CS 6375.001 Machine Learning

Assignment 2 – Report

Submitted by :
Vidya Sri Mani
vxm163230

03/26/2017

Naïve Bayes and Logistic Regression

Program Name: assignment2.java
Programming language: Java

The program is divided into two parts:

- Naïve Bayes classification
- Logistic Regression Classification with L2 regularization

Part1
Naïve Bayes

The first part of the program classifies messages as either spam of Ham. As part of the training process, the folder location to the spam and ham folders are given to the program.
For each word in the training message, the number of distinct words is stored. The frequency of these words for the training data is calculated. Depending on this frequency, their individual probabilities are calculated, for the ham class as well as the ham class.

When a new message is seen, each of their words are separated, and their probabilities for the spam class and ham class is calculated.
If the combined probability of the ham class is greater than that of the spam class, it is categorized as ham, else it is categorized as spam.
Log is used while calculating the combine probability, to avoid underflow.
Bayes' Theorem:
$P(h|d) = (P(d|h) * P(h)) / P(d)$
Where,
$P(h|d)$: probability of hypothesis h given the data d.
$P(d|h)$: probability of data d given that the hypothesis h was spam.

P(h) : probability of hypothesis h being true (regardless of the data).
P(d) : probability of the data (regardless of the hypothesis).

Maximum a posteriori (MAP) hypothesis.
This can be written as:
MAP(h) = max((P(d|h) * P(h)) / P(d))

Part2

MCAP Logistic Regression with L2 regularization

Chosen value of Lambda: 0.001
Restricted number of iterations: 500

Gradient ascent for learning weights , with an initial weight set as 1 for words.
Logistic Regression algorithm learns the weight for each word in the message. When a new word is encountered, it finds the word in the list and obtains the sum of its weight.

P(Y=spam| X) = exp(w*x)/(1+exp(w*x))
Where w → weight of word
X → 1 or 0, with 1 in the position correspond to each word.

The resulting probability lies between 0 and 1.
The probability is calculated as
$P(Y=0|X,W) = 1/(1+\exp(w0 + \sum_i w_i x_i))$
The conditional likelihood is a concave function. Gradient ascent is used to find the optimum value.

Working of the algorithm

The probability for each message is calculated. This is compared with a threshold value. If it exceeds the threshold value, it is marked as spam, otherwise it is classified as ham.

The learning rate chosen is 0.001.
The programs allows for a change for this value.

An appropriate value for lambda needs to be chosen to ensure that the direction of the gradient is not too large or small.

When the value of lambda is too large, the progress towards obtaining the optimal value is too fast, and hence the optimal value can be missed easily.
When the value of lambda is too small, smaller steps are taken to reach the optimal value. This might provide a better accuracy, however the progress is painfully slow. To avoid the program running for an extended time, the number of iterations in the program are restricted.

This program by default uses 500 iterations. When the number of iterations are increased, the program can learn better, and hence can improve the accuracy.

The data for training and testing is taken from Elearning , CS 6375.001 Machine Learning, Assignment2

Stop Words

Stop words are general words are equally likely to be present in both classes (spam and ham). Ignoring these words can provide a better accuracy.
For Instance, words like for, of, etc., don't contribute heavily to determine the class of the message, and hence can be omitted.

The list of start words is taken from
http://www.ranks.nl/stopwords

Both parts of the program, The naïve Bayes as well as the Logistic regression is run once with the use of stop words, and once without the use of stop words.

Accuracy of the Naïve Bayes

```
Naive Bayes with stop words
----------------------------------------
Training...
probabilties calculated
Testing...
For Test Ham class files
Total Number of Ham classified Files:348
Total Number of Predicted Ham classified Files:347 out of 348
Total Number of Predicted Spam classified Files:1 out of 348
----------------------------------------
For Test Spam class files
Total Number of Spam classified Files:130
Total Number of Predicted Ham classified Files:54 out of 130
Total Number of Predicted Spam classified Files:76 out of 130
----------------------------------------
Accuracy of Naive Bayes : 88.49372384937239%
================================================================


Naive Bayes without stop words
----------------------------------------
Training...
probabilties calculated
Testing...
For Test Ham class files
Total Number of Ham classified Files:348
Total Number of Predicted Ham classified Files:344 out of 348
Total Number of Predicted Spam classified Files:4 out of 348
----------------------------------------
For Test Spam class files
Total Number of Spam classified Files:130
Total Number of Predicted Ham classified Files:39 out of 130
Total Number of Predicted Spam classified Files:91 out of 130
----------------------------------------
Accuracy of Naive Bayes : 91.0041841004184%
================================================================
```

Accuracy of the Logistic regression with lambda = 0.001 and number of iterations = 150

```
LOGISTIC REGRESSION

Logistic Regression, Number of iterations = 500, value of lamba = 0.001
Enter y/Y to change values, otherwise enter n/N
y
Enter new value for lambda :
0.001
Enter new value for number of iterations :
150


Logistic Regresion with stop words
----------------------------------------
Number of correctly classsified messages : 409.0
Total Number of files : 478.0
Accuracy : 85.56485355648536%


Logistic Regresion without stop words
----------------------------------------
Number of correctly classsified messages : 415.0
Total Number of files : 478.0
Accuracy : 86.82008368200836%
```