Problem Set 1

NAME: Harsh Bhate **GTID: 903424029**

Problem	**Points**
Problem 1	
Problem 2	
Problem 3	
Problem 4	
Total	

Problem 1

According to a Naïve Baye's classification, ***four*** *of the twelve* disputed article may be attributed to **Hamilton**. The remaining ***eight*** *of the twelve* articles may be attributed to **Madison**. The copy of the code used to classify the document is attached below.

Code

```
import numpy as np
import json
from sklearn.feature_extraction import text

x = open('fedpapers_split.txt').read()
papers = json.loads(x)

papersH = papers[0] # papers by Hamilton
papersM = papers[1] # papers by Madison
papersD = papers[2] # disputed papers

nH, nM, nD = len(papersH), len(papersM), len(papersD)

# This allows you to ignore certain common words in English
# You may want to experiment by choosing the second option or your own
# list of stop words, but be sure to keep 'HAMILTON' and 'MADISON' in
# this list at a minimum, as their names appear in the text of the papers
# and leaving them in could lead to unpredictable results
stop_words = text.ENGLISH_STOP_WORDS.union({'HAMILTON','MADISON'})
# stop_words = {'HAMILTON','MADISON'}

## Form bag of words model using words used at least 10 times
vectorizer = text.CountVectorizer(stop_words=stop_words,min_df=10)
X = vectorizer.fit_transform(papersH+papersM+papersD).toarray()

# Uncomment this line to see the full list of words remaining after filtering out
# stop words and words used less than min_df times
vectorizer.vocabulary_
d = len(vectorizer.get_feature_names())

# Split word counts into separate matrices
XH, XM, XD = X[:nH,:], X[nH:nH+nM,:], X[nH+nM:,:]
# Total number of words by Hamilton
totH = np.sum(XH)
# Total Number of Words by Madison
totM = np.sum(XM)

# Estimate probability of each word in vocabulary being used by Hamilton
fH = (XH.sum(axis=0)+1)/(totH+d)
# Estimate probability of each word in vocabulary being used by Madison
```

```python
fM = (XM.sum(axis=0)+1)/(totM+d)
# Compute ratio of these probabilities
fratio = (fH)/(fM)
# Compute prior probabilities
piH = nH/(nH + nM)
piM = nM/(nH + nM)

for xd in XD: # Iterate over disputed documents
    # Compute likelihood ratio for Naive Bayes model
    LR = np.prod(np.power(fratio,xd))*(piH/piM)
    if LR>0.5:
        print ('Hamilton')
    else:
        print ('Madison')
```

Problem 2

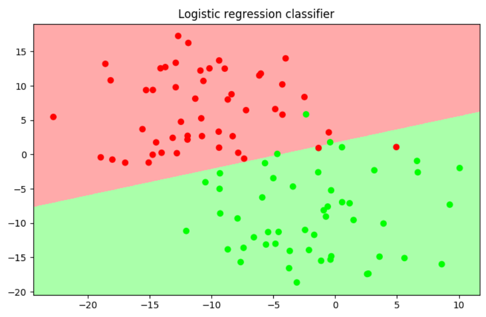
(a).

Learning Rate	Convergence Step
1	2
0.1	2
0.01	2
0.001	501
0.005	**188**
0.0001	502

A learning rate of $\alpha = 0.005$ corresponded to the best optimization with respect to convergence steps. However, a better result can be obtained with different learning policies. Their results are shown below:

Learning Policy	Convergence Step	Base Learning Rate	Parameters
Exponential	7	0.01	$\gamma = 0.1$
Inverse	223	0.01	$\gamma = 0.9$

The output of the logistic classifier is:



(b).

$$l(\theta) = \sum_{i=1}^{N}(y_i\theta^T\mathbf{x}_i - log(1 + e^{\theta^T\mathbf{x}_i})$$

The gradient is,

$$\nabla_\theta l(\theta) = \sum_{i=1}^{N}\mathbf{x}_i\big(y_i - \frac{1}{1+e^{-\theta^T\mathbf{x}_i}}\big)$$

Thus,

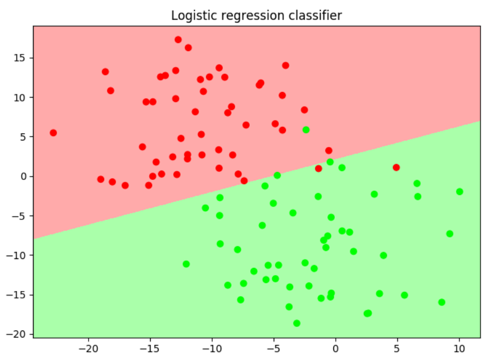
$$\begin{aligned}
\nabla^2_\theta l(\theta) &= \frac{\partial}{\partial\theta}\sum_{i=1}^{N}\mathbf{x}_i\Big(y_i - \frac{1}{1+e^{-\theta^T\mathbf{x}_i}}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\frac{\partial}{\partial\theta}\Big(\frac{1}{1+e^{-\theta^T\mathbf{x}_i}}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\frac{\partial}{\partial\theta}\Big(\frac{1}{1+e^{-\mathbf{x}_i^T\theta}}\Big)[\because A^TB = B^TA] \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\Big(\frac{-\mathbf{x}_i^Te^{-\theta^T\mathbf{x}_i}}{(1+e^{-\theta^T\mathbf{x}_i})^2}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\Big(\frac{-e^{-\theta^T\mathbf{x}_i}}{(1+e^{-\theta^T\mathbf{x}_i})^2}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\Big(\frac{-1-e^{-\theta^T\mathbf{x}_i}+1}{(1+e^{-\theta^T\mathbf{x}_i})^2}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\Big(\frac{1-(1+e^{-\theta^T\mathbf{x}_i})}{(1+e^{-\theta^T\mathbf{x}_i})^2}\Big) \\
&= -\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^T\frac{1}{(1+e^{-\theta^T\mathbf{x}_i})}\Big(1-\frac{1}{(1+e^{-\theta^T\mathbf{x}_i})}\Big)
\end{aligned}$$

(c). The iterations for convergence are:

Samples (N)	Convergence Steps
100	9
100,000	8

```
def newton_opt(theta, x, y, tol, maxiter):
    nll_vec = []
    nll_vec.append(neg_log_like(theta, x, y))
    nll_delta = 2.0*tol
    iter = 0
    while (nll_delta > tol) and (iter < maxiter):
        print (theta.shape)
        alpha = np.linalg.inv(log_hessian(theta, x))
        theta = theta - alpha@log_grad(theta, x, y)
        nll_vec.append(neg_log_like(theta, x, y))
        nll_delta = nll_vec[-2]-nll_vec[-1]
        iter += 1
return theta, np.array(nll_vec)
```

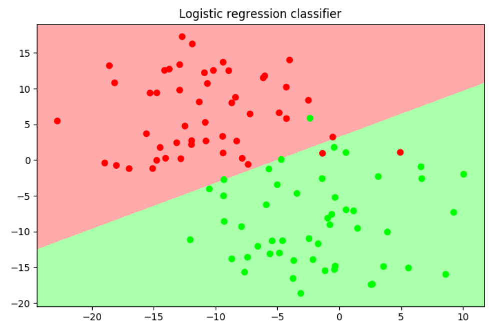
The output of the Logistic Classifier with 100 Samples is:



(d). The iterations for convergence are:

Learning Rate (α)	Convergence Steps
1	2
0.1	3
0.01	11
0.001	32
0.0001	**32**

The output of the logisitc classifier with a learning rate ($\alpha = 0.0001$) is:

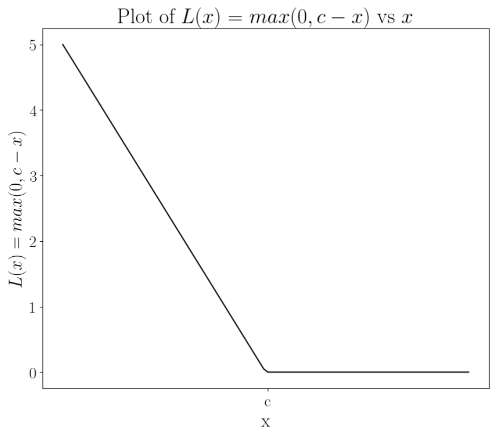


(e).

Parameter	**Gradient Descent**	**Newton Method**	**SGD**
Time (Seconds)	47.30	5.7	**0.096**
Learning Rate (α)	10^{-6}	Nil	10^{-6}
Optimization	Approximate Optimization	**Assured Optimization**	Approximate Optimization
Computation Cost	Moderate	High	**Low**
Best Use Case	Moderate Dataset	Small Dataset	Large Dataset

Problem 3

(a).



(b). The solution for Optimal Soft Margin Classifier can be found by solving the following:

$$\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||_2^2 + \frac{C}{N}\sum_{i=1}^{N}\xi_i \text{ subject to } \xi_i \geq 0; y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \tag{1}$$

Inspecting the problem, we want the value of error to be as low as possible. Thus, the constraints can be rewritten to be expressed as:

$$\begin{aligned} y_i(\mathbf{w}^T\mathbf{x}_i + b) &\geq 1 - \xi_i \\ \xi_i &\geq 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b); \xi_i \geq 0 \\ \xi_i &= 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b); \xi_i \geq 0 \text{ [Minimum Value]} \end{aligned}$$

Thus,

$$\xi_i = max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)) \tag{2}$$

Substituting (2) in (1), we get,

$$\boxed{\min_{\mathbf{w},b,\xi} \frac{1}{2}||\mathbf{w}||_2^2 + \frac{C}{N}\sum_{i=1}^{N} max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))}$$

Also,

$$\boxed{L(?) = L(y_i(\mathbf{w}^T\mathbf{x}_i + b)) = max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)); c = 1}$$

Problem 4

(a). Growth Function is the maximum number if dichotomies that can be generated by $\mathcal{H}$ on ant N points. In case of the hypothesis $h(x) = sign(x - a)$, the maximum number of dichotomies introduced is $(N + 1)$. In the case of the hypothesis $h(x) = -sign(x - a)$, the maximum number of dichotomies introduced is $(N + 1)$. It is so because in both these hypotheses, the line is split into points into the $(N+1)$ region. As we vary a, we get $(N+1)$ different hypothesis.

The hypotheses $h(x) = sign(x - a)$ and $h(x) = -sign(x - a)$ are dual of each other. *Therefore, the dichotomies induced by both these dichotomies is unique when a lies between the N points.* However, the edge cases induced by both these dichotomies are same and thus need not be counted again. Edge cases refers to the condition when a is less than all N points or when a is greater than all N points.

Thus, the growth function in this case is:

$$\boxed{m_{\mathcal{H}} = 2N}$$

(b). Akin to the previous problem, the set of hypothesis in this example are also dual of each other. Therefore, the maximum number of unique dichotomies induced in these example is decided by the two regions containing the end values of the interval. This results in $\binom{N+1}{2} + 1$ dichotomies.

However, when both these hypthoses are ensembled, only the dichotomies induced by the interval between the N points are unique to each of the hypothesis. The set of dichotomies induced by either of the interval from the left or the right are similar and need not be counted twice. The number of such repeated dichotomies is $2N$. Thus,

$$\begin{aligned} m_{\mathcal{H}} &= 2\Big(\binom{N+1}{2} + 1\Big) - 2N \\ &= 2\Big(\frac{1}{2}N^2 + \frac{1}{2}N + 1\Big) - 2N \\ &= N^2 + N + 2 - 2N \\ &= N^2 - N + 2 \end{aligned}$$

$$\boxed{m_{\mathcal{H}} = N^2 - N + 2}$$