For XOR data set

x_1	x_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 $x = \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$ $x' = \begin{pmatrix} v_1 \\ v_2 \end{pmatrix}$

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

$$(u_1 v_1 + u_2 v_2 + 1)^2$$

$$= 1 + (u_1 v_1 + u_2 v_2)^2 + 2(u_1 v_1 + u_2 v_2)$$

$$= 1 + u_1^2 v_1^2 + u_2^2 v_2^2 + 2 u_1 v_1 u_2 v_2 + 2 u_1 v_1 + 2 u_2 v_2$$

so the equivalent transformation

$$x \Rightarrow [1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1 x_2]^T$$

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

$$\phi[0,0] = [1, 0, 0, 0, 0, 0]^T$$

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

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

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

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

s.t. $\alpha_i y_i = 0 \Rightarrow \alpha_1 + \alpha_4 = \alpha_2 + \alpha_3$

$\alpha_i \geq 0$

inner product = K $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 4 & 1 & 4 \\ 1 & 1 & 4 & 4 \\ 1 & 4 & 4 & 9 \end{bmatrix}$

Our lagrange equation =

$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}\big(\alpha_1^2 + 4\alpha_2^2 + 4\alpha_3^2 + 9\alpha_4^2 +$

$- \alpha_1\alpha_2 - \alpha_1\alpha_3 + \alpha_1\alpha_4$

$- \alpha_1\alpha_2 - \alpha_1\alpha_3 + \alpha_1\alpha_4$

$+ \alpha_2\alpha_3 + \alpha_2\alpha_3 - 4\alpha_2\alpha_4 - 4\alpha_4\alpha_2 - 4\alpha_3\alpha_4 - 4\alpha_4\alpha_3\big)$

$= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \frac{\alpha_1^2}{2} + 2\alpha_2^2 + 2\alpha_3^2 + 9\alpha_4^2 - \alpha_1\alpha_2$

$- \alpha_1\alpha_3 + \alpha_1\alpha_4 + \alpha_2\alpha_3 - 4\alpha_2\alpha_4 - 4\alpha_3\alpha_4$

$\frac{\partial L}{\partial \alpha_1} = 0 \Rightarrow \alpha_1 - \alpha_2 - \alpha_3 + \alpha_4 = 1 \quad (i)$

$\frac{\partial L}{\partial \alpha_2} = 0 \Rightarrow 4\alpha_2 - \alpha_1 + \alpha_3 - 4\alpha_4 = 1 \quad (ii)$

$\frac{\partial L}{\alpha_3} = 0 \Rightarrow 4\alpha_3 - \alpha_1 + \alpha_2 - 4\alpha_4 = 1 \quad (iii)$

$\frac{\partial L}{\alpha_4} = 0 \Rightarrow 9\alpha_4 + \alpha_1 - 4\alpha_2 - 4\alpha_3 = 1 \quad (iv)$

from the following equations

$\alpha_1 = \frac{13}{3} \quad \alpha_2 = \frac{8}{3} \quad \alpha_3 = \frac{8}{3} \quad \alpha_4 = 2$

as we know

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

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

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

$$g(x_1,x_2) = \omega^T \phi(x)$$

$$= \begin{bmatrix}-1 & \frac{2\sqrt{2}}{3} & \frac{2\sqrt{2}}{3} & \cancel{-2\sqrt{2}\ \frac{2}{3}\ \frac{2}{3}} \ \frac{2}{3} & \frac{2}{3} & -2\sqrt{2}\end{bmatrix}\begin{bmatrix}1\\\sqrt{2}x_1\\\sqrt{2}x_2\\x_1^2\\x_2^2\\\sqrt{2}x_1x_2\end{bmatrix}$$

∴ decision boundary

$$= -1 + \frac{4}{3}x_1 + \frac{4}{3}x_2 + \frac{2}{3}x_1^2 + \frac{2}{3}x_2^2 - 4x_1x_2 = 0$$

Q Bayes classifiers means classification of datapoints into 2 or more classes based on their probability of occurrence of their attributes in Class1 or Class2.

Date
Page

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

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

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

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

from 11 we can compute

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

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

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

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 $(\bar{m}_2 - \bar{m}_1)$

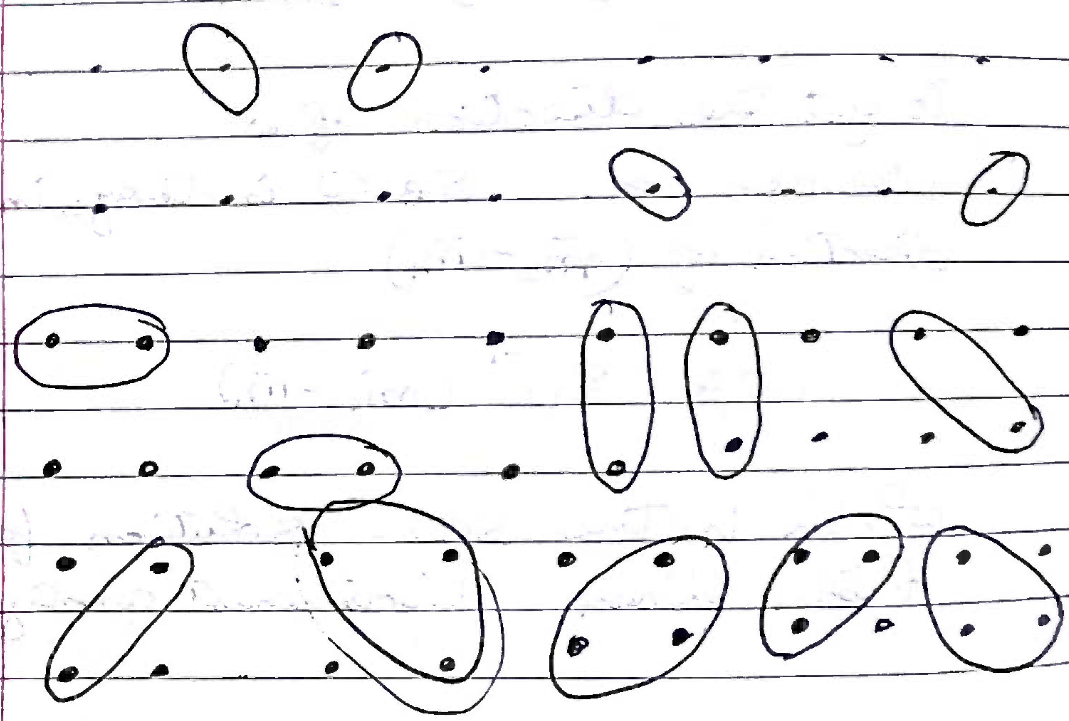
$$\omega \propto S_\omega^{-1} (\bar{m}_2 - \bar{m}_1)$$

which is the same solution for fisher linear discriminant analysis

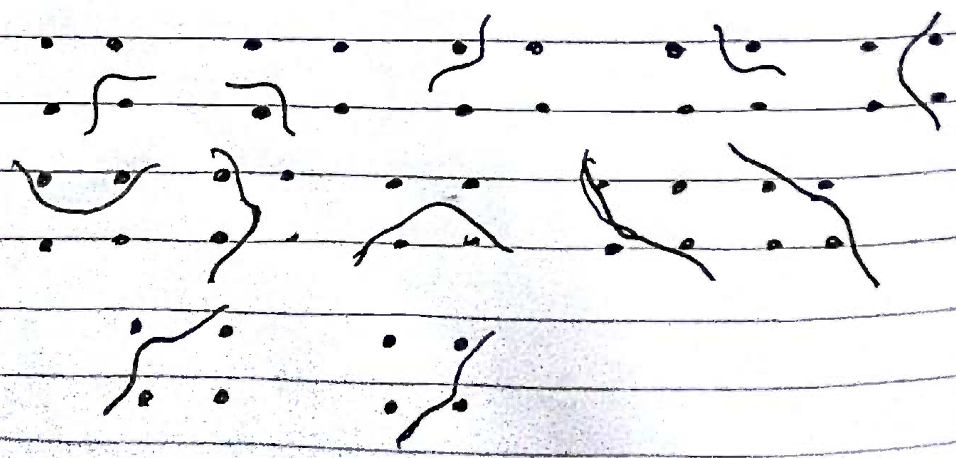
Q5 Data points x_1 (-1,-1) x_2 (-1,+1)
x_3 (+1,-1) x_4 (+1,+1)

a) Linear kernel cannot shatter this

b) Polynomial kernel of degree 2 can shatter this. We take here ~~a possible~~ an ellipse to show shattering



c Gaussian kernel can ~~at~~ 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

$$\mathbf{w}^{(\mathrm{new})} = \mathbf{w}^{(\mathrm{old})} - (\mathbf{\Phi}^{\mathrm{T}}\mathbf{R}\mathbf{\Phi})^{-1}\mathbf{\Phi}^{\mathrm{T}}(\mathbf{y} - \mathbf{t})$$

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