Further notes on Naive Bayes

Jurafsky and Martin's draft chapter https://web.stanford.edu/~jurafsky/slp3/4.pdf is a very good introduction to Naive Bayes and its use, so we will not repeat that material. The current document elaborates on some related topics and suggests some further investigations of the classifier (starred ticks).

Models and approximations

The aphorism "All models are wrong but some are useful" (Box, 1978) sums up much of what ML is about. The assumptions we make in the Naive Bayes approach to sentiment classification are wrong, but this is true of the assumptions made in all current formal models of human language (statistical or otherwise), with the possible exception of a few which are very restricted indeed. However, the question is whether a model provides useful results. We could mean a number of things by "useful" here. Practical utility in some system with real users is one possible goal. The development of deeper understanding of the phenonomenon is another. Naive Bayes is extremely useful as a **baseline** system in modelling human languages (**baseline** is a concept we discuss further below).

We are forced to use statistical models which are wrong because of two things:

1. computational tractability
2. acquisition of training data

There is a very nice, succinct description of 'An optimal Bayesian agent' in Bostrom's *Superintelligence* (Bostrom 2014). Bostrom explains that even a simple computer monitor with 1,000 by 1,000 binary pixels has too many possible states to model exhaustively, even if the maximal computational capacity of the entire observable universe were available. The Naive Bayes assumption is equivalent to saying that the pixels on the monitor all behave independently of each other. Naive Bayes works (to an extent) when we are interested in an overall characterization of some collection of subentities: it could give us a model of the overall shade that the monitor was displaying. It would not be a useful model if we were trying to decide whether the monitor was displaying a picture of a cat or a horse.

Intelligent behaviour requires taking advantage of interactions between states. Living creatures have necessarily evolved in a way which allows them to learn correlations. More sophisticated probabilistic models than Naive Bayes also rely on modelling interactions. Later in the course, we will look at Hidden Markov Models, which provide us with simple (but often very effective) ways of modelling sequence information.

In general, making a model a better fit (i.e., less wrong) requires more training data. A model may be computationally tractable but it may be impossible to

acquire the data to train it: this is particularly the case for models which require manually annotated data.

The Naive Bayes independence assumption

It is clear that the Naive Bayes independence assumption is wrong. To take a trivial example, the probability that "James" occurs in a movie review is not independent of the probability that "Bond" occurs. Nevertheless, Naive Bayes often works quite well. One explanation that is often given for this is that we don't use the probabilities themselves, but merely look at which class has the highest estimated probability.

Naive Bayes as a baseline

Whenever we investigate an ML model, we want to know whether it performs better than the competition, especially if there is a simpler or more tractable approach. The existing approach is referred to as the **baseline**. Naive Bayes is often a good choice as a baseline model.

Whenever we investigate a new task, we want to see whether there is some hope of a ML approach working at all. It is often worth trying a simple model, such as Naive Bayes, before trying something more complex. This is not exactly the same as developing a baseline, but is clearly related.

The type/token distinction

Consider the following sentence:

> The old horse went to stand next to the old man.

We would normally say this sentence has 11 words in it, but we could also say it has 8 words because "the", "old" and "to" are repeated. To avoid the ambiguity, we say that the sentence has 11 tokens but only 8 types.

A note here: we could also claim that the sentence has 9 words, because the two uses of "to" are completely different from each other. For the purposes of this course, we will take the simple-minded approach, and count two tokens as being of the same type if they share a spelling.

Further investigation of the classifier (starred ticks, strictly optional)

Training data

How does performance of the classifier vary with the amount of training data available? To investigate this, do multiple runs with different amount of training data (keeping the test data constant) and plot a graph of accuracy against the size of the training set. Does this suggest that performance would improve with more training data? How much training data is needed for the Naive Bayes classifier to outperform the lexicon based classifier?

Size of lexicon

One could ask a similar question about the size of the lexicon: might a much bigger sentiment lexicon be better? Assume that the lexicon was created by taking words in their frequency order in some general text and marking those which were thought to indicate sentiment: would going further down that list improve the lexicon? We can approach this question by removing lower frequency words and looking at how the performance degrades and extrapolate from that in the other direction to suggest how performance might improve. Various frequency lists are available for English, or you could just use the frequencies in the movie texts.

Most distinguishing features

Some words are intuitively better indicators of sentiment than others, and this should show up in the probabilities you acquire from the training data. How might you express this formally? Which 20 words are the best features for each class by this definition? Do you find all of these words in the sentiment lexicon? Do you think they would be good features for another domain, such as phone reviews?

Reliability of classifier results

In some cases, the difference between the estimated probability that the sentiment is positive versus the probability it is negative will be small. Investigate whether the classifier is less reliable in this case.