CSE537 Artificial Intelligence

Assignment-5 Report

Project SpamFilter

Authors:

Vishal Nayak: 109892702: vnayak@cs.stonybrook.edu

Ashish Chaudhary: 109770154: ashchaudhary@cs.stonybrook.edu

Contents

1. Intro and details

1.1. Compilation, machine details and commands

Compiler: ***python 2.7.6***

Machine: Windows 8 64-bit

Command for implementation 1: **python spamfilter.py**

Command for implementation 2: **python bayes.py train test**

1.2. Goals and Approaches

Spam Filter Project:

Classifying the emails as spam or not-spam messages. Approach is to construct a classifier using Naïve Bayes algorithm and predicting if the email is a spam or not.

Apparently, for learning purposes, both of us implemented this project and both of us have good results.

2. SpamFilter

Implementation 1:

While parsing the training samples, dictionaries of spam words and ham words are populated. Using these, the conditional probabilities of each word in spam and ham dictionaries are calculated. After this, test samples are parsed and probability scores for each test sample for being spam or a ham is calculated. Classification is done based on the dominant probability.

Implementation 2:

Train and test data are passed as arguments to the script. Dictionaries are used to store the computed conditional probabilities, which represents the model of the train data. The only logical difference is in the calculation of conditional probabilities. As instructed, different smoothing parameters were tried out and the performance has indeed improved.

3. Smoothing in SpamFilter

Smoothing is done to avoid situations where-in a single conditional probability making the entire score of sample belonging to category as zero. Hence a small value is assumed for each word to be present. This prevents the value being computed as zero.

The submitted code uses both Laplace smoothing and Lidstone smoothing.

The differences in the results obtained are minimal. However, the observation is that, lesser the value of the smoothing factor (less than 1), greater is the accuracy.

4. Results

4.1. SpamFilter Implementation 1

Accuracy: 91.3 %
***Spam count in test data** = 580*
***Ham count in test data** = 420*
***Spam count in train data** = 5163*
***Ham count in train data** = 3837*

4.2. SpamFilter Implementation 2

***Smoothing:** Lidstone Smoothing*

***Alpha** = 0.5*

***Mismatches:** 116*

***Accuracy:** 88.4*

***Running Time:** 1.46166651469*

***Smoothing:** Laplace Smoothing*

***Alpha** = 1*

***Mismatches:** 117*

***Accuracy:** 88.3*

***Running Time:** 1.4822570477*

***Observation:** As the alpha value decreases, the accuracy increases.*

5. Statistics

5.1. Accuracy Percentages