NLP Assignment 1 : Report on Text Classification With Perceptron

180128022

February 2019

1 Introduction

We have implemented the Perceptron Algorithm to classify texts, i.e., positive and negative reviews. The program is a little slower because of the usage of Deep Copy in order to address the problem of referencing in Python.
The functions in use are:
1. **set_seed**: Setting seed
2. **ngram_traintest_pos** and **ngram_traintest_neg**: Operations on positive and negative data sets respectively
3. **ngrams_generation**: Pre-processing activities
4. **trainertester_adder**: Summing of data
5. **weight_set**: Setting weights
6. **standard_trainer**, **shuffled_standard_trainer** and **update_trainer**: Implementation of Perceptron under different settings like zero-weights, multiple passes and shuffling
7. **prediction**: Retrieving accuracy
8. **average_weights**: Averaging weights for each class
9. **plotter**: Plotting graphs
10. **top_ten_values**: Finding out top 10 features

2 Answers

epochs = 15, seed = 180128022
For Unigram Implementation :

- The standard binary perceptron accuracy is **50%**.
- Randomising the order of training instances, the accuracy is **72.25%**.
- Multiple passes over the training instances give the following error graph with an average accuracy of **83.25%**:

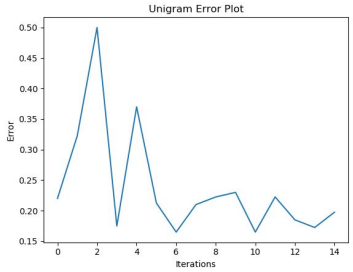

Figure 1: Learning Rate for Unigram - Error Graph for First 15 Iterations

- Beyond bag-of-words, the two feature type implementations are Bigram and Trigram with their respective graphs.

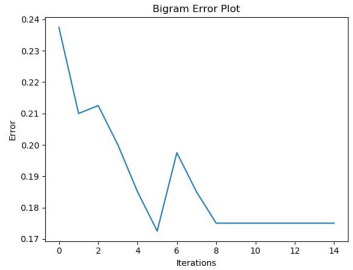

Figure 2: Learning Rate for Bigram - Error Graph for First 15 Iterations

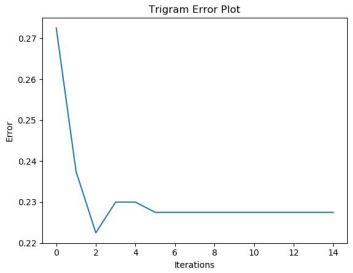

Figure 3: Learning Rate for Trigram - Error Graph for First 15 Iterations

- The average accuracy for Bigram is **82.25%**, while that of Trigram is **77.25%**, which means that the accuracy decreases with dimension, and reaches a constant level.
- The top 10 most positively-weighted features of each class with positive words like 'great', 'best', etc, and negative class like 'bad', 'worst', etc, give us clear classification where we can say that the training data generalises well on the test data. For laptop or film reviews, these features would not generalise well because there are many film-dependent words (for example - *pulp fiction* for bigram in the positive list) as well as objective words in the top ten list. The better features could be adjectives.