Project 2 Report

Philippe Lorange[27295286], Martin Spasov[40000916], and Nicolas Champagne[27043751]

Concordia University

1 Estimating parameters for the Naive Bayes Classifier

The first step in developing a Naive Bayes Classifier is to estimate the parameters needed to classify new documents. Our classifier uses four elements:

- positive_word_probabilities, a list containing the probability of each word showing up in a positive document;
- negative_word_probabilities, a list containing the probability of each word showing up in a negative document;
- positive_probability, the probability that a document is positive;
- negative_probability, the probability that a document is negative.

Using add-1 smoothing, every the parser comes across is added to the overall vocabulary, and to the positive/negative counters. The following are notable results of the estimation step:

- $P(positive) = 0.509$
- $P(negative) = 0.491$

- $P(\text{"}excellent\text{"}|positive) = 0.00057$
- $P(\text{"}excellent\text{"}|negative) = 0.00013$

- $P(\text{"}awful\text{"}|positive) = 2.586 \times 10^{-5}$
- $P(\text{"}awful\text{"}|negative) = 0.000121$

- $P(\text{"}problem\text{"}|positive) = 0.000357$
- $P(\text{"}problem\text{"}|negative) = 0.000672$

- $P(\text{"}the\text{"}|positive) = 0.0433$
- $P(\text{"}the\text{"}|negative) = 0.0416$

The results demonstrate that the estimator correctly identifies that words with a negative connotation have higher probabilities of being found in negative documents, with the opposite being true for positive documents. For very common words like "the", the estimator gives a high probability to both the positive and negative documents; this does not truly affect the outcome of our classifier, however. Since both positive and negative documents use these words, we can assume that they do not have an impact on the classification.

2 Classifying new documents

When classifying new documents, the probability of the document being positive is compared to the probability of the document being negative. To do so, the logarithm of each word's probability is added to the logarithm of the probability of the document being either positive or negative. The higher score determines whether the document is positive or negative.

Class	Documents correctly identified	Total number of documents	Accuracy
Positive	907	1153	78.664%
Negative	1036	1230	84.227%
Overall	1943	2383	81.536%

These results show that the negative documents were generally more accurately classified, compared to positive documents. A hypothesis as to why would be that negative phrases can include positive words, with a negation. An example would be "The product is not good". On the contrary, positive documents do not negate negative words nearly as much, e.g. "The product is not awful". The probabilities are therefore tilted towards negative classification; in other words, a document would need to be "distinctly positive" to be classified as positive.

Another source of error was perceived to come from anecdotal documents. A document such as the following has a lot of "noise", that can confuse the classifier:
"i am a br5-49 aficionado from the early days at robert 's when the band played for tips and usually played requests until 2am with little or no breaks . admittedly , i long for those days as much as i long for stirrups in major league baseball . so i 'm nostalgic , " it 's a free country . " it is also legal for a band to evolve and morph in any which way it feels . yet , whenever i read a review in some fishwrap publication that states that a band has " matured , " i can do nothing but cringe . {...} ok , i 'll go cry me a river , but dammit , just go listen to the big backyard beat show one more time . mature ? i just do n't want to grow up that bad"

The results are still satisfactory, in that the algorithm correctly classifies over 80% of documents.

3 Difficulties

Given the large size of the training data, a difficulty for this classifier was to differentiate between an error in our algorithm, or a badly classified document in the training/testing data. For example, the document "The cd came as promised and in the condition promised. I'm very satisfied" was originally classified as being a negative document, but in reality it can be agreed that this is a positive

sentence. These were initially researched, but it was concluded that their impact on the final accuracy was negligible.

4 Future Reflections

Overall, the classifier discussed in this report is accurate enough to generally classify simple documents into positive or negative classes. However, a future project would be to classify documents into ratings, such as a 1-5 start rating system. Such a classification could be done one of two ways: the first, by having a rating instead of a pos/neg tag in the documents. The second way could be to use the probability of the document being in a class as an indicator of our "confidence". For example, a document that is overwhelmingly positive would have a high probability of being classified as positive, and would therefore have a high star-rating. Same with negative, and documents that barely have a difference in pos/neg probabilities would have a 3-star rating, for example.

Should this project be developed further, it would be interesting to research the question of anecdotal documents adding noise to the classifier. Would it be optimal to crop these documents based on length? Perhaps filter out sequences that don't have "key" words that help classify documents? This would be an interesting reflection.

References

1. Houari, N.: COMP-472 Machine Learning I Slides. Class Material (2019)