Nicholas Louis Brown

Professor Wallace

IST 707

November 6, 2018

MNB and SVM for Sentiment Classification and Lie Detection

**Intro:**

Attempting to classify review text by sentiment and truthfulness present two very different challenges which can be tackled using two popular algorithms. Using both Support Vector Machine and Multinomial Naïve Bayes algorithms I will attempt to create models capable of predicting the review text’s sentiment and if the review is fake or not. As with previous attempts I believe that both algorithms will be very effective in classify sentiment as it is easier to predict based on tokenized word data. Like humans, however, I believe that the algorithms will have trouble detecting if a review is fake or not. It is very difficult to for modern algorithms to detect things like fake news and fake product reviews, so I don’t think restaurant reviews will be the exception. My primary goal is to answer the question: can reviews be classified as positive or negative based on the words used? Using the Multinomial Naïve Bayes and Linear Support Vector Machine algorithms in Weka I tried to answer this question in two parts: first, using MNB algorithm I made a model which would try to determine if the review text was positive or negative. A second model was also created also determining sentiment using a support vector machine. Both of these models attempted to solve the same problem using the differing algorithms. These tasks demonstrated both the limitations and abilities of multinomial naïve Bayes and support vector machines while helping us to gain a more in-depth understanding of the algorithms themselves. In the end, lie detection proved to be very difficult with accuracy barley reaching 50%. Sentiment analysis was successful using both models.

**Methodology:**

For task 1, both the multinomial naïve bayes (MNB) and support vector machine (SVM) models used the same settings listed below. Additionally, both models were evaluated using a holdout test splitting the data 60/40 training to testing ratio. Finally, to gain greater insight into the models more tests were conducted and documented further below.

**MNB and SVM Settings:**

Tokenization was already done to our data using weka’s tokenizer and arff converter. To create the models the following settings were tweaked to modify minimum document frequency to 3.

* Minimum document frequency: 3

**Analysis:**

My analysis was split into three separate tasks. First, I looked at the words of the classifiers to understand what was happening with the tokenization and classification of tokens. Second, I tweaked the settings of the vectorizers to see if the models could be made more efficient or to see if I could see some differences in the way MNB and SVM classify text. Finally, I concluded my report summing up the findings and testing the model using holdout tests instead of cross-validation.

**Task 1- Lie Detection:**

|  |  |
| --- | --- |
| Most True | Most Fake |
| shrimp | comfort |
| only | could |
| grocery | tea |
| ignored | plate |
| noisy | 2 |
| old | cold |
| where | iced |
| packed | southern |
| chair | cake |
| table | drinks |
| sister | expensive |

The following table shows the inner workings of the MNB and SVM model attempting the lie detection task. Additionally, the table on the right shows the most true/false features that the algorithms recognized as belonging to a fake or real review. From a quick manual analysis, no clear trend presented itself. Using hold out test validation the predicted accuracy for both the MNB and SVM model were 48.57% and 50.41% respectively. While some trends could potentially be useful it is difficult to know with such a low predicted accuracy. Because the accuracy is so low this model should not be used. After trying to predict fake reviews my initial impression of the difficulty of the task is reinforced. I believe that spotting fake reviews from text alone is an immensely difficult task which would require much more work.

**Task 2- Sentiment Detection**

|  |  |
| --- | --- |
| Most Positive | Most Negative |
| town | terrible |
| coffee | none |
| view | took |
| great | asked |
| friendly | our |
| awesome | come |
| chocolate | worst |
| atmosphere | said |
| ice | minutes |
| flavors | hour |
| ingredients | came |

Both the MNB and SVM model created from our positive/negative data had a much better predicted accuracy of 84.44% and 85.97% respectively. As I predicted I was able to see how much better both SVM and MNB were for the sentiment classification task. My initial impression was reinforced after I analyzed the tokens using python. Like in the true/false data before I used python to calculate the log ratios for each of the tokens. Then the top and bottom 10 features were selected with the lowest being the ‘most negative’ and the top most being the ‘most positive’. Upon examination these features give us insight into the workings of the model. Both the most positive and most negative words are ones you would expect to see in a good review and bad review. For customers who had a good time it is clear that desert and location were both important aspects of the restaurant experience. Similarly, for customers who had a bad time, it is clear that service is a major issue over food. Considering both the predicted accuracy of the model and the top 20 tokens I feel that both multinomial Bayes and support vector machine model is very useful for sentiment classification.

**Conclusion:**

From the results of the tests we can see a slight increase in capability of SVM over MNB for both lie detection and sentiment categorization. While examining the top features of the models reveals much about the efficacy of the MNB and SVM algorithms, achieving such increases in accuracy through tweaking the model I am very interested to see just how accurate the model could be made through tokenization and model tweaks alone. Both algorithms proved to be successful in sentiment classification and I would consider using both in tandem perhaps as a double check to see if they agree on the outputted sentiment. It would be very interesting to see the types of reviews or text the models disagree on.