Coursework 2 - Sentiment Analysis

John Georgousis
ig441@bath.ac.uk

Emtiaz Samad
ems88@bath.ac.uk

Adam Rasool
aar73@bath.ac.uk

1 Introduction

In this project we provide two systems for extracting sentiment from pieces of natural language text. The first method implemented uses a support vector machine (SVM) to extract sentiment from film reviews originally retrieved from IMDB. The second methodology employed here makes use of a boosting algorithm for binary sentiment classification of the same dataset.

2 SVM Classification

2.1 Kernel Analysis

In producing the SVM used in this task we experimented with 5 different kernels, comparing their performance and ultimately choosing the one that produced the best results. The five kernels tested are listed below

Linear

$$k(x, y) = x^T y + c \tag{1}$$

Polynomial

$$k(x, y) = (\alpha x^T y + c)^d \tag{2}$$

Gaussian

$$k(x, y) = \exp(-\frac{\|x - y\|^2}{2\sigma^2}) \tag{3}$$

Laplacian

$$k(x, y) = (-\frac{\|x - y\|}{\sigma}) \tag{4}$$

Logarithmic

$$k(x, y) = -\log(\|x - y\|^d + 1) \tag{5}$$

Each kernel's performance was investigated in order to determine which of the 5 produced the best results for the task at hand.

As seen in the figure, the polynomial kernel produced the best results in our testing and as such was chosen for the implementation of this task.

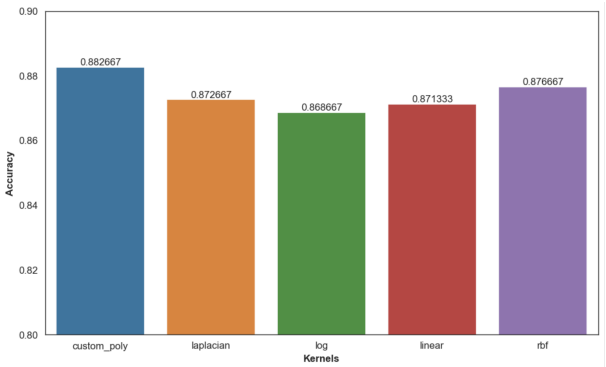

Figure 1: A graph showing the performance of each kernel on the test dataset

2.2 Cross Validation for Hyper Parameter Selection

Once it was determined that the polynomial kernel provided the best performance, we proceeded to carry out a gridsearch in order to tune our model's hyper parameters to further optimise the model's performance. Through this process, it was determined that the optimal coefficient value was 0.444 and the optimal number of degrees was 3.

2.3 Analysis of Performance and Mis-classifications

Our model showed promising results with an accuracy of 88% on the testing dataset. An analysis was carried out on the mis-classified reviews in an attempt to determine the cause of their mis-classification.

In viewing the wrongly classified reviews a common theme established was positive reviews that contained a large number of negative words and vice versa. As the model only processes the tokenised reviews with stop words removed, it is expected that the model will fail to correctly predict sentiments of reviews that conatin a balanced share of words associated with positive and negative sentiment. Further, misclassifica-

tions may also occur due to processing including punctuation, removing words and word order. Punctuation can sometimes change the meaning of a sentence. Likewise, removing words such as 'not' and 'don't' can actually reverse the meaning of a phrase in the removal of the negation. Word order is also important to consider as changing the order of words in a sentence can change its meaning. As a final point to consider, when manually analysing the dataset it was found that among the wrongly classified reviews there are reviews which themselves are not easily interpreted by a human reader most commonly due to poor grammatical structure. As such, it is unsurprising that the model would fail to correctly discriminate these reviews as they do not follow a pattern for declaring sentiment which can be easily learned from the majority of other reviews in the dataset.

3 Boosting Algorithm Classification

In this task the SAAME boosting algorithm was used to determine review sentiment.

3.1 Data Preprocessing

For the preparation of our data we converted each review in the dataset into a so called 'bag of words'. Word stems were removed from the data in order to simplify words which are variations of one another. Further, because of the formatting of the data within the set it was necessary to parse over each entry to remove HTML markup text.
The entries within the dataset were then vectorised. To achieve this, both the tfidf vectoriser and the count vectoriser were tested. The former was found to give better results when making predictions and so this was the class chosen. The reviews within the dataset were then shuffled in order to ensure that the ordering of data would not be a source of bias.

3.2 Parameter Analysis

As with the creation of our SVM for the previous task, we arrived at the optimal parameter values for our boosting algorithm through testing various values in order to determine which provided us with the best performance. The parameter that was tuned was the number of decision tree classifiers (the weak learners).
Using 800 training examples with K-Fold cross validation, the accuracy of the model was measured with classifiers ranging from 10 to 100 in steps of 10. The resulting accuracy values are shown in the plot below:

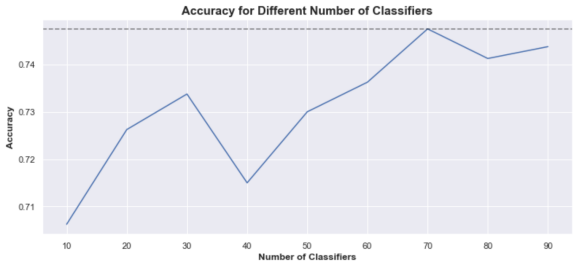

Figure 2: A graph showing the performance of each kernel on the test dataset

As the number of DT classifiers increased, the performance of the model improved.

3.3 Analysis of Performance and Mis-classification

As with the SVM model, the classification of reviews in this task was negatively impacted when the sentiment associated with particular words within a reviews 'bag' was not concurrent with the sentiment of the overall review itself. Further in instances where positive or negative words were negated by the context which preceded them eg., 'not good', 'I didn't think it was bad'. The classifier failed to recognise the effect of context on the sentiment and relevant reviews were misclassified.