CSCI-567: Assignment # 2

Due on Monday, October 5, 2015

Saket Choudhary
2170058637

Contents

Todo list

Problem 1

Problem 1: (a)

Linear regression assumes that the regressors have been observed 'truly' and as such the dependent variables Y are the ones that are uncertain. The analogy is simpler to think when Y is a 'response', caused due to some indepndent variable X. hence though X is measured absolutely, (dependent varible) Y's measurement should be accounted for errors.

Problem 1: (b)

In order to make linear regression robust to outliers, a näive solution will choose "absolute deviation"($L1$ norm) over "squared error"($L2$ norm) as the criterion for loss function. The reason this might work out in most cases(especially when the outliers belong to a non normal distribution) is that "squared error" will blow up errors when they are large. Thus $L2$ norm will give more weight to large residuals($|y - w^T x|^2 >> |y - w^T x|$ and we are trying to minimise this error), while the $L1$ norm gives equal weights to all residuals.

Problem 1: (c)

A quick way to realise this is to consider the "scale" of any two independent variables. Say one of the dependent variables is 'time'. Rescaling time from hours to seconds will also rescale its coefficient(approximately by a factor of 60), but the importance remains the same!
Another example is to consider a model with two dependent variables that affect the dependent variable in a similar manner(or are equally important regressors). However they are on different scales say $X1$ on [1-100] and $X2$ on [0-1], resulting in the coefficient of $X1$ being too smaller than that of $X2$ in a linear regression setting.

Problem 1: (d)

If the dependent variables are perfect linear combination, the matrix XX^T will be non invertible.

Problem 1: (e)

A simple solution would be to use $k-1$ bits instead of k bits for k categories. For example, using the following setup for a 3 category setup:

$$\begin{aligned} Red &= 0\ 0 \\ Green &= 1\ 0 \\ Blue &= 0\ 1 \end{aligned}$$

Here is an alternate solution that exploits the property of features still being equidistant from the origin(though they are no longer equidistant from each other)

Problem 1 continued on next page. . .

Problem 1: (f)

If the independent variables are highly correlated, the coefficients might still be entirely different. From the example in Part (c) abov

Problem 1: (g)

Using a posterior probability cutoff of 0.5 in linear regression is not same as 0.5 for logistic. A 0.5 rehsold on logistic guarantees that the point all points lying to the right belong to one class. However for a regression problem, this is not true, because the predicted value of y is an 'intepolated or extraplolated' In any case, logistic regression is a better choice, since the output is constrained in the range of $0-1$, which can be treated directly as a probability values as compared to the less intuitive relation with the output of the linear regression.

Problem 1: (h)

When the number of variables exceed the number of samples, the system is undetermined. And yes, it can be solved by simply obtaining psuedo-inverse of X which is always defined.

Problem 2

Problem 2: (a)

Class 1: $\vec{x} = (x_1, x_2, \cdots x_{2D})$ where each $x_i \sim N(0, \sigma^2)$
Class 2: $\vec{x} = (x_1, x_2, \cdots x_D, x_{D+1} + \delta, \cdots x_{2D})$
From first principles, the discriminant curve is given by:

$$P(y = 1|x) \geq P(y = 0|x)$$

And hence we have:

$$\begin{aligned}
\log(P(y = 1|x)) &\geq \log(P(y = 0|x)) \\
\log(P(x|y = 1)p(y = 1) &\geq \log(P(x|y = 0)p(y = 0) \\
\log(P(x|y = 1)) + \log(P(y = 1)) &\geq \log(P(x|y = 0)) + \log(p(y = 0))
\end{aligned} \tag{1}$$

Now, since x is 2D dimensional and assuming independece of all attributes:

$$\begin{aligned}
P(x|y = 1) &= \prod_{i=1}^{2D} p(x_i|y = 1) \\
\log(P(x|y = 1) &= \sum_{i=1}^{2D} \log(p(x_i|y = 1)) \\
\log(P(x|y = 1)) &= -D\log(2\pi\sigma^2) - \sum_{i=1}^{2D} \frac{x_i^2}{2\sigma^2}
\end{aligned} \tag{2}$$

Similarly for class 0: $x_i \sim N(0, \sigma^2) \forall x \in \{1..D\}$ and $x_i \sim N(\delta, \sigma^2) \forall x \in \{D + 1..2D\}$ Notice that the latter is a shifted normal.

$$\begin{aligned}
P(x|y = 0) &= \prod_{i=1}^{2D} p(x_i|y = 0) \\
\log(P(x|y = 0) &= \sum_{i=1}^{D} \log(p(x_i|y = 1)) + \sum_{i=1}^{D} \log(p(x_i|y = 1) \\
\log(P(x|y = 0)) &= -D\log(2\pi\sigma^2) - \sum_{i=1}^{D} \frac{x_i^2}{2\sigma^2} - \sum_{i=D+1}^{2D} \frac{(x_i - \delta)^2}{2\sigma^2}
\end{aligned} \tag{3}$$

Plugging $(2), (3)$ in (1) we get:

$$-D\log(2\pi\sigma^2) - \sum_{i=1}^{2D}\frac{x_i^2}{2\sigma^2} + \log(p(y=1)) \geq -D\log(2\pi\sigma^2) - \sum_{i=1}^{D}\frac{x_i^2}{2\sigma^2} - \sum_{i=D+1}^{2D}\frac{(x_i-\delta)^2}{2\sigma^2} + \log(p(y=0))$$

$$\log(p(y=1)) + -\sum_{i=D+1}^{2D}\frac{x_i^2}{2\sigma^2} \geq -\sum_{i=D+1}^{2D}\frac{x_i^2 - 2\delta x_i + \delta^2}{2\sigma^2} + \log(p(y=0))$$

$$\log(p(y=1)) - \log(p(y=0)) \geq \frac{D\delta}{\sigma^2}\sum_{i=D+1}^{2D} x_i - D\frac{\delta^2}{2\sigma^2}$$

$$\log(p(y=1)) - \log(p(y=0)) \geq \frac{D\delta}{\sigma^2}\sum_{i=1}^{D} x_i - D\frac{\delta^2}{2\sigma^2}$$

$$-\frac{D\delta}{\sigma^2}\sum_{i=1}^{D} x_i + D\frac{\delta^2}{2\sigma^2} + \log(p(y=1)) - \log(p(y=0)) \geq 0$$

Where the change of indices in the penultimate step is permitted since x_i are i.i.d(after taking care of the shifted mean)
Now consider the general form solution of LDA and GDA:

$$\sum_i b_i x_i + c \geq 0 \tag{LDA}$$

$$\sum_i a_i x_i^2 + \sum b_i x_i + c \geq 0 \tag{GDA}$$

Where x_i represents the i^{th} dimension indepdent variable, where each x_i is a feature/attribute and hence is not limited to 2 dimensional special case.
In this case, owing to the homoscedasticity assumption(the variance of the two class conditions being equal to σ^2) LDA and GDA return the same solution.
<u>Solution form for *LDA*:</u>

$$-\frac{D\delta}{\sigma^2}\sum_{i=1}^{D} x_i + D\frac{\delta^2}{2\sigma^2} + \log(p(y=1)) - \log(p(y=0)) \geq 0$$

Assuming equal priors, $p(y=1) = p(y=0)$,

$$-\frac{D\delta}{\sigma^2}\sum_{i=1}^{D} x_i + D\frac{\delta^2}{2\sigma^2} \geq 0$$

$$-\sum_{i=1}^{D} x_i + \frac{\delta}{2} \geq 0$$

and hence for LDA $b_i = 0 \forall i \in \{1..D\}$ and $b_i = -\frac{D\delta}{\sigma^2} \forall i \in \{D+1..2D\}$(simpliifies to -1 for the case with equal priors). and $c = \frac{D\delta^2}{2\sigma^2}$ (simplifies to $\frac{\delta}{2}$ for the case with equal priors)
In either case it does depend on δ
<u>Solution for *GDA*:</u>

$$-\frac{D\delta}{\sigma^2}\sum_{i=1}^{D} x_i + D\frac{\delta^2}{2\sigma^2} + \log(p(y=1)) - \log(p(y=0)) \geq 0$$

and hence $a_i = 0 \forall i$ and $b_i = 0 \forall i \in \{1..D\}$ and $b_i = -\frac{D\delta}{\sigma^2} \forall i \in \{D+1..2D\}$(simplifies to -1 for the case with equal priors).
In either case it does depend on δ

Problem 2: (b1)

$P(X|Y=c_1) \sim N(\mu_1, \Sigma)$ and $p(X|Y=c_2) \sim N(\mu_2, \Sigma)$
where $\mu_1, \mu_2 \in R^D, \Sigma \in R^{D\times D}$

$$\begin{aligned}
P(Y=1|X) &= \frac{P(X|Y=1)P(Y=1)}{P(X)} \\
&= \frac{P(X|Y=1)P(Y=1)}{P(X|Y=1)P(Y=1)+P(X|Y=2)P(Y=2)} \\
&= \frac{1}{1+\frac{P(X|Y=2)P(Y=2)}{P(X|Y=1)P(Y=1)}} \\
&= \frac{1}{1+\exp(\log(\frac{P(X|Y=2)P(Y=2)}{P(X|Y=1)P(Y=1)}))} \\
&= \frac{1}{1+\exp(\log(P(X|Y=2)P(Y=2))-\log(P(X|Y=1)P(Y=1)))} \\
&= \frac{1}{1+\exp(-(\log(\frac{P(Y=1)}{P(Y=2)}))+\log(P(X|Y=2))-\log(P(X|Y=1)))} \qquad (4)
\end{aligned}$$

$$\begin{aligned}
\log(P(X|Y=1)) &= -\frac{1}{2}\ln(|\Sigma|)-\frac{D}{2}\ln(\pi)-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1) \\
\log(P(X|Y=2)) &= -\frac{1}{2}\ln(|\Sigma|)-\frac{D}{2}\ln(\pi)-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) \\
\log(P(X|Y=2))-\log(P(X|Y=1)) &= \frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)-\frac{1}{2}(x-\mu_2)^T\Sigma^{-1}(x-\mu_2) \\
\log(P(X|Y=2))-\log(P(X|Y=1)) &= (\mu_1^T-\mu_2^T)\Sigma^{-1}x+x^T\Sigma^{-1}(\mu_1-\mu_2) \\
\log(P(X|Y=2))-\log(P(X|Y=1)) &= 2(\mu_1^T-\mu_2^T)\Sigma^{-1}x+\mu_1^T\Sigma^{-1}\mu_1-\mu_2^T\Sigma^{-1}\mu_2
\end{aligned}$$

Plugging (5) in 4:

$$\begin{aligned}
P(Y=1|X) &= \frac{1}{1+\exp(-(\log(\frac{P(Y=1)}{P(Y=2)})+\mu_1^T\Sigma^{-1}\mu_1-\mu_2^T\Sigma^{-1}\mu_2+2(\mu_1^T-\mu_2^T)\Sigma^{-1}x))} \\
\P(Y=1|X) &= \frac{1}{1+\exp(-(C)+\theta^T x))}
\end{aligned}$$

Where

$$\boxed{C = \frac{P(Y=1)}{P(Y=2)} - \mu_1^T\Sigma^{-1}\mu_1 + \mu_2^T\Sigma^{-1}\mu_2}$$

$$\boxed{\theta = 2(\mu_1-\mu_2)\Sigma^{-1}}$$

since

$$(\Sigma^{-1})^T = \Sigma^{-1})$$

Problem 2 continued on next page. . .

Problem 2: (b2)

Given $p(y|x)$ is logistic $P(Y=1|X) = \frac{1}{1+\exp(-(C+\theta^T x))}$
Consider the simplification using first principles as in part(b1):

$$P(Y=1|X) = \frac{1}{1+\exp(-(\log(\frac{P(Y=1)}{P(Y=2)})) + \log(P(X|Y=2)) - \log(P(X|Y=1)))}$$

Now, consider the distribution $P(X=x|Y=1) = e^{-\lambda_1}\frac{\lambda_1^x}{x!}$ and $P(X=x|Y=2) = e^{-\lambda_2}\frac{\lambda_2^x}{x!}$
$\log(P(X|Y=2)) - \log(P(X|Y=1) = \lambda_1 - \lambda_2 + x(log\frac{\lambda_1}{\lambda_2})$
and hence,

$$P(Y=1|X) = \frac{1}{1+\exp(-(\log(\frac{P(Y=1)}{P(Y=2)})) + \lambda_1 - \lambda_2 + x(log\frac{\lambda_1}{\lambda_2})))}$$

implying it is possible to arrive at a logistic regression expression even from a poisson distribution and hence the $p(x|y)$ need not be gaussian.

Problem 3

3

$$\begin{aligned}
L(w_{i+1},\lambda) &= ||w_{i+1} - w_i||_2^2 + \lambda(w_{i+1}^T x_i)y_i \\
&= (w_{i+1} - w_i)^T(w_{i+1} - w_i) + \lambda(w_{i+1}) \\
\Delta_{w_{i+1}} L &= 2w_{i+1} - 2w_i + \lambda x_i y_i = 0 \qquad (3.1) \\
\Delta_\lambda L &= w_{i+1}^T x_i y_i = 0 \qquad (3.2)
\end{aligned}$$

Thus, from 3.1 and 3.2,

$$w_{i+1} = w_i - \frac{1}{2}\lambda x_i$$

where

$$w_{i+1}^T x_i y_i = 0$$

Problem 4

4

Problem 4: (a)

Variable	$\#Missing$
pclass	0
survival	0
name	0
sex	0
age	263
sibsp	0
parch	0
ticket	0
fare	1
cabin	1014
embarked	2
boat	823
body	1188
home	564

Problem 4: (b)

From the graph above we see that *pclass* and 'age' might not be really informative. (Given they do not have any monotonicity). whereas the rest all variables have.

Problem 4: (c)

Variable	Information
home	0.999467
name	0.963746
sex	0.963746
ticket	0.963746
embarked	0.963129
cabin	0.940286
fare	0.493161
boat	0.095333
pclass	0.066290
parch	0.034475
age	0.026518
sibsp	0.012308
body	0.00

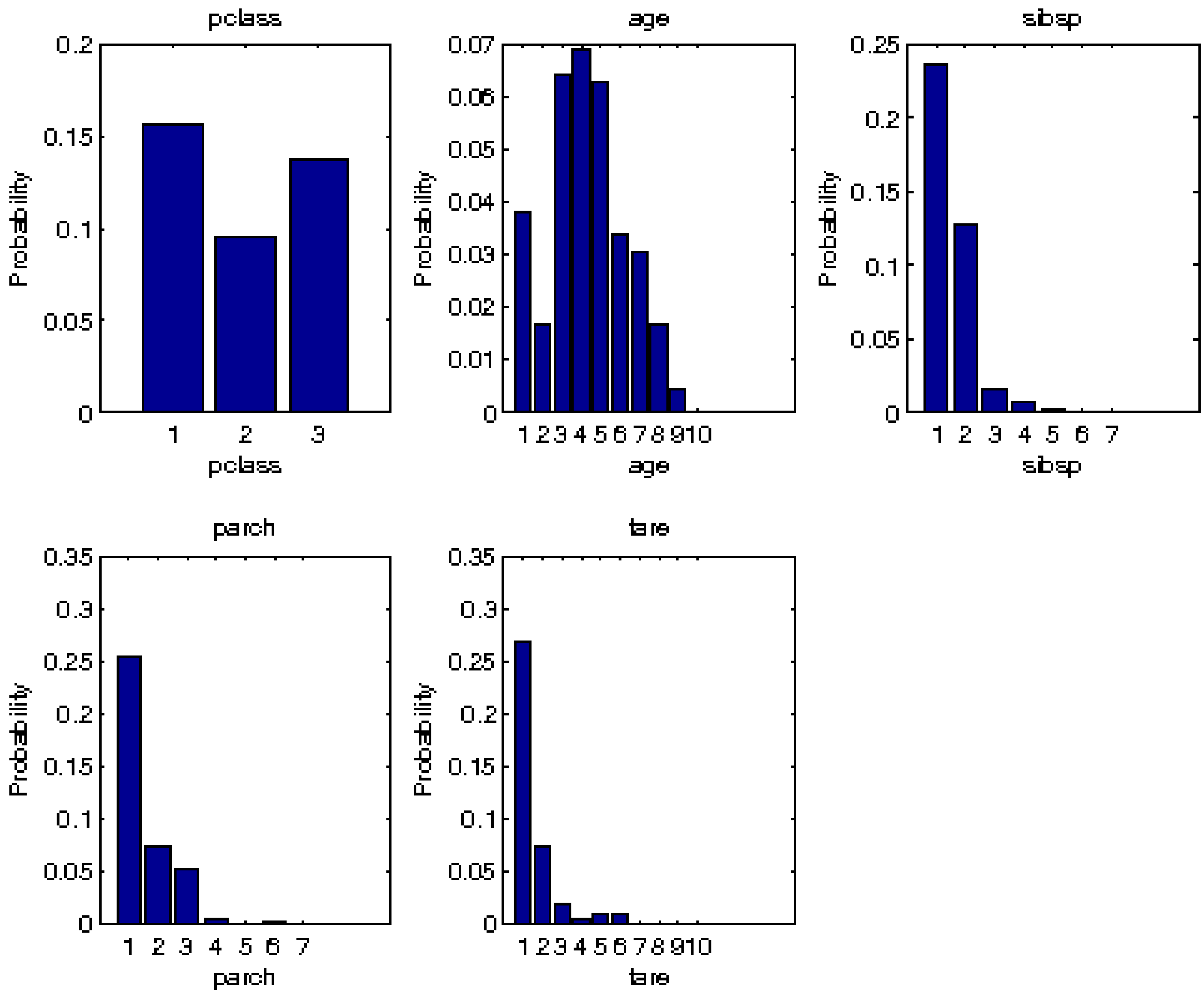

Figure 1: 4a. Montonic relationship

Problem 4: (d)

MM: Multiple Models
SM: Substituted Models
IM: Individual Model

	With age column(IM)	Without Age column(IM)	With NaN substituted(SM)	MM
Training Accuracy	0.776758	0.801223	0.793578	0.629969
Testing Accuracy	0.769466	0.781679	0.775573	0.781679

Thus, the individual model(IM) with the age column completely removed seems to have worked better than the MM. Though in training MM perfroms poorer than SM, its performance is at part with SM for the test dataset. In totality, Substituted model seems to have worked better.(considering both training and test datasets).

This is an indicative that the 'age' factor is not really informative as is also evident from part (c) above where 'age' features low in the information table.

Problem 4: (e)

Total number of columns: 602.

Problem 4: (f)

The method of forward selection seems to have worked well. The training accuracy increased by increasing the number of features iteratively. This also lead to an increase in test accuracy though only marginally. As evident, the training accuracy plot seems to flatten (and hence saturate) near 0.85 for around 10 features. So 10 features can be assumed to be an optimal choice for number of features.

Problem 4: (g)

Alpha: 1.000000e-03 Iterations to Converge: 4862 Accuracy: 7.022901e-01
Alpha: 2.000000e-03 Iterations to Converge: 6114 Accuracy: 7.423664e-01
Alpha: 3.000000e-03 Iterations to Converge: 5807 Accuracy: 7.843511e-01
Alpha: 4.000000e-03 Iterations to Converge: 5731 Accuracy: 8.015267e-01
Alpha: 5.000000e-03 Iterations to Converge: 5393 Accuracy: 8.091603e-01
Alpha: 6.000000e-03 Iterations to Converge: 5192 Accuracy: 8.187023e-01
Alpha: 7.000000e-03 Iterations to Converge: 4914 Accuracy: 8.339695e-01
Alpha: 8.000000e-03 Iterations to Converge: 4730 Accuracy: 8.473282e-01
Alpha: 9.000000e-03 Iterations to Converge: 4624 Accuracy: 8.454198e-01
Alpha: 1.000000e-02 Iterations to Converge: 4513 Accuracy: 8.454198e-01
Alpha: 1.100000e-02 Iterations to Converge: 4371 Accuracy: 8.492366e-01
Alpha: 1.200000e-02 Iterations to Converge: 4218 Accuracy: 8.549618e-01
Alpha: 1.300000e-02 Iterations to Converge: 4102 Accuracy: 8.625954e-01
Alpha: 1.400000e-02 Iterations to Converge: 4036 Accuracy: 8.645038e-01
Alpha: 1.500000e-02 Iterations to Converge: 4024 Accuracy: 8.664122e-01
Alpha: 1.600000e-02 Iterations to Converge: 4078 Accuracy: 8.721374e-01
Alpha: 1.700000e-02 Iterations to Converge: 4198 Accuracy: 8.740458e-01
Alpha: 1.800000e-02 Iterations to Converge: 4366 Accuracy: 8.759542e-01
Alpha: 1.900000e-02 Iterations to Converge: 4564 Accuracy: 8.797710e-01
Alpha: 2.000000e-02 Iterations to Converge: 4760 Accuracy: 8.835878e-01

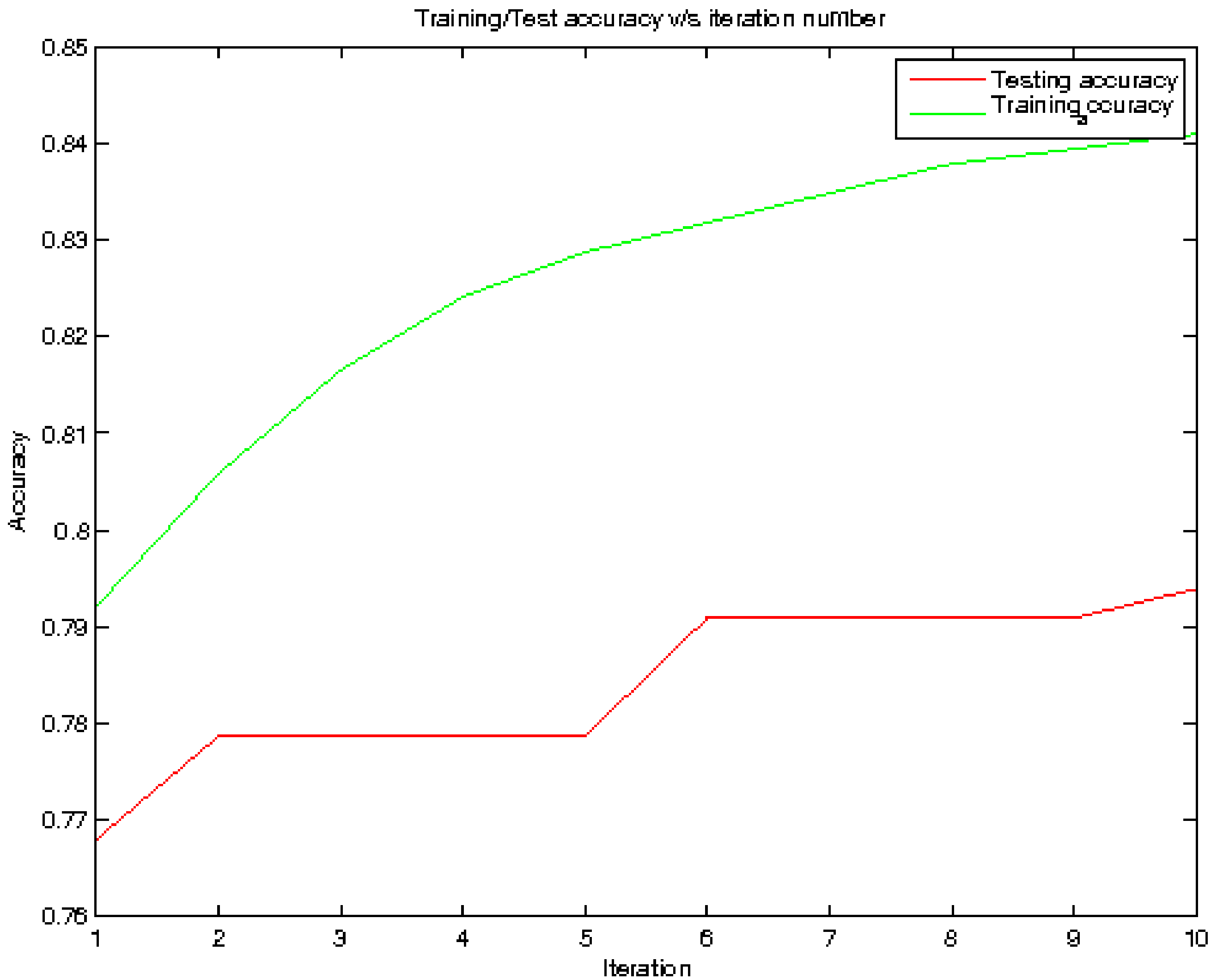

Figure 2: 4f. Training/testing accuracy v/s iteration

Alpha: 2.100000e-02 Iterations to Converge: 4897 Accuracy: 8.835878e-01
Alpha: 2.200000e-02 Iterations to Converge: 4953 Accuracy: 8.874046e-01
Alpha: 2.300000e-02 Iterations to Converge: 4948 Accuracy: 8.874046e-01
Alpha: 2.400000e-02 Iterations to Converge: 4913 Accuracy: 8.912214e-01
Alpha: 2.500000e-02 Iterations to Converge: 4864 Accuracy: 8.912214e-01
Alpha: 2.600000e-02 Iterations to Converge: 4811 Accuracy: 8.931298e-01
Alpha: 2.700000e-02 Iterations to Converge: 4759 Accuracy: 8.912214e-01
Alpha: 2.800000e-02 Iterations to Converge: 4710 Accuracy: 8.931298e-01
Alpha: 5.000000e-02 Iterations to Converge: 4292 Accuracy: 9.217557e-01
Alpha: 5.100000e-02 Iterations to Converge: 4270 Accuracy: 9.236641e-01
Alpha: 5.200000e-02 Iterations to Converge: 4248 Accuracy: 9.236641e-01
Alpha: 5.300000e-02 Iterations to Converge: 4226 Accuracy: 9.236641e-01
Alpha: 5.400000e-02 Iterations to Converge: 4204 Accuracy: 9.217557e-01
Alpha: 5.500000e-02 Iterations to Converge: 4183 Accuracy: 9.217557e-01
Alpha: 5.600000e-02 Iterations to Converge: 4162 Accuracy: 9.217557e-01
Alpha: 5.700000e-02 Iterations to Converge: 4142 Accuracy: 9.217557e-01
Alpha: 1.300000e-01 Iterations to Converge: 3801 Accuracy: 9.408397e-01
Alpha: 1.310000e-01 Iterations to Converge: 3805 Accuracy: 9.408397e-01
Alpha: 1.320000e-01 Iterations to Converge: 3808 Accuracy: 9.408397e-01
Alpha: 1.330000e-01 Iterations to Converge: 3811 Accuracy: 9.408397e-01
Alpha: 1.340000e-01 Iterations to Converge: 3814 Accuracy: 9.408397e-01
Alpha: 1.350000e-01 Iterations to Converge: 3818 Accuracy: 9.408397e-01
Alpha: 1.360000e-01 Iterations to Converge: 3821 Accuracy: 9.408397e-01
Alpha: 1.370000e-01 Iterations to Converge: 3825 Accuracy: 9.408397e-01
Alpha: 1.380000e-01 Iterations to Converge: 3828 **Accuracy: 9.408397e-01**

Thus, it takes approximately 4000 iterations to converge with the choice of the slope parameter around 0.1 and gives an accuracy of 0.94. (Also very low values of the slope parameter seems to hit a local minima) It does seem to be converging in a stable way.
glmfit accuracy: 0.984733

Problem 4: (h)

Number of iterations: 25 Accuracy: 0.583969
glmfit accuracy: 0.984733
Newton's methods implementation seems to be buggy somewhere.