DHANISH RAJ
dr2979

HW 2

Problem 1 (written)

$$y_0 = \arg\max_y \; p(y_0 = y|\pi) \prod_{d=1}^{D} p_d(x_{0,d}|\theta_y^{(d)})$$

$$p_1(x_{0,1}|\theta_y^{(1)}) = (\theta_y^{(1)})^{x_{0,1}} (1-\theta_y^{(1)})^{1-x_{0,1}}, \quad p_2(x_{0,2}|\theta_y^{(2)}) = \theta_y^{(2)}(x_{0,2})^{-(\theta_y^{(2)}+1)}$$

$$\vec{\pi}, \vec{\theta}_y^{(1)}, \vec{\theta}_y^{(2)} = \arg\max_{\pi, \theta_y^{(1)}, \theta_y^{(2)}} \left(\sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i=1}^{n} \ln p(x_{i1}|\theta_{y_i}^{(1)}) + \sum_{i=1}^{n} \ln p(x_{i2}|\theta_{y_i}^{(2)}) \right) \quad ①$$

a) Derive $\hat{\pi}$

We can ignore the second and third terms in equation ① since they do not have π in them

$$\therefore \quad \vec{\pi} = \arg\max_{\pi} \sum_{i=1}^{n} \ln p(y_i|\pi)$$

$$= \arg\max_{\pi} \sum_{i=1}^{n} \ln \left(\pi^{y_i} (1-\pi)^{1-y_i} \right)$$

$$= \arg\max_{\pi} \sum y_i \ln \pi + (1-y_i) \ln(1-\pi) = P$$

$$\frac{\partial P}{\partial \pi} = \sum_{i=1}^{n} \left(\frac{y_i}{\pi} - \frac{(1-y_i)}{1-\pi} \right) = 0$$

$$\Rightarrow \sum (1-\pi) y_i - (1-y_i)\pi = 0$$

$$\Rightarrow \sum y_i - \pi y_i - n\pi + \pi y_i = 0$$

$$\Rightarrow \sum y_i = n\pi \Rightarrow \boxed{\pi = \frac{\sum y_i}{n}}$$

Similarly, $\theta_y^{(j)} = \underset{\theta_y^{(j)}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln p(x_{ij} | \theta_y^{(j)})$

we can ignore the other terms which don't contain $\theta_y^{(j)}$

$$\theta_y^{(j)} = \underset{\theta_y^{(j)}}{\operatorname{argmax}} \sum \ln \left[(\theta_y^{(j)})^{x_{ij}} (1-\theta_y^{(j)})^{1-x_{ij}} \right]$$

$$= \underset{\theta_y^{(j)}}{\operatorname{argmax}} \sum x_{ij} \ln \theta_y^{(j)} + (1-x_{ij}) \ln (1-\theta_y^{(j)}) = Q$$

$$\frac{\partial Q}{\partial \theta_y^{(j)}} = \sum \frac{x_{ij}}{\theta_y^{(j)}} - \frac{(1-x_{ij})}{(1-(1-\theta_y^{(j)}))} = 0$$

$$= \sum x_{ij}(1-\theta_y^{(j)}) - (1-x_{ij})\theta_y^{(j)} = 0$$

$$= \sum x_{ij} - x_{ij}\theta_y^{(j)} - \theta_y^{(j)} + x_{ij}\theta_y^{(j)} = 0$$

$$= \sum x_{ij} = n\theta_y^{(j)} \Rightarrow \boxed{\theta_y^{(j)} = \frac{\sum_{i=1}^{n} x_{ij}}{n}}$$

Similarly, $\vec{\theta}_y^{(2)} = \underset{\theta_y^{(2)}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln P(x_{i2} | \theta_{yi}^{(2)})$

We can ignore the other terms which don't contain $\theta_y^{(2)}$

$\Rightarrow \hat{\theta}_y^{(2)} = \underset{\theta_y^{(2)}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \left(\theta_y^{(2)} (x_{i2})^{-(\theta_y^{(2)}+1)} \right)$

$\Rightarrow \underset{\theta_y^{(2)}}{\operatorname{argmax}} \sum_{i=1}^{n} \ln \theta_y^{(2)} - (\theta_y^{(2)}+1) \ln x_{i2} = R$

$\frac{\partial R}{\partial \theta_y^{(2)}} \Rightarrow \sum_{i=1}^{n} \frac{1}{\theta_y^{(2)}} - \ln x_{i2} = 0$

$\Rightarrow \frac{n}{\theta_y^{(2)}} = \sum_{i=1}^{n} \ln x_{i2} \Rightarrow \boxed{\hat{\theta}_y^{(2)} = \frac{n}{\sum_{i=1}^{n} \ln x_{i2}}}$

Problem 2 (coding)

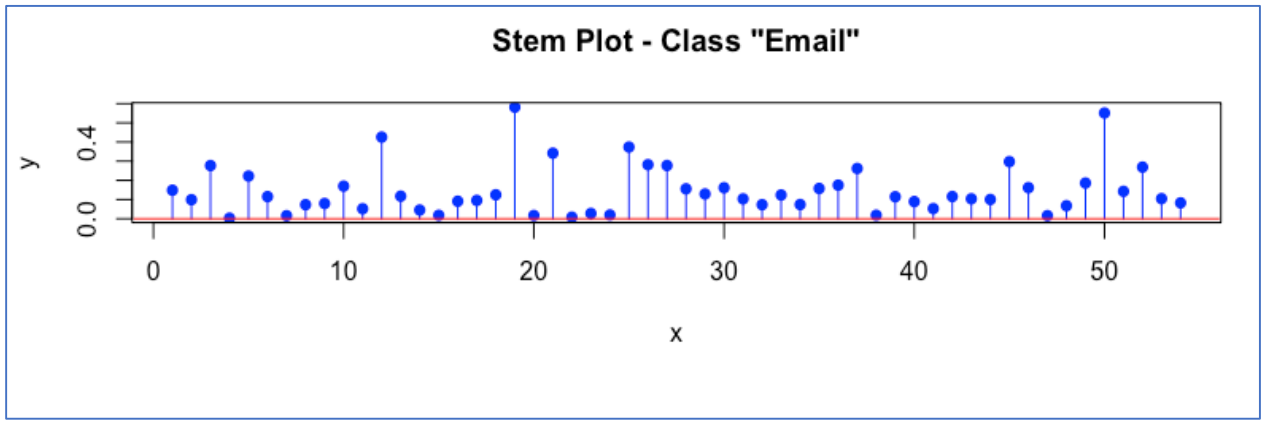
Qs 2a) Naïve Bayes

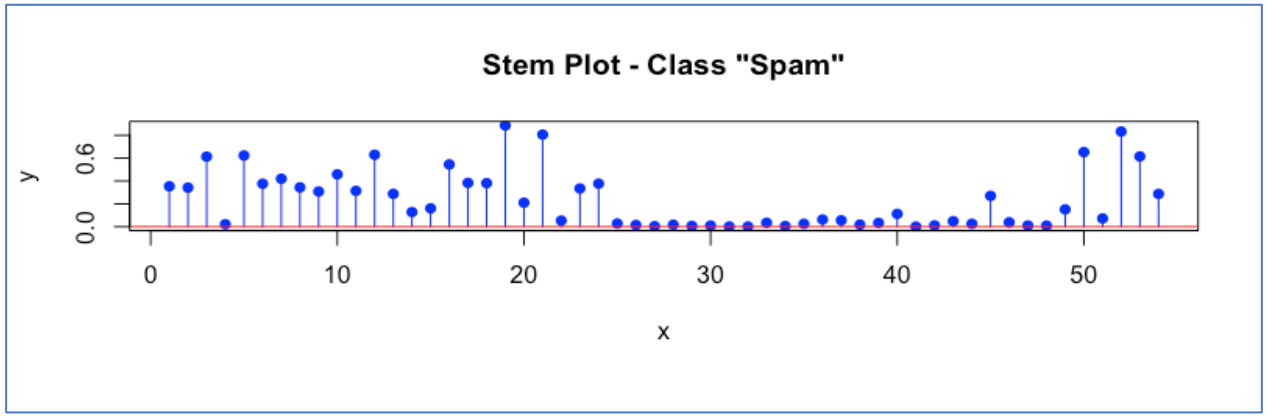
Accuracy

```
        Predicted y
Actual y  0  1
       0  54 2
       1  5  32
```

Prediction Accuracy = (54 + 32)/93 = 92.47312%

Qs 2b)





The 16th dimension corresponds to the word “free”. The probability of this word appearing in a good email is 0.09114202. The probability of this word appearing in a spam email is 0.545045

The 52nd dimension corresponds to the character "!". The probability of this character appearing in a good email is 0.2690337. The probability of this word appearing in a spam email is 0.8333333.

We can see this disparity between the values of these class conditional parameters in the stem plot.

Note: Unlike matlab and Python, there is no way to plot 2 variables in the same stem plot. Hence, I have made two separate plots.

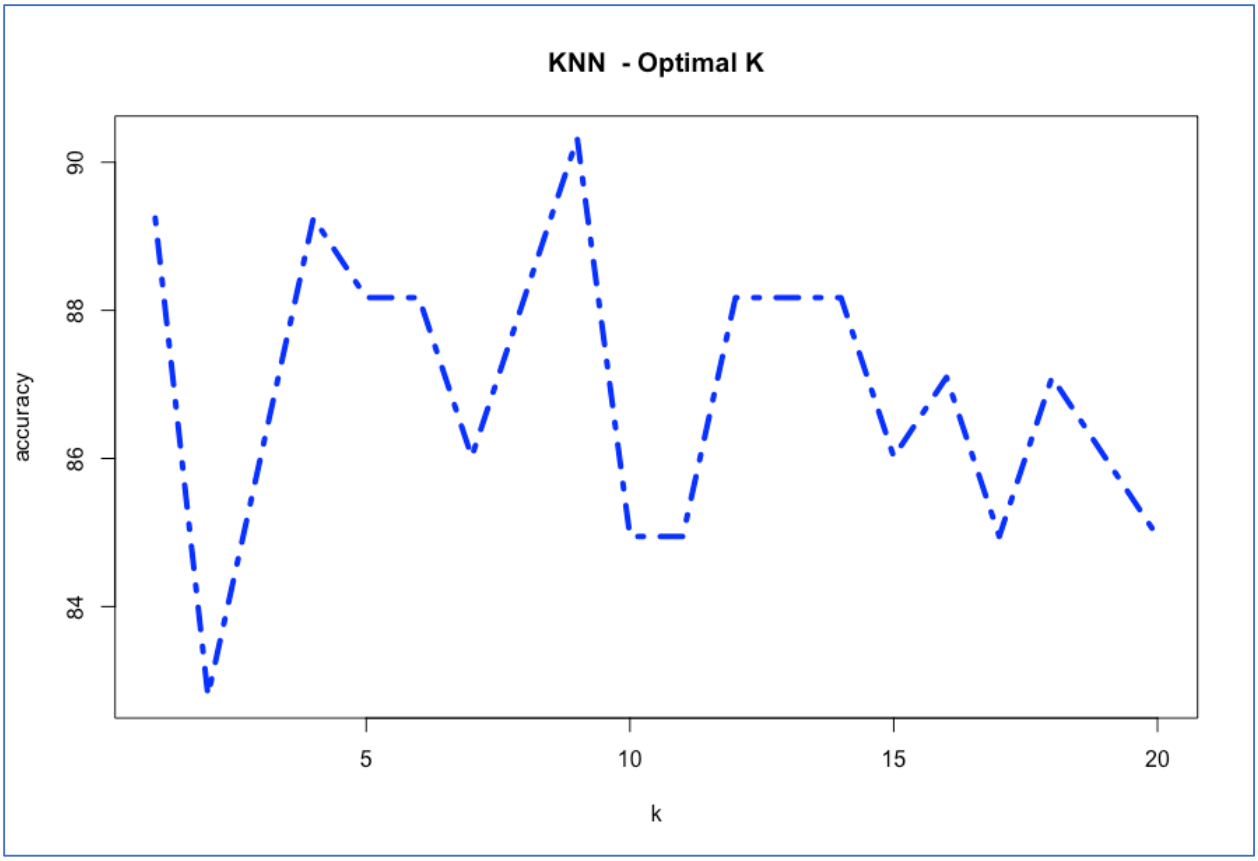
2c)
KNN implementation:

KNN – Decision rule – each time, there is a tie, randomly classify test point as 0 or 1.

The accuracy rates for different values of K are given below:

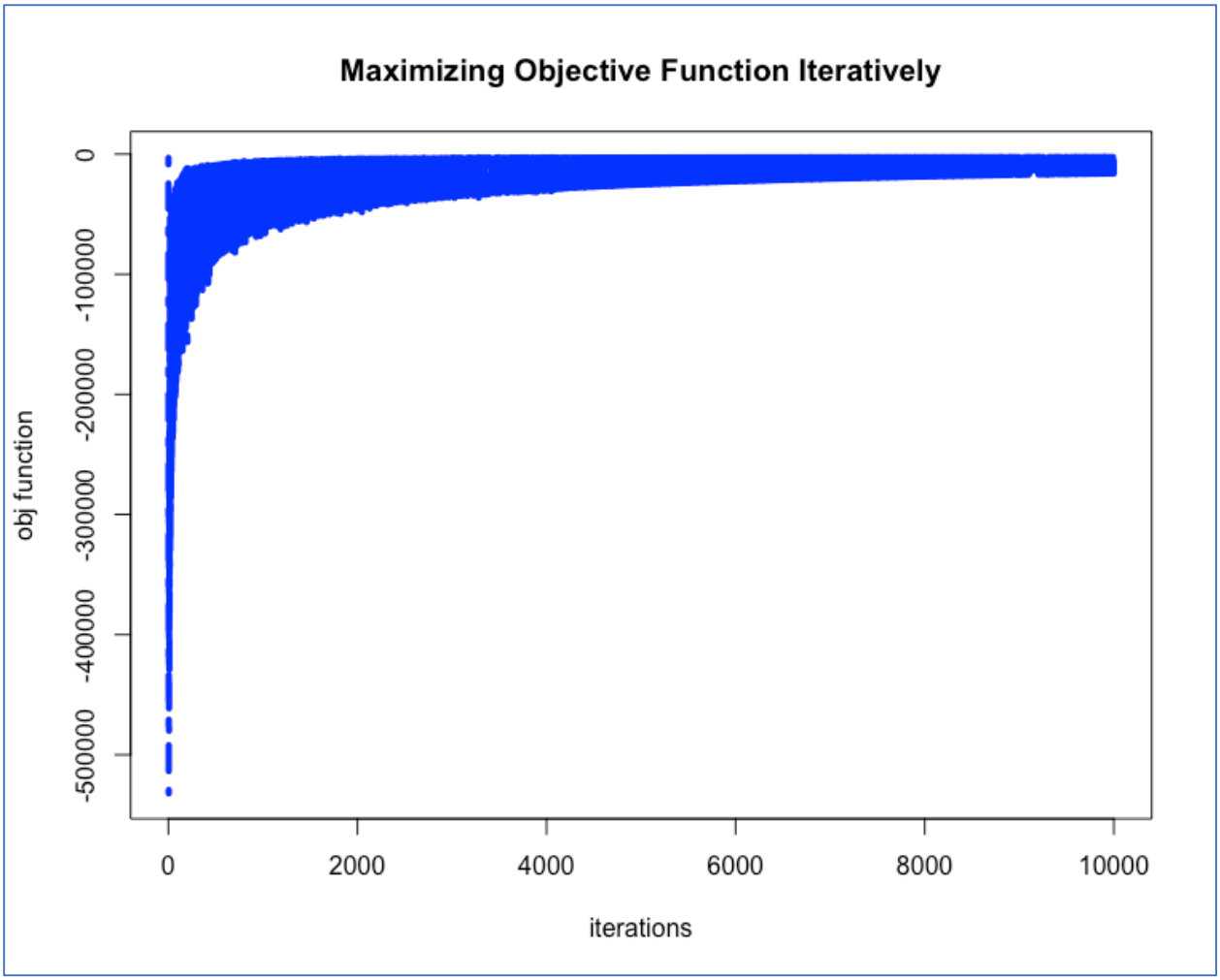
[1] 89.24731 82.79570 86.02151 89.24731 88.17204 88.17204 86.02151 88.17204 90.32258
[10] 84.94624 84.94624 88.17204 88.17204 88.17204 86.02151 87.09677 84.94624 87.09677
[19] 86.02151 84.94624

The plot is below:



We find that k=9, with an accuracy rate of 90.322% is the optimal k. k=2 is the worst k.

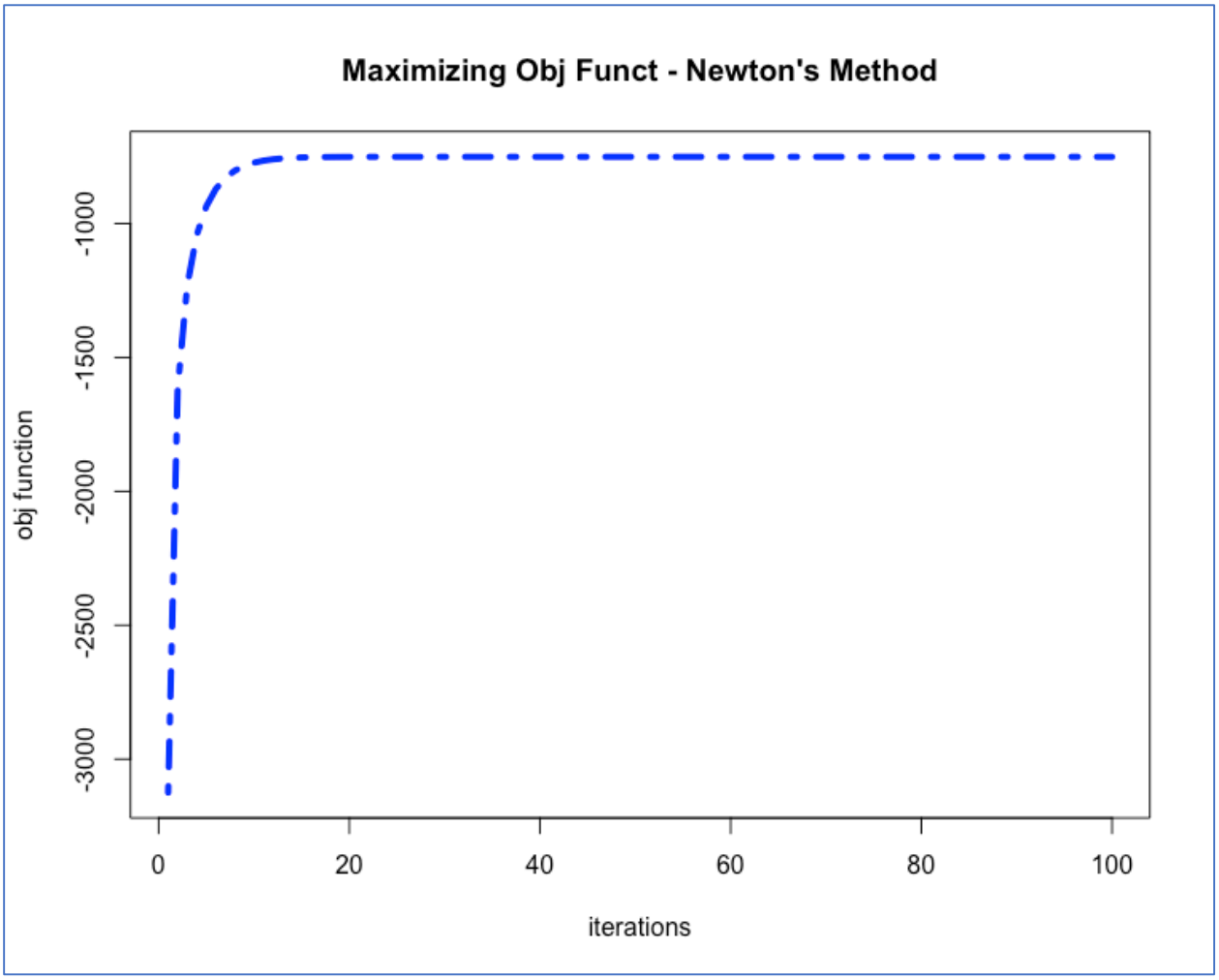
Qs 2d)



The plot looks like this because the loss is not increasing monotonically. We are relying on the gradient. However, when the gradient is 0, the critical point could be a max or a min. Because we are not using information from the hessian, Loss value increases sometimes, then decreases. Eventually, it will converge, but it takes many iterations for this to happen.

The last value of Loss is: -2380.427

Qs 2e) Newton's Method:




```
Accuracy:

       Predicted y
Actual y  0  1
       0 54  2
       1  6 31
```

The accuracy is (54 + 31)/93 * 100 = 91.39785%

As we can see, the loss function increases monotonically. Also, we observe that it converges to its optimal value of -750.71 after only a few iterations **(See the values of vector below)**

```
> newL
 [1] -3124.7075 -1615.4193 -1230.7961 -1042.3731  -935.5359  -870.5155  -829.1870  -
802.1895
 [9]  -784.3607  -772.6337  -764.9700  -759.9714  -756.7101  -754.5827  -753.1971  -752.2980
[17]  -751.7186  -751.3483  -751.1135  -750.9653  -750.8721  -750.8134  -750.7764  -750.7530
[25]  -750.7381  -750.7285  -750.7224  -750.7184  -750.7158  -750.7141  -750.7130  -750.7123
[33]  -750.7118  -750.7114  -750.7112  -750.7111  -750.7109  -750.7109  -750.7108  -750.7108
[41]  -750.7108  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[49]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[57]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[65]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[73]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[81]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[89]  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107  -750.7107
[97]  -750.7107  -750.7107  -750.7107  -750.7107
```