Assignment 2

Shivakanth Thudi

11/13/2016

Analysis

Baseline model

The baseline model had as the features the rates of the English "function words" as specified in the assignment; some examples are "I", "the", "and", "to", and "you". The model also included the rates of the following punctuation symbols in the tweet: ".", ",", and "!".

Once the feature set was created, the data was split into a training set and test set. The model was built on the training set using three different algorithms - Logistic Regression, Linear Discriminant Analysis and k Nearest Neighbours. The model was tested on the training set and the Misclassification Rate was used to compare different models.

Part 1 - Feature Addition and Motivation

I wanted to use a bag-of-words classifier, so I built a corpus using all the tweets available. After tokenizing each tweet using a tweet-specific tokenizer that captures emoticons, I used a stemmer and reduced words to their lexical roots. Emoticons were preserved as is. I then built a corpus on these roots and emoticons, and established them as input variables for the classifiers.

I **added these features alongside the original features** which included the rates of the function words and the punctuation symbols.

With the bag-of-words linear classifier, we **simplify the tweets to a multi-set of term frequencies**. A tweet **sentiment tag will depend on what words appear in the tweet**, but will discard grammar or word order while keeping multiplicity.

Part 2 - Classifier Performance

With the baseline model, we observed that Linear Discriminant Analysis performed the best among all three algorithms.

We got the following misclassification rates:

Table 1: Machine Learning Algorithms and Misclassification Rates

Model	Testing Set Misclassification Rate
Logistic Regression	0.4107
Linear Discriminant Analysis	0.4083
k-Nearest Neighbors	0.46 with 3 neighbors

Part 3 - Model Performance after Corpus and Bag-of-Words Feature Additions

I extracted all the tokens and reduced them to their lexical roots, and established them as input variables for the classifiers. This resulted in a feature set that, apart from the original 23 features, included around 24,000 features. However, this resulted in a very **sparse matrix of features** since each tweet would contain very few of these features.

We observed that **the model peformance increased and the Misclassification Rate on the testing set decreased from 0.4107 using the baseline model to 0.2648 on the improved, augmented model.**

Part 4 - System Improvement Over Baseline Model

The augmented model increased performance and brought the misclassification rate down from 0.4107 to 0.2648, when using Logistic Regression against the full feature set.

Algorithm	Baseline Model Testing Set Misclassification Rate	Augmented Model Testing Set Misclassification Rate
Logistic Regression	0.4107	0.2648
Linear Discriminant Analysis	0.4083	0.2871

The Logistic Regression Model outperformed the Linear Discriminant Analysis model and the computational complexity and runtime was also much shorter with Logistic Regression.

Part 5 - Improving model performance

Feature addition improved the performance of the models through the use of the bag-of-words approach, but I would expect better performance by pruning and selecting relevant features and also **adding other features such as parts-of-speech tagging, bigrams, etc. Using Doc2Vec to encode documents (tweets) into vectors would also allow efficient feature creation.**

Also, **increasing the size of the training dataset by incorporating more tweets that are labeled**, will allow us to train a much better model.

Finally, I would improve the performance of these models by exploring the use of other algorithms as well. **Support Vector Machines and Decision Trees should provide comparable, if not better, results than Logistic Regression.** Using a Naive-Bayes model, I was able to get a Misclassification rate of 0.2661 on the testing set, which is very close to the error rate of 0.2648 using Logistic Regression.