CS6140: Machine Learning

Homework Assignment # 4

Assigned: 03/15/2019 *Due: 03/31/2019, 11:59pm, through Blackboard*

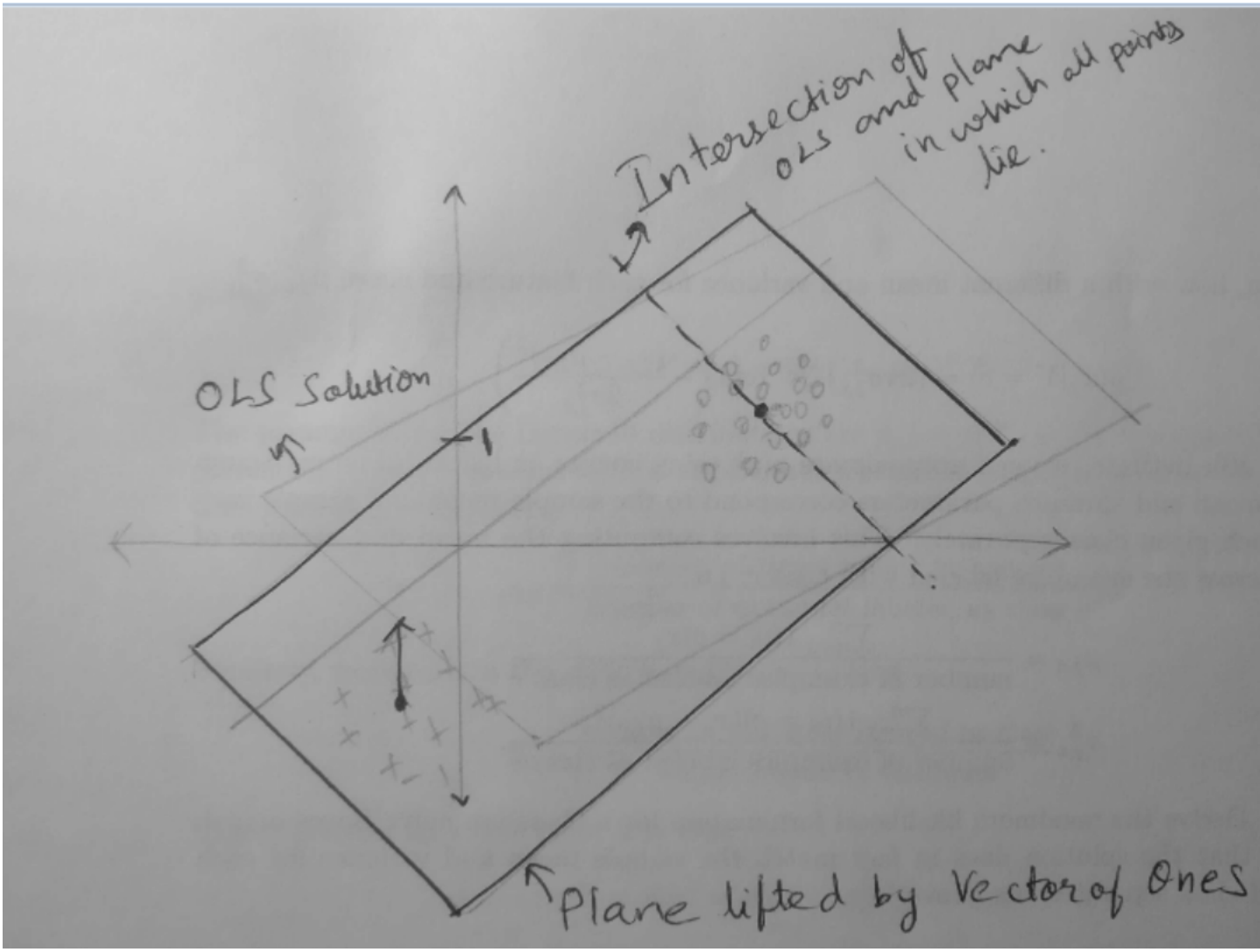
Solution 1:

a) When we plot the scatter of the 2-D distribution after adding vector of ones in 3-D space we find that the all points are lifted by a distance of one and still lie in the same plane in 3D space. The OLS solution given by following,

$w = (X^T X)^{-1} X^T y$, can be used to write the final regression plane as follows:

$y_{solution}$=X$(X^T X)^{-1} X^T y$,

The above equation essentially projects the y vector on the solution plane.



In above picture, imagine a 3D space and the highlighted plane as the plane in which all points lie. The OLS solution projects the y vector onto the solution plane and since the points of class 0 have y as 0 the OLS solution plane will intersect the points-plane where the class 0 points lie. We visualized the highlighted plane and the dotted line as the initial solution for the Logistic Regression problem in Week 6 class code.

b) If we want to modify initial solution such that it passes through somewhere between the positive and negative class labels, we should subtract some positive value c from y vector such that y-c is still positive for positive class label and the y-c value drops negative for negative class label. This way the solution plane will have to capture the negative projection of negative class labels and will have to pass through the points-plane somewhere in between of +ve and -ve points. Interestingly, even if we reverese the (y-c) values for positive and negative class labels, still the plane will pass from somewhere between the points. The idea here is to create opposite signed target values for both classes.

c) By assigning y as {+1,-1} the OLS solution will pass very close to between the +ve and -ve class labels since we have assigned equally but opposite values of target for both the classes.

Please note that above explaination consideres well separated classes. In case of highly intermingled classes

the solution plane will be more towards the orthogonal plane to the plane where points lie.

Solution 2:

Decision surface is given as $P(Y=1|x) = P(Y=0|x)$.

We can use bayes rule to get the probability in terms of posterior and marginalized distributions of x.

$P(Y=1|x) = \frac{P(x|y=1).P(y=1)}{P(x)}$

$P(Y=0|x) = \frac{P(x|y=0).P(y=0)}{P(x)}$

Now we get the decision boundary surface in terms of posterior distributions:

$\frac{P(x|y=1).P(y=1)}{P(x)} = \frac{P(x|y=0).P(y=0)}{P(x)} \Rightarrow P(x|y=1).P(y=1) = P(x|y=0).P(y=0)$

We will use above rule for decision surface for following problems.

a)

We are given $P(Y=1) = P(Y=0) = \frac{1}{2}$, so the above rule reduces to $P(x|y=1) = P(x|y=0)$.

Also, $p(x|Y=i) = \frac{1}{(2\pi)^{d/2}|\sum_i|^{1/2}}.e^{-\frac{1}{2}(x-m_i)^T \sum_i^{-1}(x-m_i)}$

d = 2 is given since it is a 2 dimensional gaussian distribution.

$$p(X|Y=0) = \frac{1}{(2\pi)^{2/2}}.e^{-\frac{1}{2}([\ x_1 \quad x_2\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_1 \quad x_2\ ])}$$

also, $\sum_0 = \sum_1 = I_2, |\sum| = 1$ (since $\sum$ is identity matrix)

$$p(X|Y=0) = \frac{1}{(2\pi)^{2/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}$$

$$\text{and, } p(X|Y=1) = \frac{1}{(2\pi)^{2/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])}$$

Substituting, $m_0 = (1,2), m_1 = (6,3)$

$$p(X|0) = \frac{1}{2\pi}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ 1 \quad 2\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_0 \quad x_1\ ]-[\ 1 \quad 2\ ])}$$

$$\text{and, } p(X|1) = \frac{1}{2\pi}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ 6 \quad 3\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_0 \quad x_1\ ]-[\ 6 \quad 3\ ])}$$

We can clearly see that while equating above two probabilities for getting the decision surface, it will be very helpful to consider log terms on both sides,

Taking the logs of both p(X|0) and p(X|1) we get,

$$-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ 1 \quad 2\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} ([\ x_0 \quad x_1\ ]-[\ 1 \quad 2\ ]) = -\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ 6 \quad 3\ ])^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ]-[\ 6 \quad 3\ ])$$

This equation reduces to following after multiplying the vectors and matrices:

(Please note that $I_2^{-1} = I_2$)

$(x_0 - 1)^2 + (x_1 - 2)^2 = (x_0 - 6)^2 + (x_1 - 3)^2$

$\Rightarrow x_0^2 + 1 - 2x_0^2 + x_2^2 + 4 - 4x_2 = x_0^2 + 36 - 12x_0 + x_1^2 + 9 - 6x_2$

Solving further we will get the decision plane as following:

$$5x_0 + x_1 - 20 = 0$$

b) Generalized solution for decision surface given 2-dimensional gaussian distribution:

Let us state our general parameters first,

$m_0 = (m_{01}, m_{02}), m_1 = (m_{11}, m_{12})$ and $\sum_0 = \sum_1 = \sigma^2 I_2 \Longrightarrow |\sum| = \sigma$

Substituting above values in given distribution for $p(x|y = i)$

$$p(X|Y=0) = \frac{1}{(2\pi)\sigma}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}$$

and, $$p(X|Y=1) = \frac{1}{(2\pi)\sigma}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])^T \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])}$$

(please note in above equation we have used $(\sigma^2 I_2)^{-1} = \sigma^{-2} I_2$

since its given that $p(y = 0) = p(y = 1)$, we will use the rule $P(x|y = 1).P(y = 1) = P(x|y = 0).P(y = 0)$

$$\Rightarrow [\frac{1}{(2\pi)\sigma}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}.P(y=0)] = [\frac{1}{(2\pi)\sigma}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11}$$
1)

we can cancel out common term $\frac{1}{(2\pi)\sigma}$ and then take log on both sides,

$$\frac{-1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ]) + log(P(y=0)) = \frac{-1}{2}([\ x_0 \quad x_1\ ]-$$
$$[\ m_{11} \quad m_{12}\ ])^T \frac{1}{\sigma^2} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^{-1} ([\ x_0 \quad x_1\ ] - [\ m_{11} \quad m_{12}\ ]) + log(P(y=1))$$

$$\Rightarrow \frac{-1}{2\sigma^2}((x_0 - m_{01})^2 + (x_1 - m_{02})^2) + log(P(y=0)) = \frac{-1}{2\sigma^2}((x_0 - m_{11})^2 + (x_1 - m_{12})^2) + log(P(y=1))$$

$$\Rightarrow \frac{-1}{2\sigma^2}((x_0^2 + m_{01}^2 - 2x_0 m_{01} + x_1^2 + m_{02}^2 - 2x_1 m_{02})) + log(P(y=0)) - \frac{1}{2\sigma^2}(x_0^2 + m_{11}^2 - 2x_0 m_{11} + x_1^2 + m_{12}^2 -$$
$$2x_1 m_{12}) - log(P(y=1)) = 0$$

$$\Rightarrow \frac{-1}{2\sigma^2}(x_0^2 + m_{01}^2 - 2x_0 m_{01} + x_1^2 + m_{02}^2 - 2x_1 m_{02} - x_0^2 - m_{11}^2 + 2x_0 m_{11} - x_1^2 - m_{12}^2 + 2x_1 m_{12}) + log(P(y =$$
$$0)) - log(P(y=1)) = 0$$

$$\Rightarrow \frac{-1}{2\sigma^2}(m_{01}^2 - 2x_0 m_{01} + m_{02}^2 - 2x_1 m_{02} - m_{11}^2 + 2x_0 m_{11} - m_{12}^2 + 2x_1 m_{12}) + \frac{log(P(y=0))}{log(P(y=1))} = 0$$

$$\Rightarrow \frac{(m_{01} - m_{11})}{\sigma^2} x_0 + \frac{(m_{02} - m_{12})}{\sigma^2} x_1 + \frac{(m_{11} + m_{01})(m_{11} - m_{01}) + (m_{12} + m_{02})(m_{11} - m_{02})}{2\sigma^2} + \frac{log(P(y=0))}{log(P(y=1))} = 0$$

Using above equation of 2D plane (since it is a linear combination with 2 dimensions) we can get decision surface for any combination of means and standard deviations.

c)

Solving for general parameters, that is to say, considering $\sum_0$ and $\sum_1$to be any random 2x2 matrices:

We will again begin with $P(x|y=1).P(y=1) = P(x|y=0).P(y=0)$

$[\frac{1}{(2\pi)|\sum_0 —^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_0^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}.P(y=0)] = [\frac{1}{(2\pi)|\sum_1|^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11}$
1)

$[\frac{1}{|\sum_0 —^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_0^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}.P(y=0)] = [\frac{1}{|\sum_1|^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}$
1)]

$[\frac{1}{|\sum_0 —^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_0^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])}.P(y=0)] = [\frac{1}{|\sum_2|^{1/2}}.e^{-\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}$
1)]

$\Rightarrow log\frac{1}{|\sum_0|^{1/2}} + (\frac{-1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_1^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ]) + log(P(y=0))-$

$log\frac{1}{|\sum_1|^{1/2}} - (\frac{-1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])^T \sum_2^{-1}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])) - log(P(y=1)) = 0$

$\Rightarrow -\frac{1}{2}log|\sum_0| - \frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_1^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ]) + \frac{1}{2}log|\sum_1| + (\frac{1}{2}([\ x_0 \quad x_1\ ]-$
$[\ m_{11} \quad m_{12}\ ])^T \sum_2^{-1}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])) + \frac{log(P(y=0))}{log(P(y=1))} = 0$

$\Rightarrow -\frac{1}{2}log|\sum_0| - \frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_1^{-1}([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ]) + \frac{1}{2}log|\sum_1| + (\frac{1}{2}([\ x_0 \quad x_1\ ]-$
$[\ m_{11} \quad m_{12}\ ])^T \sum_2^{-1}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])) + \frac{log(P(y=0))}{log(P(y=1))} = 0$

$\Rightarrow \frac{1}{2}\frac{log|\sum_1|}{log|\sum_0|} + (\frac{1}{2}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ])^T \sum_2^{-1}([\ x_0 \quad x_1\ ]-[\ m_{11} \quad m_{12}\ ]) - ([\ x_0 \quad x_1\ ]-[\ m_{01} \quad m_{02}\ ])^T \sum_1^{-1}([\ x_0 \quad x_1$
$[\ m_{01} \quad m_{02}\ ])) + \frac{log(P(y=0))}{log(P(y=1))} = 0$

The above solution can be used in the most general case to find the optimal decision surface for 2D gaussian distributions.

Solution 3:

We are given the probability of positive class and using that we can form the bernoulli distribution for positive and negative outcomes,

$$P(Y=1|X,W) = \frac{1}{2}\left(1 + \frac{W^T X}{\sqrt{1+(W^T X)^2}}\right)$$

Now classes can be distributed as a bernoulli of above probability as success:

$$p(y|x) = \begin{cases} \frac{1}{2}\left(1 + \frac{W^T X}{\sqrt{1+(W^T X)^2}}\right)^y & y=1 \\ \left(1 - \frac{1}{2}\left(1 + \frac{W^T X}{\sqrt{1+(W^T X)^2}}\right)\right)^{1-y} & y=0 \end{cases}$$

We now have basic probability distribution to form our conditional likelihood function $p(y|X,w)$ or l(w):

$\text{l(w)} = \prod p(y_i|X_i, W)$

$$\text{l(w)} = \prod \frac{1}{2}\left(1 + \frac{W^T X_i}{\sqrt{1+(W^T X_i)^2}}\right)^{y_i}\left(1 - \frac{1}{2}\left(1 + \frac{W^T X_i}{\sqrt{1+(W^T X_i)^2}}\right)\right)^{1-y_i}$$

Now our objective becomes to maximize the likelihood function to achieve $w_{ML} = argmax\{l(w)\}$

The Maximum Likelihood Estimate of l(w) will give us the optimal weights for the decision boundary for our classifier.

We will first take the log of likelihood ll(w) and then differentiate the ll(w) to optimize l(w)

$$\text{ll(w)} = \sum y_i.log\left(\frac{1}{2}\left(1+\frac{W^TX_i}{\sqrt{1+(W^TX_i)^2}}\right)^2\right) + (1-y_i).log\left(-\frac{1}{2}\left(1+\frac{W^TX_i}{\sqrt{1+(W^TX_i)^2}}\right)\right)$$

$$=\sum y_i log\frac{1}{2} + y_i log\left(1+\frac{W^TX_i}{\sqrt{1+(W^TX_i)^2}}\right) + (1-y_i)log\left(\frac{1}{2}\right) + (1-y_i)\left(log\left(1+\frac{W^TX_i}{\sqrt{1+(W^TX_i)^2}}\right)\right)$$

$$=nlog\frac{1}{2} + \sum y_i log\left(1+\frac{W^TX}{\sqrt{1+(W^TX_i)^2}}\right) + (1-y_i)log\left(1-\frac{W^TX}{\sqrt{1+(W^TX_i)^2}}\right)$$

Now, we will take the derivative of above function

We can easily find the derivative of above function using Wolfram Alpha Calculator, we will get the following result

$$\frac{\partial ll(w)}{\partial w_j} = \sum y_i x_{ij}\left(\frac{\sqrt{1+(W^TX_i)^2}-W^TX_i}{1+(W^TX_i)^2}\right) - (1-y_i)x_{ij}\left(\frac{\sqrt{1+(W^TX_i)^2}-W^TX_i}{1+(W^TX_i)^2}\right)$$

$$= \sum \frac{y_i x_{ij}(\sqrt{1+(W^TX_i)^2}-W^TX_i) - (1-y_i)x_{ij}(\sqrt{1+(W^TX_i)^2}+W^TX_i)}{1+(W^TX_i)^2}$$

$$= \sum \frac{y_i x_{ij}\sqrt{1+(W^TX_i)^2} - y_i x_{ij}W^TX_i - x_{ij}\sqrt{1+(W^TX_i)^2} - x_{ij}W^TX_i + y_i x_{ij}\sqrt{1+(W^TX_i)^2} + y_i x_{ij}W^TX_i}{1+(W^TX_i)^2}$$

$$= \sum \frac{(2y_i-1)x_{ij}\sqrt{1+(W^TX_i)^2} - X_{ij}W^TX_i}{1+(W^TX_i)^2} = \sum x_{ij}\frac{(2y_i-1)\sqrt{1+(W^TX_i)^2} - W^TX_i}{1+(W^TX_i)^2}$$

Given above rule for gradient calculation we can write the following parameter update rule:

$$w^{(t+1)} = w^{(t)} + \eta(\frac{\partial ll(w^{(t)})}{\partial w})$$

Since we are using Stochastic Gradient Descent algorithm, we will need to set the learning rate(η) and set an escape criteria say when change in parameter update is not significant or in other words, when difference in w is less than a certain tolerance (τ) we exit the iterative loop.

Using some hit and trial in the beginning we can arrive at optimum values of η and τ. In this particular case we set the values as below:

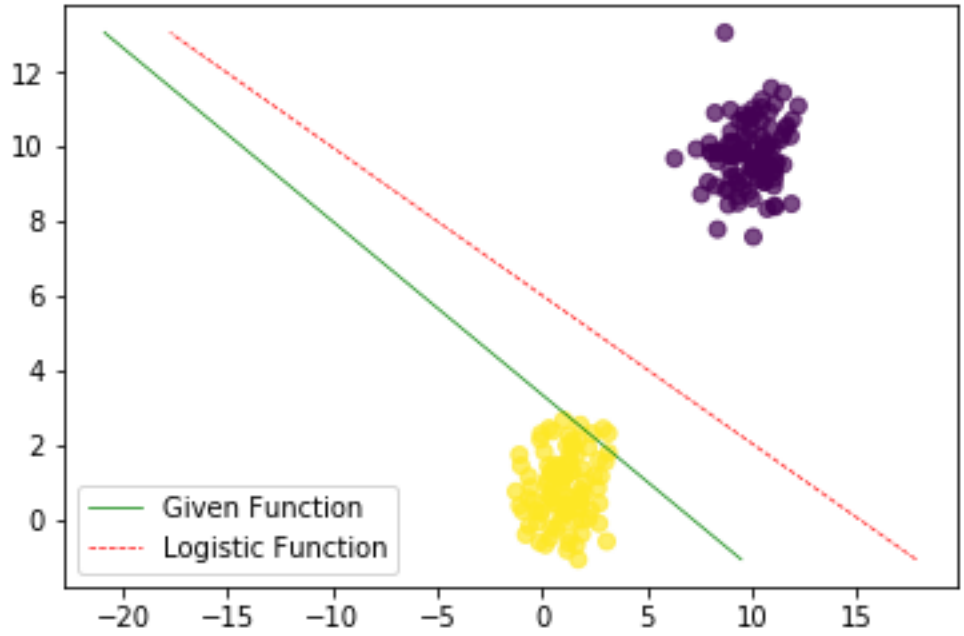
eta = 1e-4, tol = 1e-6

Now we will proceed with 10 test cases and try to understand the difference in linear classifier using given function and classic logistic function.

1. #INITIALIZE

npos = 100; nneg = 100; mu1 = np.array([1,1]); mu2 = np.array([10,10]); sig1 = np.array([[1,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

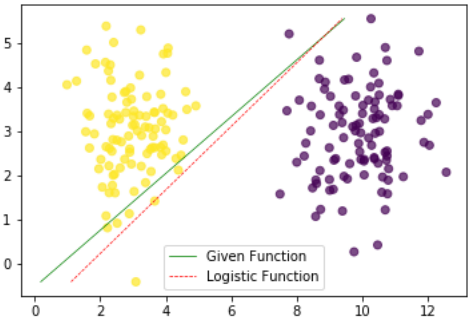
eta = 1e-4; tol = 1e-6;



2. #INITIALIZE

npos = 100; nneg = 100; mu1 = np.array([3,3]); mu2 = np.array([10,3]); sig1 = np.array([[1,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

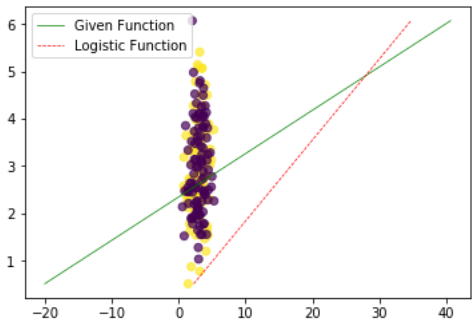
eta = 1e-5; tol = 1e-6;



3. #INITIALIZE

npos = 100; nneg = 100; mu1 = np.array([3,3]); mu2 = np.array([3,3]); sig1 = np.array([[1,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

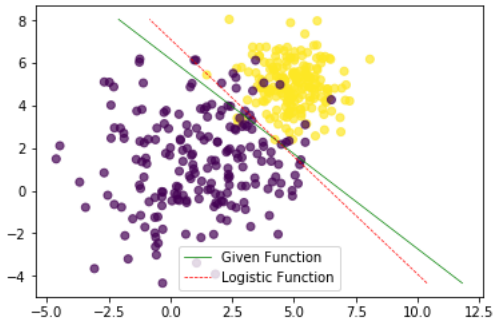
eta = 1e-5; tol = 1e-6;



4. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([5,5]); mu2 = np.array([1,1]); sig1 = np.array([[1,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

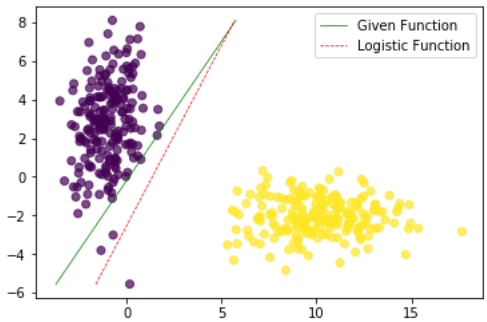
eta = 1e-5; tol = 7e-8;



5. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([10,-2]); mu2 = np.array([-1,3]); sig1 = np.array([[5,0],[0,1]]); sig2 = np.array([[1,0],[0,5]]); max_step = 2e5;

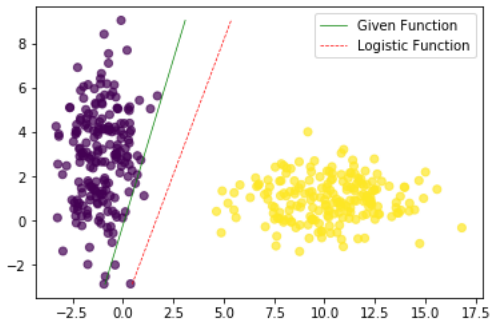
eta = 1e-5; tol = 7e-8;



6. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([10,1]); mu2 = np.array([-1,3]); sig1 = np.array([[5,0],[0,1]]); sig2 = np.array([[1,0],[0,5]]); max_step = 2e5;

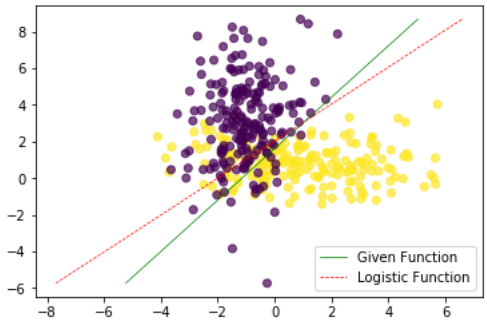
eta = 1e-5; tol = 1e-6;



7. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([1,1]); mu2 = np.array([-1,3]); sig1 = np.array([[5,0],[0,1]]); sig2 = np.array([[1,0],[0,5]]); max_step = 2e5;

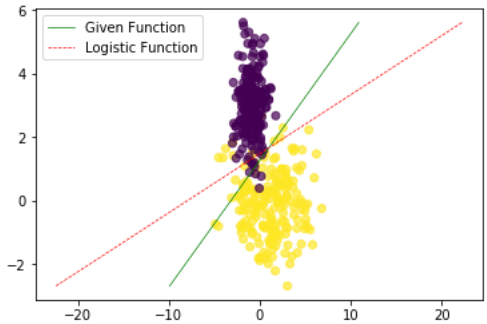
eta = 1e-5; tol = 1e-6;



8. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([1,0]); mu2 = np.array([-1,3]); sig1 = np.array([[5,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

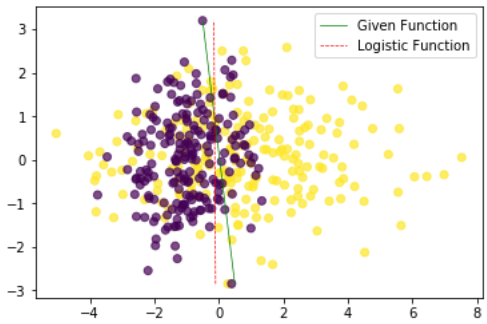
eta = 1e-5; tol = 1e-6;



9. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([1,0]); mu2 = np.array([-1,0]); sig1 = np.array([[5,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

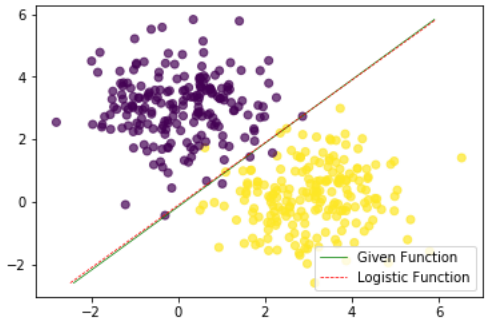
eta = 1e-5; tol = 1e-6;



10. #INITIALIZE

npos = 200; nneg = 200; mu1 = np.array([3,0]); mu2 = np.array([0,3]); sig1 = np.array([[1,0],[0,1]]); sig2 = np.array([[1,0],[0,1]]); max_step = 2e5;

eta = 1e-5; tol = 1e-6;



From above experimentation we noted that when classes are too far off, logistic regression works better and when points are intermingled then given regression works better.

From last test we can see that when separation and intermingling are both mininmum then both the regressions give almost same decision boundary.

Solution 4:

Ordinary Least Squares VS Generalized Linear Model (Poisson)

a) We use OLS method to obtain the linear fit using below weights and target calculations:

$w = (X^T X)^{-1} X^T y$

$y_{target} = Xw$

This implementation can be found in the code provided in file "Problem4.py"

We get the following results using above method:

OLS LINEAR REGRESSION:

SumOfSquares: 1453.7913639723213

MeanSquaredErr: 8.403418288857349

b) Next we proceed with generalized linear model using Poisson distribution as our link function.

We can express the distribution as following link:

$E[y|x] = e^{w^T x}$or $log(E[y|x]) = w^T x$

and, $p(y|x) = Poisson(\lambda)$

Poisson distribution has a mean of λ which is same as saying the following:

$E[y|x] = \lambda$or $\lambda = e^{w^T x}$

Now, we can write the distribution of our prediction as follows:

$p(y|x) = \frac{e^{w^T xy}.e^{-e^{w^T x}}}{y!}$

And use MLE of above distribution to estimate w. Taking log of p(y|x),

$ll(w) = \sum w^T x_i y_i - e^{w^T x_i} - y_i!$

or $ll_i(w) = w^T x_i y_i - e^{w^T x_i} - y_i!$

We will optimize the above function by minimizing the negative log likelihood function:

$-\frac{\delta(ll_i(w))}{\delta w_j} = e^{w^T x_i} x_{ij} - x_{ij} y_i = x_{ij}(e^{w^T x_i} - y_i)$

here, $e^{w^T x_i}$is our prediction for x_i data point.

Therefore, $- \bigtriangledown (ll_i(w)) = x_i(p_i - y_i)$

We can calculate the gradients for all data points using following matrix operation:

$-\bigtriangledown(ll(w)) = \boldsymbol{X^T(p - y)}$

Also, we can use second order derivative hessian, for our final weight update rule:

$-\frac{\delta^2(ll_i(w))}{\delta w_j \delta w_k} = x_{ij} e^{w^T x} x_{ik} = x_{ij} p_i x_{ik}$

Similar to gradient we can use matrix operation to find hessian matrix:

$-\frac{\delta^2(ll(w))}{\delta w_j \delta w_k} = \boldsymbol{-X^T P X}$

Using gradient and hessian calculated above we get the following update rule for weights:

$w^{(t+1)} = w^{(t)} - \boldsymbol{(X^T P X)^{-1} X^T (p - y)}$

Implementation of above rule to calculate GLM predictions is present in the file “Problem4.py”

We get the following results from this implementation,

GLM (POISSON):

SumOfSquares: 1509.6152821310325

MeanSquaredErr: 8.726099896711172