Elias Posen November 7, 2017

Text Classification

(PLEASE USE PYTHON 2.7)

Version 1: raw Naïve Bayes

Findings:

In general, the raw naïve Bayes (NB) classifier does not accurately guess the correct category for the test data. Omitting the exception of the first category in the output table (rec.motorcycles) the raw NB classifier estimates the correct category 7.3% of the time (averaged over 3 tests). Over these same 3 tests rec.motorocycles averaged a correct classification of 99.2% of the time. The reason for this inflated correctness is the same reason for the low rates of accuracy that plagues the rest of the categories.

Source of Inaccuracy:

The main issue with this approach is that if the test data contains a word that was not in the training data, then the probability of that word being in any of the categories is zero (according to the information gathered during training). Thus, when the probabilities are passed to argmax, they are just a list of zeros and therefore argmax will simply return the first argument in the list by default. This is the reason for the inflated accuracy of rec.motorcycles, for it is chosen for every test "sentence" that contains a new word. Over the 3 test runs the raw NB classifier guessed rec.motorcylces an average of 7044 times, whereas all the other categories were chosen by the raw NB classifier less than 40 times each. This indicates that there are many sentences within the test data that contain words that have not been seen by the categories in the training set and therefore the raw NB classifier will default to choosing rec.motorcycles. This represents and inherent flaw to the raw probability approach.

Version 2: m-estimate Naïve Bayes

Findings:

The m-estimate NB classifier improved on the raw NB classifier. It correctly predicted the category of a test "sentence" 80.3% of the time (over ten times as accurate). The primary reason for this increase in accuracy is that the issue of zeros is dealt with. By this I mean that if a word in the test set has not been seen by classifier *c* in the training set then the m-estimate NB assigns a very small probability to P(word|*c*), instead of simply marking it as a zero. Thus the problem of passing argmax a list of zeros (whereby it returns the rec.motorcycles) is curtailed.

Elias Posen November 7, 2017

Version 3: tf-dif Naïve Bayes

Findings:

The tf-idf NB classifier is intended to be an improvement on the m-estimate NB classifier. It does this by weighting each probability based on the occurrences of the current word in each of the categories. The intuition goes as follows: if the word occurs in many of the categories than weight it by a small number to reduce its significance. If, on the other hand, the current word occurs in a small amount of the categories than weight it by a large number to increase its significance in calculating the overall probability that the sentence is in a given category. It correctly predicted the category of a test "sentence" 82% of the time, over three separate tests. This increases the accuracy of the m-estimate because it is weighting the probability of the word given the category based on the word's uniqueness across the categories. Thus, the tf-dif version of the NB classifier works the best for this classification.