Lab Sentiment Analysis

March 29, 2018

Your goal for this homework is to perform Sentiment Analysis: classifying movie reviews as positive or negative. Recall from lecture that sentiment analysis can be used to extract people's opinions about all sorts of things (congressional debates, presidential speeches, reviews, blogs) and at many levels of granularity (the sentence, the paragraph, the entire document). Our goal in this task is to look at an entire movie review and classify it as positive or negative.

1 Algorithm

You will be using **Naïve Bayes**[1], and Laplace smoothing. Your classifier will use words as features, add the logprob scores for each token, and make a binary decision between positive and negative. You will also explore the effects of stopword filtering. This means removing common words like "the", "a" and "it" from your train and test sets. We have provided a stop list with the starter code in the file:

```
sentiment/data/english.stop
```

2 Assignment

Train a **Naïve Bayes** classifier on the **imdb1** data set provided with the starter code. This is the actual Internet Movie Database review data used in the original Pang and Lee paper[2]. The starter code comes already set up for 10-fold cross-validation training and testing on this data. Recall that cross-validation involves dividing the data into several sections (10 in this case), then training and testing the classifier repeatedly, with a different section as the held-out test set each time. Your final accuracy is the average of the 10 runs. When using a movie review for training, you use the fact that it is positive or negative (the hand-labeled "true class") to help compute the correct statistics. But when the same review is used for testing, you only use this label to compute your accuracy. The data comes with the cross-validation sections; they are defined in the file:

[1] the pseudocode in Manning, Raghavan, and Schutze (page 241 in the paper, offline edition; page 260 in the "pdf for printing" or "pdf for online viewing" version that's online

[2] **Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan**. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7986.

```
sentiment/data/poldata.README.2.0
```

Your first task is to implement the classifier training and testing code and evaluate them using the cross-validation mechanism. Next, evaluate your model again with the stop words removed. Does this approach affect average accuracy (for the current given data set)?

3 Evaluation

Your classifier will be evaluated on two different data sets—both **imdb1** mentioned above and a second held-out test set—and each of these is then in turn evaluated twice, once with and once without invoking stopword filtering.

4 Running the code

Execute

```
cd python
python NaiveBayes.py ../data/imdb1
```

This will train the language models for each cross-validation and output their performance.

Adding a `-f` flag...invokes the stopword filering.

```
python NaiveBayes.py -f ../data/imdb1
```

If you're curious how your classifier performs on separate training and test data sets, you can specify a second directory, in which case the program should train on the entirety of the first set (i.e., without cross-validation) then classify the entire held-out second set.

```
python NaiveBayes.py (-f) train test
```