HARIPRIYAA U MANIAN(hum160030)

MACHINE LEARNING ASSIGNMENT 2

NAÏVE BAYES AND LOGISTIC REGRESSION FOR TEXT CLASSIFICATION

Language Used: Python

Command line Arguments:

TrainingHamPath TrainingSpamPath TestHamPath TestSpamPath LearningRate Lamda StopWordsTextPath

For eg:

training_ham_folder="C:/Users/Priyaa/Desktop/Machine Learning/ML_Assignment2/2/hw2_train/train/ham"
training_spam_folder="C:/Users/Priyaa/Desktop/Machine Learning/ML_Assignment2/2/hw2_train/train/ham"
test_ham_folder="C:/Users/Priyaa/Desktop/Machine Learning/ML_Assignment2/2/hw2_test/test/ham"
test_spam_folder="C:/Users/Priyaa/Desktop/Machine Learning/ML_Assignment2/2/hw2_test/test/spam"
Learning_Rate=0.01
Lamda=-0.001
stop_words_file="C:/Users/Priyaa/Desktop/Machine Learning/ML_Assignment2/STOP_WORDS.txt"

List of python files:

Main.py

Gets the command line arguments and has list of function calls to find accuracy and prints accuracy

ExtractTrain.py

Code to extract Training data set – ham and spam, converting into a list and putting the contents as Bag of Words – matrix and lists having all the counts, positions needed for NB and LR calculation

ExtractTest.py

Code to extract Test data set – ham and spam, converting into a list and putting the contents as Bag of Words – matrix and lists having all the counts, positions needed for NB and LR calculation

The following files does the same as above but it excluded

ExtractTrainWithoutStopWord.py

Excluding Stop Words Code to extract training data set – ham and spam, converting into a list and putting the contents as Bag of Words – matrix and lists having all the counts, positions needed for NB and LR calculation

ExtractTestWithoutStopWord.py

Excluding Stop Words Code to extract Test data set – ham and spam, converting into a list and putting the contents as Bag of Words – matrix and lists having all the counts, positions needed for NB and LR calculation

NaiveBayes.py

Code for Naïve Bayes accuracy calculation

LogisticRegression.py

Code for Logistic Regression accuracy calculation

ANALYSIS ON THE ACCURACY OBTAINED:

1.Multinomial Naïve Bayes Algorithm:

Here, Naïve Bayes is applied in two cases – with and without Stop Words. The algorithm uses add-one Laplace smoothing as mentioned in the pdf. All the calculations are done in the log-scale to avoid underflow.

First the training is done using the training data set. While training, the basic punctuation marks such as {'-', ':', ',', '.', '/'} are considered as unwanted words and they are removed and while testing using the testing data set, the following formula is used for classification

$$c_{\text{map}} = \arg\max_{c \in \mathbf{C}} [\log \hat{P}(c) + \sum_{1 \le k \le n_d} \log \hat{P}(t_k|c)].$$

Finally the accuracy is calculated. The same procedure is repeated again but excluding the stop words mentioned in http://www.ranks.nl/stopwords. Now the accuracy is calculated.

It is observed that accuracy for Naïve Bayes with stop words is lower and once we remove the stop words, the accuracy increases.

Naïve Bayes Accuracy	Including Stop words	Excluding Stop Words
	77.82426778242679	89.3305439330544

2. Logistic regression:

MCAP Logistic Regression algorithm with L2 regularization is used with varying values of

--Learning Rate
--Lamda
--Hard limit on iterations.

Here, the code learns from the training set and reports accuracy on the test set for both cases – with and without stop words.

$$P(Y=0|X) = \frac{1}{1+\exp(w_0+\sum_{i=1}^{n} w_i X_i)}$$

$$P(Y=1|X) = 1 - P(Y=0|X)$$

Accuracy differs for different values of learning rate, lamda and iteration values

No of Iterations	Learning Rate	Regularization	Accuracy	
			Including Stop Words	Excluding Stop Words
5	0.01	0.001	73.01255230125523	76.98744769874477
10	0.01	0.001	73.22175732217573	76.98744769874477

15	0.09	0.01	70.50209205020921	76.56903765690377
30	0.07	0.06	73.22175732217573	76.56903765690377
40	0.1	0.2	74.89539748953975	78.03347280334728

Inferences Made:

The accuracy is observed to be high when stop words are excluded and the accuracy depends on the values of Learning Rate and Regularization, that it increases with increasing learning rate and lamda values.