_1 x_2\end{bmatrix}$$

• decision boundary

$$= -1 + \frac{4}{3}x_1 + \frac{4}{3}x_2 + \frac{2}{3}x_1^2 + \frac{2}{3}x_2^2 - 4x_1 x_2 \geq 0$$

Q. Bayes classifiers means classification of datapoints into 2 or more classes based on their probability of occurence of their attributes in Class, or Class 2.

Q3   Let N be equal to the total number of data points and N_1 be the data points for ~~test~~ Class 1 and N_2 for ~~test~~ Class 2

We label t as $\frac{N}{N_1}$ for C_1 and $-\frac{N}{N_2}$ for C_2

Then the sum of square error function

$$E = \frac{1}{2} \sum_{n=1}^{N} \left(\omega^T x_n + \omega_0 - t_n \right)^2$$

$$\frac{\partial E}{\partial \omega} = 0 \Rightarrow \sum_{n=1}^{N} \left(\omega^T x_n + \omega_0 - t_n \right) x_n = 0 \quad (1)$$

$$\frac{\partial E}{\partial \omega_0} = 0 \Rightarrow \sum_{n=1}^{N} \left(\omega^T x_n + \omega_0 - t_n \right) = 0 \quad (11)$$

from 11 we can compute

$$\omega_0 = -\omega^T \bar{m}$$

here $\bar{m} = \frac{1}{N} \sum_{n=1}^{N} x_n$

ie $\bar{m} = \frac{1}{N} \left(N_1 \bar{m}_1 + N_2 \bar{m}_2 \right)$

Here $\bar{m}_1$ and $\bar{m}_2$ are corresponding mean of dataset belong to class C_1 and C_2

and we can see

$$\sum_{n=1}^{N} t_n = N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2} = 0$$

equation (d) can also be written as

$$\left(S_W + \frac{N_1 N_2}{N} S_B\right) \omega = N(\bar{m}_1 - \bar{m}_2)$$

Here $S_W = \sum_{n \in C_1} (x_n - \bar{m}_1)(x_n - \bar{m}_1)^T$

$+ \sum_{n \in C_2} (x_n - \bar{m}_2)(x_n - m_2)^T$

$$S_B = (\bar{m}_2 - \bar{m}_1)(\bar{m}_2 - \bar{m}_1)^T$$

To get the direction of ω

we can see $S_B \omega$ is always in the direction of $(\bar{m}_2 - \bar{m}_1)$

$$\omega \propto S_W^{-1}(\bar{m}_2 - \bar{m}_1)$$

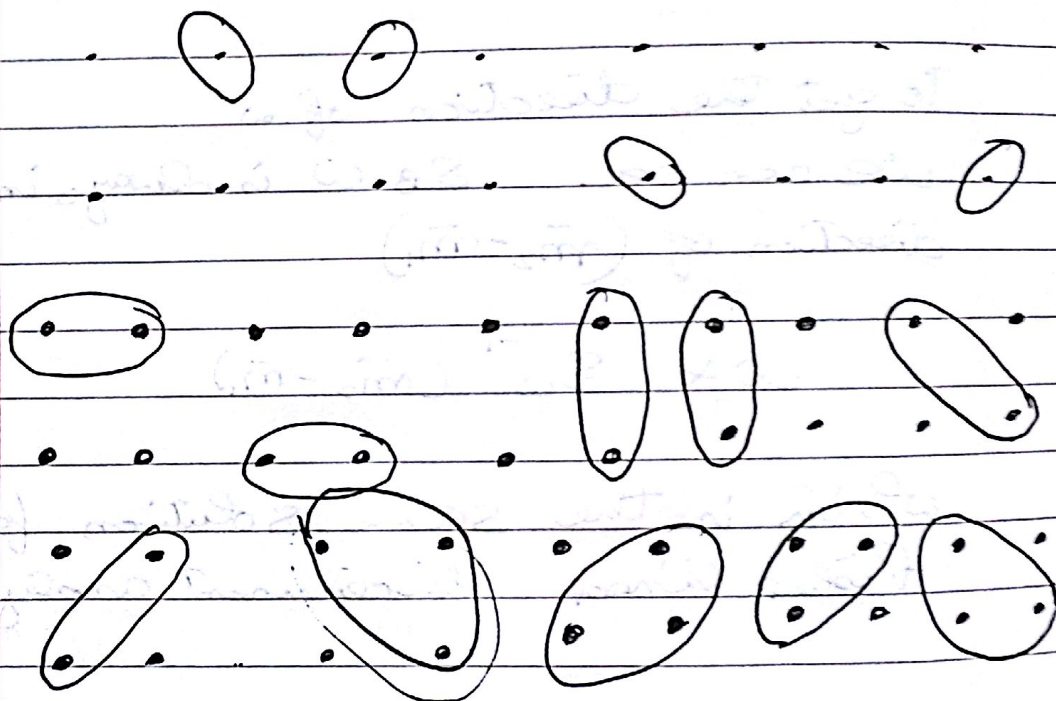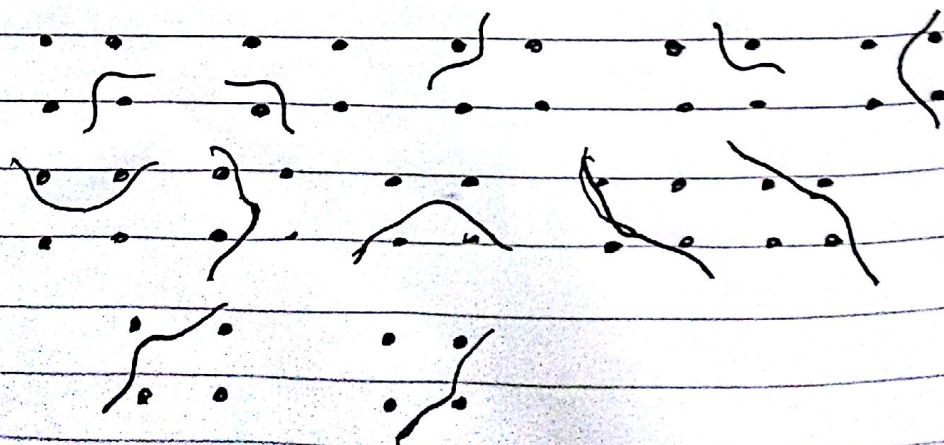which is the same solution for fisher linear disciminant analysis

Q5   Data points x_1 (-1,-1) x_2 (-1,+1)
                      x_3 (+1,-1) x_4 (+1,+1)

a) Linear kernal cannot shatter this

b) Polynomial kernal of degree 2 can
    shatter this. we take here ~~apossible~~
    as ellipse to show shattering

c   Gaussian kernal can ~~od~~ shatter it

Algorithms for programs

Q4) for Bayesian classification we are

1. A loop will iterate over all values of K
2. We will randomly divide the dataset into k chunks and another loop will iterate over all k chunks
3. Particular i^{th} chunk = test and rest is training dataset
4. We are here using the Gaussian distribution for continuous dataset to find the probability of the occurrence of a given value to an attribute
   P(Y = 2 | attr_1, attr_2,.. attr_n) = P(Y = 2) * P(attr_1 | Y = 2) * P(attr_2 | Y = 2) * P(attr_3 | Y = 2)…. P(attr_n | Y = 2)
   similarly   P(Y = 4 | attr_1, attr_4,.. attr_n) = P(Y = 4) * P(attr_1 | Y = 4) * P(attr_4 | Y = 4) * P(attr_3 | Y = 4)…. P(attr_n | Y = 4)
5. The final output of our test dataset will depend upon whichever probability is greater
6. In the end of the loop we will match with each of the answer of the test dataset and count the total number of error and hence calculate the accuracy

Q6)

a) Iterative re-weighted least squares. – As we know Equation for Iterative re-weighted for logistic regression is
$$\mathbf{w}^{(\text{new})} \;=\; \mathbf{w}^{(\text{old})} - (\Phi^{\mathrm{T}}\mathbf{R}\Phi)^{-1}\Phi^{\mathrm{T}}(\mathbf{y}-\mathbf{t})$$
here y is the predicted output, t is the actual output and R is a diagonal matrix and
$$R_{nn} = y_n(1 - y_n).$$

We will continue with 10 epochs

b) Stochastic Gradient Descent. : It works by taking alpha value and doing iteration over all training dataset and changing weight after computing hypothesis (predicted value) for each data point
$$\theta_j := \theta_j - \alpha(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

here θ_j is the j th term of weight vector $h_\theta(x^{(i)})$ is the predicted value for data point x^i and $x_j^{(i)}$ is the j th value of attribute of data point x^i and α is the learning rate