Machine Learning – Assignment 3

Team Members

Ajaykrishna Karthikeyan (2015A7PS0044H)
Nikhil Iyer (2015A7PS0139H)
Anirudh Srinivasan (2015A7PS0382H)

Overview of Procedure

The .feat files included tokenized bag of words features which is the primary source of training and testing data for the classifier. The imdb.vocab file is used to obtain the word corresponding to a given index whenever needed.

The /train/labeledBow.feat file is used to train the classifier. As the file is read, the frequency of occurrence of each word is stored along with the sentiment of the review it is present in. When binarization is used, multiple occurrences of a word in the same review is counted only once. Additionally if the given word is a stop word, then the occurrence of the word (if any) is ignored.

With the frequency of words corresponding to a particular sentiment know, the next step is to calculate for each word, the conditional probability of occurrence of the word given a particular sentiment. This can be computed as follows:

$$P(w_k/v_j)=\frac{(n_k+1)}{(n+s)}$$

where w_k is the word, v_j is the target value, n is the total frequency of all words in all reviews whose target value is v_j , n_k is the total frequency of w_k in all reviews whose target value is v_j and s is the size of the vocabulary.

Now, the classfier is ready to estimate target values for unseen instances (in this case reviews). The file used for testing is /test/labeledBow.feat. The estimated target value is:

$$\underset{v_j \in V}{argmax}\, P(v_j) + \sum_{i \in positions} \log\left(P(a_i/v_j)\right)$$

where $positions$ is all word positions in the unseen review.

The reason for using logarithm is because conditional probability of each is word is itself a small value and multiplying it with other small probabilities may result in floating point underflow.

Results

Method	Accuracy	Precision	Recall	F1 Measure
Without removing stopwords and without binarization	81.3600%	0.859040	0.750320	0.801008
Without removing stopwords and with binarization	82.9920%	0.872269	0.773040	0.819662
After removing stopwords and without binarization	82.6360%	0.865645	0.772640	0.81653
After removing stopwords and with binarization	83.7920%	0.872290	0.791760	0.830076

```
assignment@ml:$ ./a.out

Without removing stopwords
``````````````````````````

Without binarization
Accuracy: 81.360000%
Precision: 0.859040
Recall: 0.750320
F1 Measure: 0.801008
Time taken: 0.9101 seconds
10 most informative words:
edie antwone din goldsworthy gunga gypo yokai paulie visconti flavia

With binarization
Accuracy: 82.992000%
Precision: 0.872269
Recall: 0.773040
F1 Measure: 0.819662
Time taken: 0.8912 seconds
10 most informative words:
edie mcintire antwone tsui gunga din visconti goldsworthy quibble joss

After removing stopwords
````````````````````````

Without binarization
Accuracy: 82.636000%
Precision: 0.865645
Recall: 0.772640
F1 Measure: 0.816503
Time taken: 2.0186 seconds
10 most informative words:
edie antwone din goldsworthy gunga gypo yokai paulie visconti flavia

With binarization
Accuracy: 83.792000%
Precision: 0.872290
Recall: 0.791760
F1 Measure: 0.830076
Time taken: 2.0183 seconds
10 most informative words:
edie mcintire antwone tsui gunga din visconti goldsworthy quibble joss
```

## Reasoning for Results

After removing stopwords, an increase is seen in all the measures. All the sentences would contain stopwords, thus they would not help in distinguishing between classes. The query that a user would input would not contain stopwords, and thus the classifier would not perform better if they are present, hence they are prefered to be removed to reduce computation.

A very small improvement is observed for binarization. This could be attributed to the sentences in the dataset being short and not benefiting due to multiple occurances of words.

## Most Informative Features

It so happens that some features tell more about what the target value of a particular instance is than others, in this case the former is said to be more informative than the latter.

In the present problem, the most informative features are those words that give more information about what the sentiment of the review is, than the others. The most informative word is calculated as:

$$\underset{w \in vocabulary}{argmax} \frac{P(w/+ve\ sentiment)}{P(w/-ve\ sentiment)}$$

Instead if we need the 5 most informative words, we select the top 5 instead of just the one which has the maximum $\frac{P(w/+ve\ sentiment)}{P(w/-ve\ sentiment)}$ value

| Method | 10 most informative features (in decreasing order) |
|---|---|
| Without removing stopwords and without binarization | edie, antwone din goldsworthy gunga gypo yokai paulie visconti flavia |
| Without removing stopwords and with binarization | edie mcintire antwone tsui gunga din visconti goldsworthy quibble joss |
| After removing stopwords and without binarization | edie antwone din goldsworthy gunga gypo yokai paulie visconti flavia |
| After removing stopwords and with binarization | edie mcintire antwone tsui gunga din visconti goldsworthy quibble joss |