Sentiment Analysis on Rotten Tomatoes Movie Reviews

Yuliya Astapova, William Murdy, Alexander Tairbekov

The Problem

Our Project is to take a one sentence movie review and classify it based on how positive/negative it is. We had a training set of movie reviews that all have a sentiment value based on how positive/negative the movie review is. With 0 being the most negative value and 4 being the most positive and 2 being a neutral score.

Feature Extraction

To extract the features from the data we used a bag of words. To learn the bag of words we used Scikit-learn. We first took the sentences from the training set and removed the non-letter characters. Then we looked at each word in the sentence and removed any stopwords (a, the, is, not etc.). We then turned each sentence in to a vector representation. We then took and created a vocabulary based off all the words that appear in the training set.

Our Naive Bayes implementation

For our own implementation for naive bayes we used the multinomial model. The model we used is $P(Y, W1 \ldots Wn) = P(Y) \prod P(Wi \mid Y)$. Where Y is the sentiment value $Y = \{0, 1, 2, 3, 4\}$. W1 … Wn are the words in the sentence to be classified. P(Y) it the prior probability. Which is calculated by taking the number of sentences in each category divided by the total number of sentences in the training set. $P(Wi \mid Y)$ is the probability of the ith word in the sentence being in each category of Y.

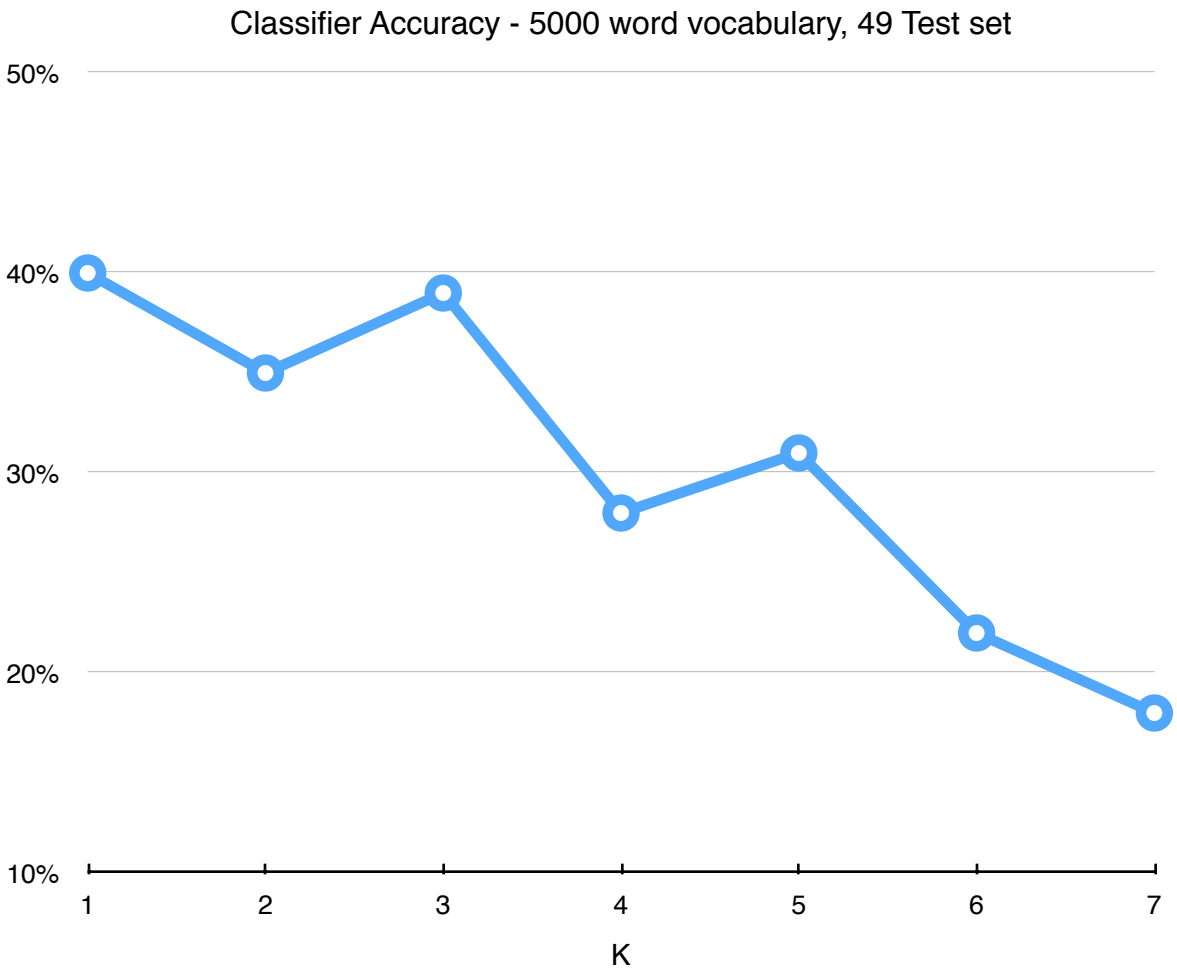
To train the classifier we first calculated the priors. The priors is an array of five values that correspond to the probabilities of a sentence being in each of the 5 categories. Next we calculate the probabilities for each word. To do that first we separated the training data into 5 different sets based on the sentiment value. Then we go through each word in the vocabulary and and calculate the

probability of that word showing up in a sentence for each sentiment value. so at the end of the training we have a dictionary that for each word in the vocabulary it has an array of five values that are the probability that that word is in a sentence of that sentiment value.

We used laplace smoothing to smooth the data. Laplace works by adding a “fake” instance of each word. This is used to prevent any word from having a zero probability which will cause problems later when trying to classify a sentence. We tried Multiple values for the laplace smoothing. We found the best value to be one or two.

After training the classifier we are ready to start classifying movie reviews. To classify a movie review we create an array that will hold the probabilities of for each sentiment value. Following the equation $P(Y, W1 \ldots Wn) = P(Y) \prod P(Wi \mid Y)$ we first take the prior probability. Then we multiply the probabilities of each word in the sentence. We actually take the log of each probability and add them together to prevent any problems that could happen if the probability gets very small. And we classify the sentence according to the sentiment that has the highest probability.

Naive Bayes Results



To train the classifier we used a training set with 8463 reviews. When we extracted the features from the training set we had a vocabulary with 5000 words. We used a test set of 49 sentences. For our own implementation of Naive Bayes we got an accuracy of 40.8163%. We tested multiple values for the laplace smoothing value. We settled on k = 1.