UNIVERSITÉ DE BOURGOGNE

University of Bourgogne

MSCV

SSI-Homework II

Supervisor:
Dr.Desire Sidibe

Submitted By,
Majeed Hussain

Contents

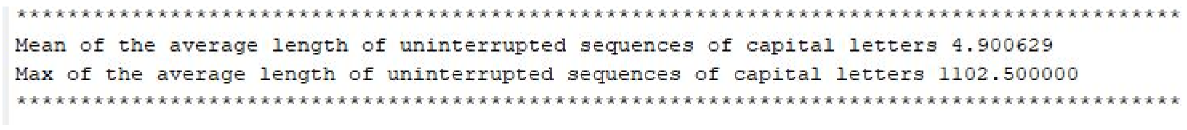
1 Logistic Regression

We want to build a classifier to filter spam emails. We were given training and test data sets of emails[1]. Each email has been processed and a set of 57 features were extracted.

1. 48 features, in [0; 100], giving the percentage of words in a given message which match a given word on a predefined list (called vocabulary).
2. 6 features, in [0; 100], giving the percentage of characters in the email that match a given character on the list.
3. Feature 55: the average length of an uninterrupted sequence of capital letters.
4. Feature 56: the length of the longest uninterrupted sequence of capital letters.
5. Feature 57: the sum of the lengths of uninterrupted sequence of capital letters.

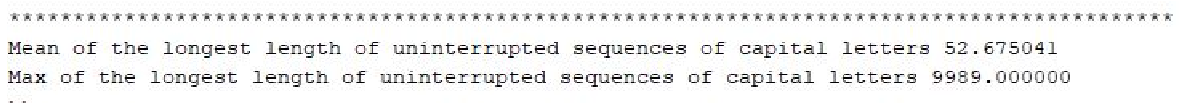
1.1 Tasks

1. What are the max and mean of the average length of uninterrupted sequences of capital letters in the training set?



```
**************************************************************************************
Mean of the average length of uninterrupted sequences of capital letters 4.900629
Max of the average length of uninterrupted sequences of capital letters 1102.500000
**************************************************************************************
```

2. What are the max and mean of the lengths of the longest uninterrupted sequences of capital letters in the training set?



```
**************************************************************************************
Mean of the longest length of uninterrupted sequences of capital letters 52.675041
Max of the longest length of uninterrupted sequences of capital letters 9989.000000
```

3. Before training a classifier, we can apply several preprocessing methods to this data. We will try the following ones:

1. Standardize the columns so they all have mean 0 and unit variance.

2. Transform the features using $\log\left(x_{ij}+0.1\right)$, i.e. add 0.1 to each feature for every example and take the log. We add a small number to avoid taking log of zero!
3. Binarize the features using $I\left(x_{ij}>0\right)$, i.e. make every feature vector a binary vector.

1.1.1 Preprocessing

As given we will perform Three preprocessing techniques and further fit a logistic regression model.Upon comparing all the preprocessing techniques we will choose the best one.

This preprocessing task is done by function *[z, fmean, fstd] = preprocessing(Data,type)*

1.1.2 First Pre-processing Technique

Here we Standardize the columns so that they will all have mean 0 and unit variance.For every feature we perform as following to standardize.

$f_i, f_i \leftarrow \frac{f_i-\overline{f}_i}{\sigma_{f_i}}$
By doing this each feature is now centered and standardized.

To perform this Pre-processing select the type = 1 in *preprocessing* function.

1.1.3 Second Pre-processing Technique

Transforming the features using $\log\left(x_{ij}+0.1\right)$,

To perform this Pre-processing select the type = 2 in *preprocessing* function.

1.1.4 Third Pre-processing Technique

Binarize the features using $I\left(x_{ij} > 0\right),$

To perform this Pre-processing select the type = 3 in *preprocessing* function.

1.1.5 Fitting the Logistic Regression Model

Below are the tasks to perform logistic regression:

1. We need to get the regularization parameter lambda
2. Cross-validation is performed by taking 80 percent of training data to train and 20 percent of training data to validate.
3. Perform optimization using cost function.
4. Prediction on test data.

The cost function is defined as follows:

$E(w) = -\sum_{n=1}^{N}\left\{y_n \log\left(\phi_n\right) + \left(1 - y_n\right)\log\left(1 - \phi_n\right)\right\} + \frac{\lambda}{2}\sum_{k=1}^{M} w_k^2$

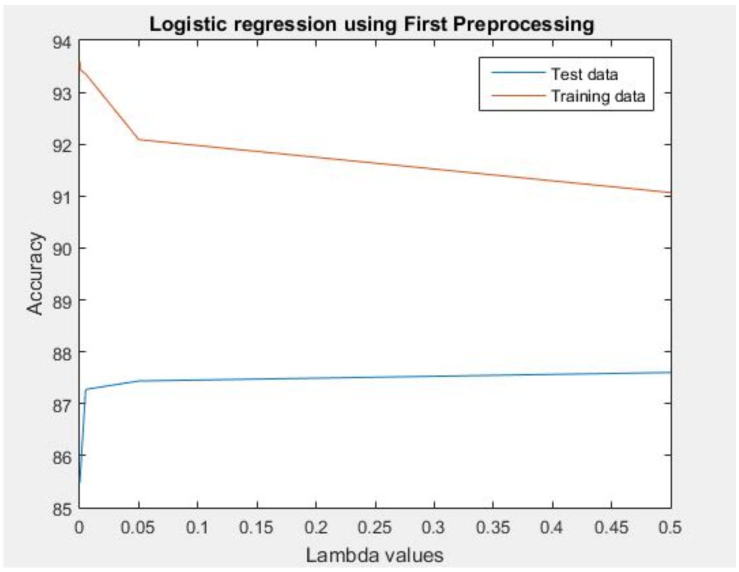
And the Prediction is done by using sigmoid function it has values between 0 and 1 in probabilities.
$\sigma(a) > 0.5 \rightarrow a$ belongs to class I
$\sigma(a) < 0.5 \rightarrow a$ belongs to class II

After we predict the validation labels and training labels, we compare with the true validation labels and train labels to get errors and acuuracy. later minimum error is chosen.

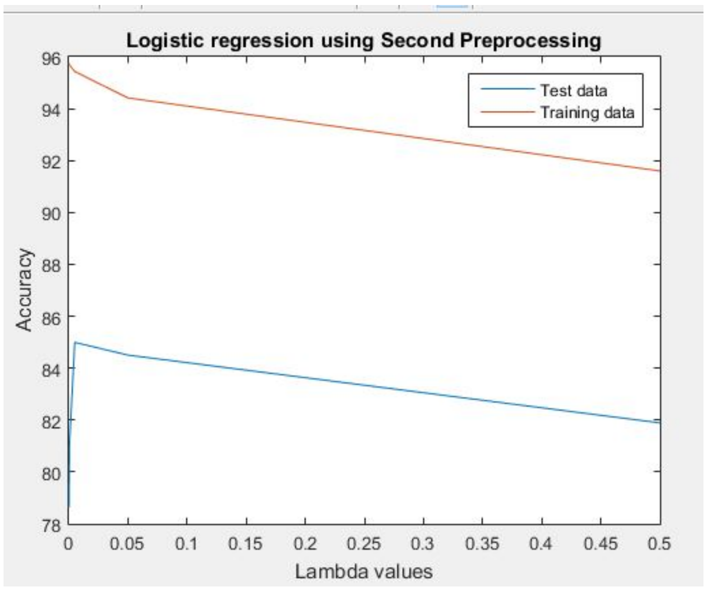
2 Results

:

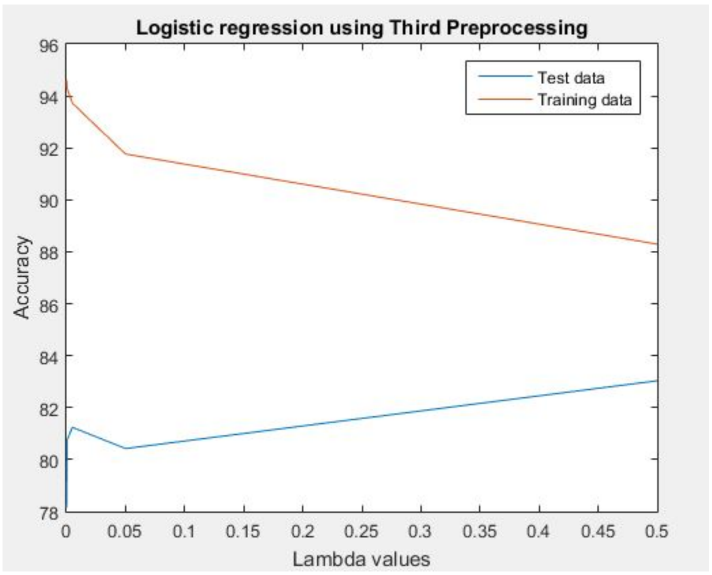
1. Using First Pre-processing Technique we get following outputs:



2. Using Second Pre-processing Technique we get following outputs:

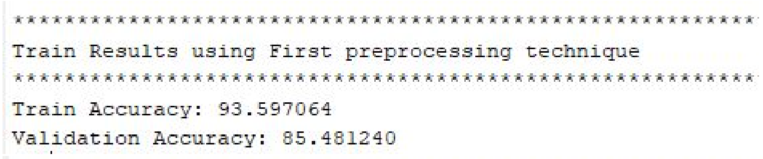


3. Using Third Pre-processing Technique we get following outputs:

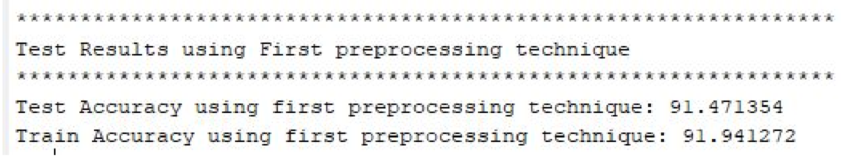


2.0.1 Accuracy

Fitting logistic regresiion model using First Pre-processing technique we get :

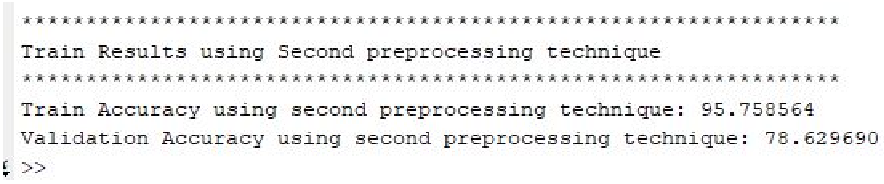


```
*********************************************************
Train Results using First preprocessing technique
*********************************************************
Train Accuracy: 93.597064
Validation Accuracy: 85.481240
```

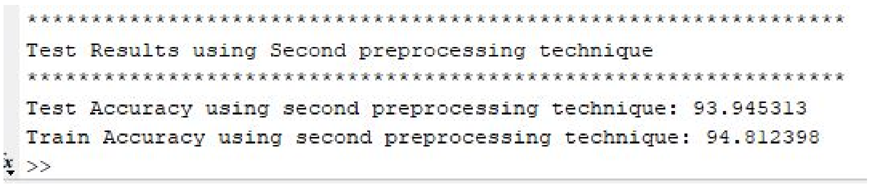


```
****************************************************************
Test Results using First preprocessing technique
****************************************************************
Test Accuracy using first preprocessing technique: 91.471354
Train Accuracy using first preprocessing technique: 91.941272
```

Fitting logistic regresiion model using Second Pre-processing technique we get :

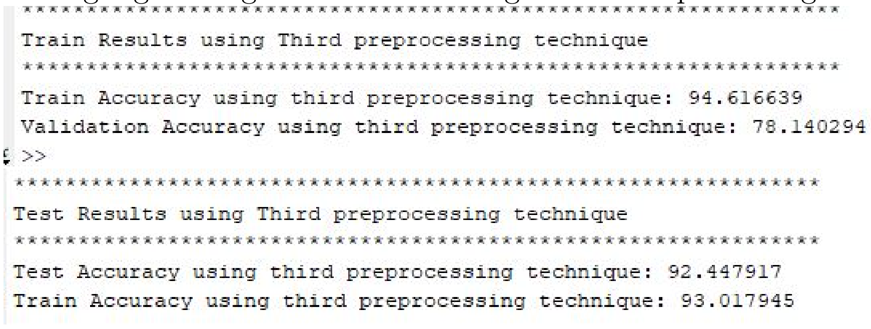


```
****************************************************************
Train Results using Second preprocessing technique
****************************************************************
Train Accuracy using second preprocessing technique: 95.758564
Validation Accuracy using second preprocessing technique: 78.629690
>>
```



```
****************************************************************
Test Results using Second preprocessing technique
****************************************************************
Test Accuracy using second preprocessing technique: 93.945313
Train Accuracy using second preprocessing technique: 94.812398
>>
```

Fitting logistic regresiion model using Third Pre-processing technique we get :



```
****************************************************************
Train Results using Third preprocessing technique
****************************************************************
Train Accuracy using third preprocessing technique: 94.616639
Validation Accuracy using third preprocessing technique: 78.140294
>>
****************************************************************
Test Results using Third preprocessing technique
****************************************************************
Test Accuracy using third preprocessing technique: 92.447917
Train Accuracy using third preprocessing technique: 93.017945
```

So we can say that Second Pre-processing technique is best strategy since it gives good results.

3 Naive Bayes

A naive Bayes classifier uses probability theory to classify data. Naive Bayes classifier algorithms make use of Bayes' theorem. The key insight of Bayes' theorem is that the probability of an event can be adjusted as new data is introduced.

What makes a naive Bayes classifier naive is its assumption that all attributes of a data point under consideration are independent of each other. A classifier sorting fruits into apples and oranges would know that apples are red, round and are a certain size, but would not assume all these things at once. Oranges are round too, after all.

A naive Bayes classifier is not a single algorithm, but a family of machine learning algorithms that make uses of statistical independence. These algorithms are relatively easy to write and run more efficiently than more complex Bayes algorithms.

The most popular application is spam filters. A spam filter looks at email messages for certain key words and puts them in a spam folder if they match.

Despite the name, the more data it gets, the more accurate a naive Bayes classifier becomes, such as from a user flagging email messages in an inbox for spam. Naive

Bayes is implemented in *naive.m* file
Naive Bayes is computed by :

$p\left(C_1|X\right) \propto p\left(X|C_1\right) p\left(C_1\right)$

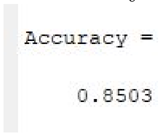
where $p(C_1)$ is $p\left(C_1\right) = \frac{1}{N}\sum_{n=1}^{N} t_{nC1} = \frac{N_1}{N}$

and $p\left(X|C_1\right) = p\left(X_1, X_2, \ldots, X_m|C_1\right)$
$p\left(X_i|C_1\right)$ can be a normal distribution $\mathcal{N}\left(X_i;\mu_1,\sigma_1\right) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-1}{2}\left(\frac{\left(X_i-\mu_1\right)^2}{\sigma^2}\right)}$

if $p\left(C_1|X\right) \geq p\left(C_2|X\right)$ assign class I else assign class II

3.1 Accuracy

: Using the First Pre-processing technique 85 percent accuracy is obtained using Naive Bayes



```
Accuracy =

    0.8503
```