COMS: W4721 Machine Learning for Data Science

HOME WORK - II

Problem 1:

Given distribution of i^{th} dimension,

$i=1$ $\quad p_1(x_{0,1}|\theta_y^{(1)}) = (\theta_y^{(1)})^{x_{0,1}}(1-\theta_y^{(1)})^{1-x_{0,1}}$ (Bernoulli)

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

&

class prior $p(y_0 = y|\pi) = \text{Bernoulli}(y|\pi) = \pi^y(1-\pi)^{1-y}$

MLEs for $\pi, \theta_y^{(1)}, \theta_y^{(2)}$:

$$\hat{\pi}, \hat{\theta}_y^{(1)}, \hat{\theta}_y^{(2)} = \arg\max_{\pi,\theta_y^{(1)},\theta_y^{(2)}} \sum_{i=1}^{n} \ln p(y_i|\pi) + \sum_{i=1}^{n} \ln p_1(x_{i1}|\theta_{y_i}^{(1)}) + \sum_{i=1}^{n} \ln p_2(x_{i2}|\theta_{y_i}^{(2)})$$

$$= \arg\max_{\pi,\theta_y^{(1)},\theta_y^{(2)}} L \quad \longrightarrow (1)$$

(a) Derive $\hat{\pi}$

from equation (1).

$$\nabla_\pi L = 0$$

$$\Rightarrow \nabla_\pi \sum_{i=1}^{n} \ln \pi^{y_i}(1-\pi)^{1-y_i} = 0$$

$$\Rightarrow \nabla_y \quad (\ln\pi)\left(\sum_{i=1}^{n} y_i\right) + \left(\sum_{i=1}^{n}(1-y_i)\right)\ln(1-\pi) = 0$$

$$\Rightarrow \frac{\sum_i y_i}{\pi} + \frac{\sum_i (1-y_i)}{(1-\pi)}(-1) = 0$$

$$\Rightarrow \frac{\sum^{n} y_i}{\pi} = \frac{\sum (1-y_i)}{1-\pi}$$

$$\Rightarrow \sum_{i=1}^{n} y_i - \cancel{\pi \sum_{i=1}^{n} y_i} = n\pi - \cancel{\pi \sum_{i=1}^{n} y_i}$$

$$\Rightarrow \boxed{\hat{\pi} = \frac{\sum_{i=1}^{n} y_i}{n}}$$

(b) $\theta_y^{(1)}$

$$\nabla_{\theta_{y_i}^{(1)}} L = 0$$

$$0 + \nabla_{\theta_{y_i}^{(1)}} \sum_{i=1}^{n} \ln p(x_{i1} / \theta_{y_i}^{(1)}) \left[1(y=y_i) \right] + 0 = 0$$

$$\nabla_{\theta_{y_i}^{(1)}} \sum 1(y=y_i) \ln \left(\theta_{y_i}^{(1)}\right)^{x_{i1}} \left(1-\theta_{y_i}^{(1)}\right)^{1-x_{i1}} = 0$$

$$\nabla_{\theta_{y_i}^{(1)}} \sum 1(y=y_i)\, x_{i1} \ln \theta_{y_i}^{(1)} + \sum 1(y=y_i)(1-x_{i1}) \ln(1-\theta_{y_i}^{(1)}) = 0$$

$$\Rightarrow \frac{\sum 1(y=y_i)\, x_{i1}}{\theta_{y_i}^{(1)}} + (-1)\frac{\sum 1(y=y_i)(1-x_{i1})}{(1-\theta_{y_i}^{(1)})} = 0$$

$$\Rightarrow \sum 1(y=y_i)\, x_{i1} \left(1-\theta_{y_i}^{(1)}\right) = \theta_{y_i}^{(1)} \sum 1(y=y_i)(1-x_{i1})$$

$$\sum 1(y=y_i)\, x_{i1} - \cancel{\theta_{y_i}^{(1)} \sum 1(y=y_i)\, x_{i1}} = \theta_{y_i}^{(1)} \sum 1(y=y_i) - \cancel{\theta_{y_i}^{(1)} \sum 1(y=y_i)\, x_{i1}}$$

$$\boxed{\theta_{y_i}^{(1)} = \frac{\sum 1(y=y_i)\, x_{i1}}{\sum 1(y=y_i)}}$$

(c) $\theta_y^{(2)}$

$$\nabla_{\theta_{y_i}^{(2)}} L = 0$$

$$\Rightarrow 0 + 0 + \nabla_{\theta_{y_i}^{(2)}} \sum 1(y=y_i) \ln \theta_{y_i}^{(2)} (x_{i2})^{-(\theta_{y_i}^{(2)}+1)} = 0$$

$$\Rightarrow \nabla_{\theta_y^{(2)}} \sum_{i=1}^{n} 1(y=y_i) \ln \theta_{y_i}^{(2)} + (-(\theta_{y_i}^{(2)}+1)) \sum_{i=1}^{n} 1(y=y_i) \ln x_{i2} = 0$$

$$\Rightarrow \frac{\sum_{i=1}^{n} 1(y=y_i)}{\theta_{y_i}^{(2)}} + (-1)\sum_{i=1}^{n} 1(y=y_i) \cdot \ln x_{i2} = 0$$

$$\Rightarrow \sum_{i=1}^{n} 1(y=y_i) = \theta_{y_i}^{(2)} \sum_{i=1}^{n} 1(y=y_i) \ln x_{i2}$$

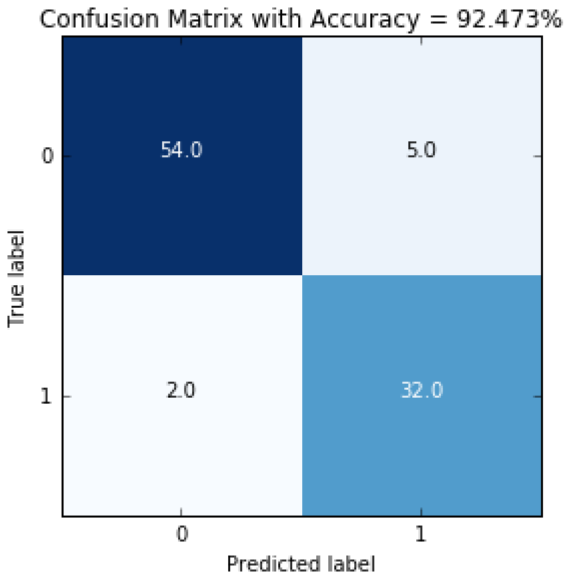
$$\boxed{\tilde{\theta}_{y_i}^{(2)} = \frac{\sum_{i=1}^{n} 1(y=y_i)}{\sum_{i=1}^{n} 1(y=y_i) \cdot \ln x_{i2}}}$$

Out[23]:

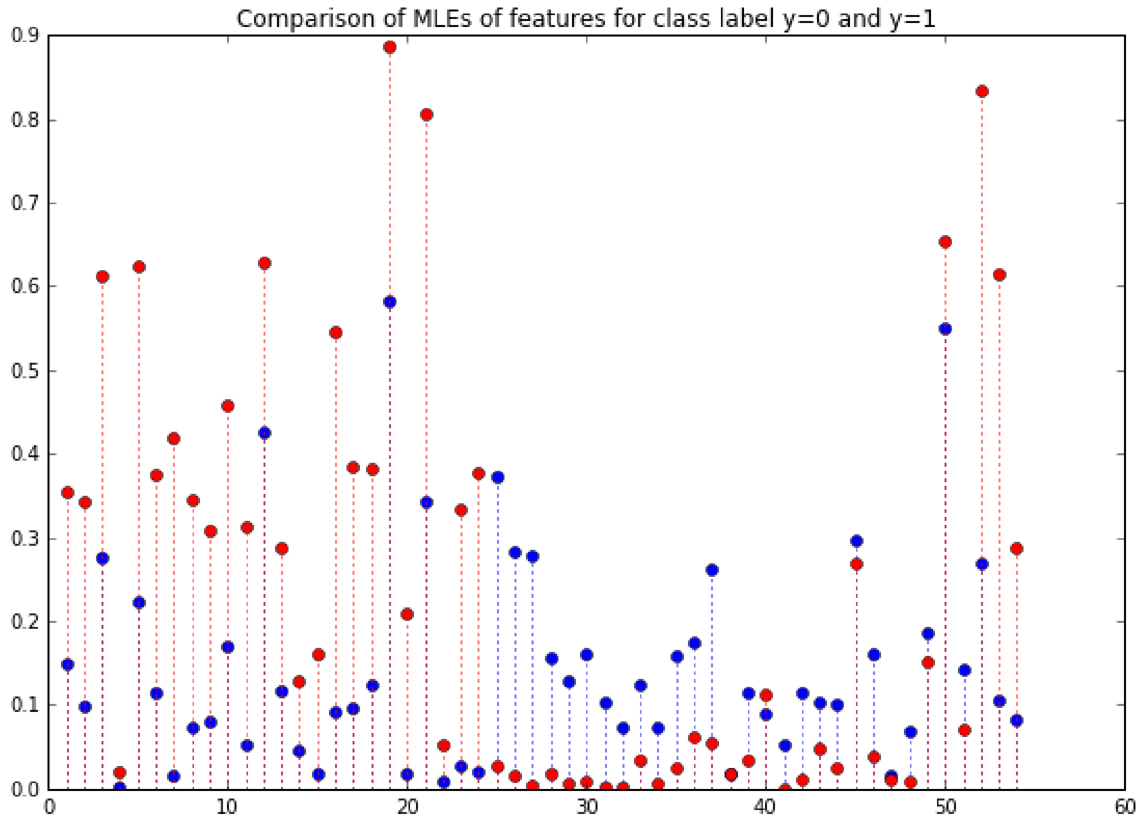
Click here to toggle on/off the raw code.

Question 2:

(a) Implementation of Naive Bayes and Accuracy Table



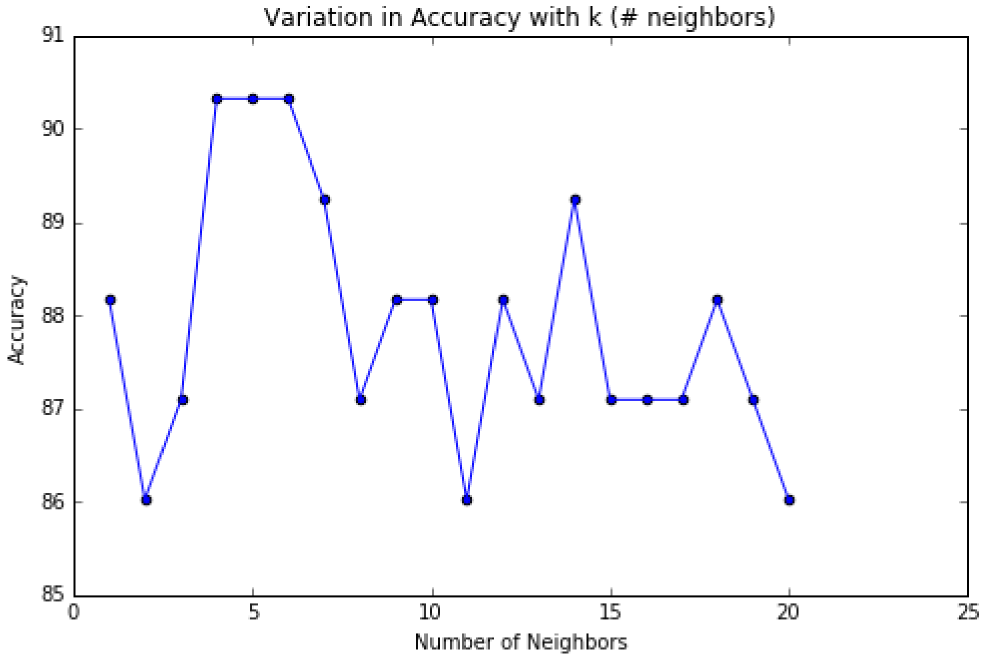
(b) In one figure, show a stem plot (stem() in Matlab) of the 54 Bernoulli parameters for each class. Use the file “spambase.names” to make an observation about dimensions 16 and 52.



The 16th dimension is the frequency of the word 'free' and 52nd dimension is the frequency of exclamation mark '!'. The following are the inferences that can be made:

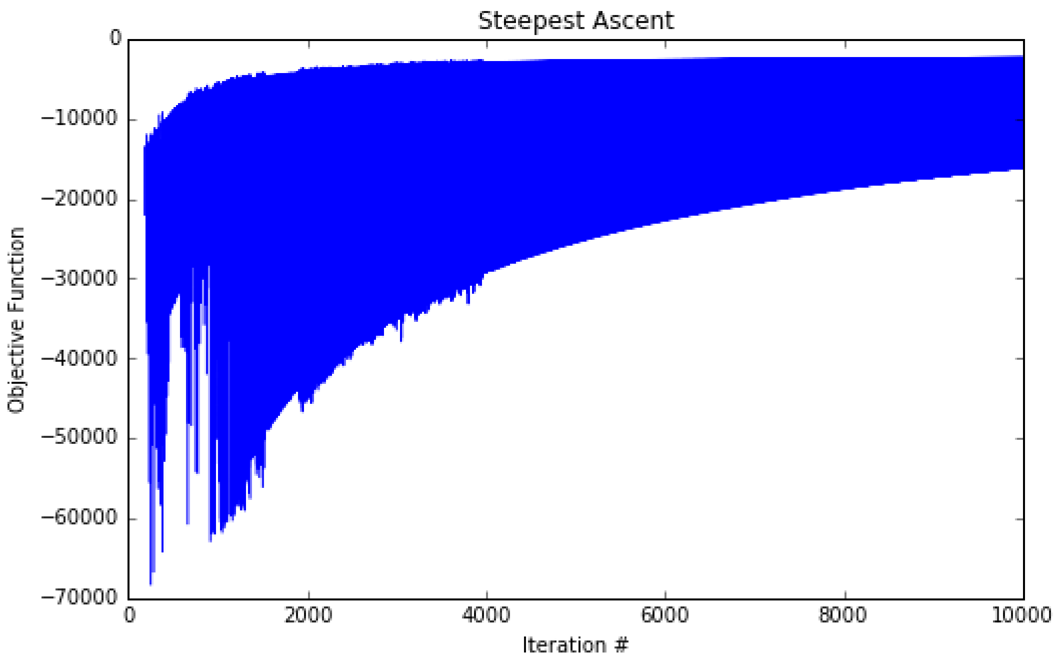
- For class label y=1, both dimensions (16th and 52nd) have much higher MLE values when compared class y=0. This is expected as most spam mails (which are promotions or advertisements) are likely to contain 'free' and '!'. **Given that the mail is spam (y=1) the probability of occurence of 'free' and '!' is high when compared to non=spam**
- The ratio of the MLE's of the two dimensions is closer to 1 (1.537) for spam mails when compared to non-spam (2.95). This shows likely co-occurance of these two expressions in spam mails. Creation of an interation variable between the two dimensions could be a good predictor of spam mails.

(c) Implementation of Knn Classification and Plot



(d) Implementing Logistic Regression and Steepest ascent

```
Out[29]:
<matplotlib.text.Text at 0xa3e7cc0>
```

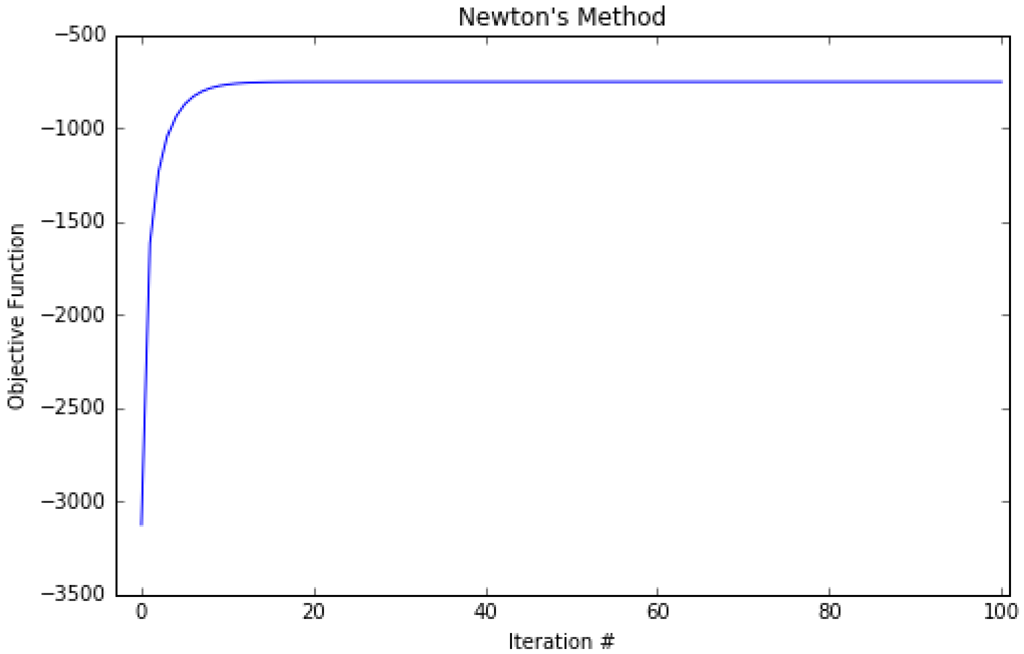


(e) Computing Accuracy using Newton's Method

```
91.398
```

Out[32]:

```
<matplotlib.text.Text at 0xbb63ef0>
```



Accuracy = 91.4 %