Sentiment Analysis using Naives Bayes Classifier

Luke Segars, Feb 7 2012
CS294: Behavioral Data Mining

Goal

Natural human language has many methods of communicating meaning that are not directly understandable through a dictionary; sentiment is one of these methods. Understanding the sentiment of a piece of text (whether it is referring to its subject in a positive or negative context) requires a more in-depth analysis than simple dictionary references and can be achieved to some degree through the use of statistical inference.

The goal of this project was to determine how well a system could perform on classifying movie reviews as either "positive" or "negative" using naïve Bayesian inference.

Parameters

A number of techniques were implemented on top of the simple naïve Bayes classifier in order to measure the impact on performance. The following techniques were implemented and run on the same set of data; the results of these trials are available in the next section.

Multimodal and Bernoulli models. The multimodal model keeps count of the number of times a word appears in each document, and the Bernoulli model simply provides indicators for whether a word appears in a particular document. The two models performed similarly in most tests but experiments with the multimodal model had slightly better performance overall.

Stemming. Stemming tokens involves removing common endings to word to allow words like "homes" and "home" to be counted as identical words. I used a standard Porter stemmer in my project, provided by the ScalaNLP library.

Stopwords, Word Filtering, and Simple Mutations. Stopwords are very common and often semantically meaningless terms like "the," "an," and "it." I used the ScalaNLP standard stopword filter for removing these terms in the appropriate tests, and the stopword filter was always applied after stemming when the two operations took place in the same experiment.

In addition, I also implemented some additional simple word filtering, removing any words that were below two letters in length. This was primarily done because I noticed that some of the terms that occurred most frequently were punctuation marks; the chi-squared statistic indicated that the distribution between the two training classes was quite similar, meaning that the terms could be removed without losing significant amounts of information. A few simple mutations were also applied, including stripping non-alphanumeric symbols and converting all tokens to lowercase.

Feature Selection. The large number of features in the dataset (46,477 distinct tokens before stemming and filter application) meant that including all of the features could generate a large degree of noise. I analyzed the significance of each term as a class partitioner using the chi-squared statistic and sorted all tokens based on this score. I then ran a number of tests using different feature set sizes to determine which size performed best on the given data.

Additive smoothing (alpha) optimization. Because of the power law distribution of words in the English language and the relatively small average document size in this experiment, the document vectors are often sparsely populated. Since introducing a zero count into the equation essentially ruins the computation, a value (alpha) is added to each true count, typically <= 1. I experimented with several different values for this parameter on a number of different configurations.

Experiment

The dataset consisted of 2000 movie reviews that were pre-labelled as being either "positive" or "negative;" there were 1000 reviews for each class. Each class was segmented into 10 partitions and each experiment was cross-validated with each partition as the experiment set (90% training, 10% experiment). All results that are reported below are the averaged results from all 10 experiments.

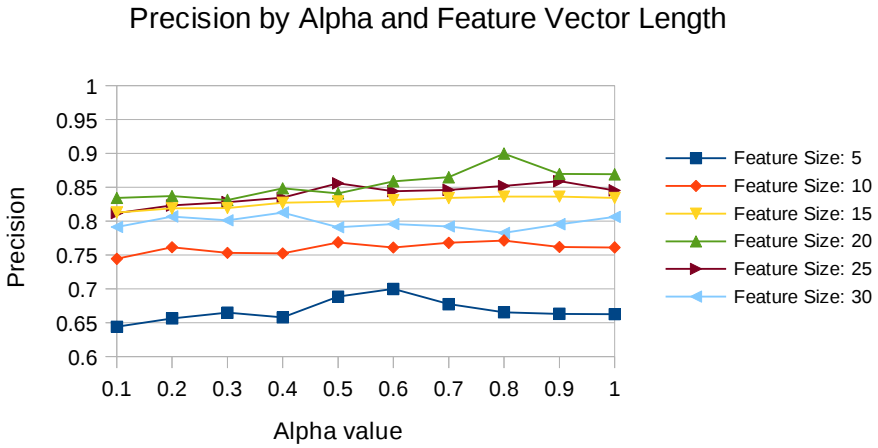
Model Type	Alpha optimization	Feature selection	Stopword filtering	Stemming	Average Precision	Average Recall	Average F1 Score
Multinomial	**N**	**N**	**N**	**N**	**.8170**	**.8155**	**.8163**
Bernoulli	**N**	**N**	**N**	**N**	**.7983**	**.7390**	**.7675**
Multinomial	Y (a = 1.0)	N	N	N	.8170	.8155	.8163
Bernoulli	Y (a = 1.0)	N	N	N	.7983	.7390	.7675
Multinomial	N	N	Y	N	.8107	.8090	.8099
Bernoulli	N	N	Y	N	.7972	.7465	.7710
Multinomial	N	N	N	Y	.8149	.8140	.8145
Bernoulli	N	N	N	Y	.7966	.7335	.7637
Multinomial	N	N	Y	Y	.8118	.8118	.8114
Bernoulli	N	N	Y	Y	.7954	.7395	.7665
Multinomial	N	Y (a = 25)	N	N	.8278	.8170	.8223
Bernoulli	N	Y (a = 25)	N	N	.8011	.8314	.8160
Multinomial	**Y (a = 0.8)**	**Y (a = 20)**	**N**	**N**	**.8998**	**.8541**	**.8764**
Bernoulli	**Y (a = 0.8)**	**Y (a = 20)**	**N**	**N**	**.8714**	**.8633**	**.8673**
Multinomial	Y (a = 0.9)	N	Y	N	.8175	.8145	.8160
Bernoulli	Y (a = 0.9)	N	Y	N	.8005	.7384	.7682
Multinomial	Y (a = 0.8)	N	N	Y	.8190	.8680	.8428
Bernoulli	Y (a = 0.8)	N	N	Y	.7995	.7391	.7681
Multinomial	**Y (a = 0.8)**	**N**	**Y**	**Y**	**.8314**	**.8114**	**.8213**
Bernoulli	**Y (a = 0.8)**	**N**	**Y**	**Y**	**.8408**	**.8080**	**.8241**

Results for variation permutations of the parameters. Values enclosed in parentheses are the ones that were found to be optimal for each experiment. The configurations that have higher performance metrics have been emphasized for clarity.

A number of different experimental permutations were run to determine how the precision and recall rates were impacted. Other permutations were possible but were not run due to time constraints. Many of the parameters did not seem to have a significant performance impact, and the cause of this is

examined more closely in the Analysis section.

Feature vector length and alpha parameter selection
Each experiment that required feature selection or alpha optimization was performed by iterating over a set of potential values; feature selection attempted to optimize over feature vectors of lengths 5, 10, 15, 20,..., 40 and alpha values of .1, .2, .3, …, 1.0 were measured for each test. The data from each of these experiments is presented in the chart below, with feature sizes of 35 and 40 being left out for readability (their performance continues to decline, similarly to how feature sizes of 25 and 30 decline from one size smaller than themselves).



Analysis

Stopword filtering and stemming didn't have a significant impact in the tests that were conducted; this is likely because the feature vector was still very long and the percentage of tokens impacted by these techniques would be relatively small. It would be interesting to run an experiment with feature selection as well as stopword filtering and stemming, but I was unfortunately unable to do so due to time constraints.

Feature selection seemed to be the most impactful technique of those that were implemented. This is likely due to the high signal-to-noise ratio in a dataset of this size. There were approximately 46,500 distinct tokens in the original dataset and the vast majority of them had low chi-squared values when viewed over the whole dataset; eliminating a large number of these helped focus on the key features and kept the important information from being overshadowed by the hundreds of thousands of irrelevant data points. The impact of optimizing the alpha value seemed to be somewhat unpredictable when used with feature selection; it typically improved performance but there was no clear trend in where the optimal alpha value would be (see graph above).

In some cases the tokens that appeared in my feature selection vectors were surprisingly relevant to a human reader. Many of the terms have sentiments associated with them (both positive and negative) and are also words that are not commonly negated, i.e. “laughable,” “brilliant,” and “unfortunately.” This was particularly interesting as the adjectives and adverbs that *could* be commonly negated *seemed* to be and did not appear near the top of the list; this is, in some ways, a partial functional overlap with bigrams that are able to detect and appropriately handle term negation.

Appendix: Top Words

Boring	?	Film	Godzilla	Worst
Excellent	Mess	Effective	Both	Perfectly
Subtle	Pointless	Unfunny	Worse	Performance
Waste	Ridiculous	Supposed	Lame	Memorable
Dull	Stupid	Awful	Poorly	Outstanding

Highest-ranking words according to the chi-squared test. This list is pulled from an experiment that did not include stemming but did include stopword filtering.