**Naïve Bayes and sentiment group exercises**

**AU CogSci MA – NLP**

**Oct 9, 2019**

Part 1: Group Exercise

We want to build a naive bayes sentiment classifier using add-1 smoothing, as described in the lecture (not binary naive bayes, regular naive bayes). Here is our training corpus:

Training Set:

- just plain boring

- entirely predictable and lacks energy

- no surprises and very few laughs

+ very powerful

+ the most fun film of the summer

Test Set:

predictable with no originality

1. Compute the prior for the two classes + and -, and the likelihoods for each word given the class (leave in the form of fractions).
2. Then compute whether the sentence in the test set is of class positive or negative (you may need a computer for this final computation).
3. Would using binary multinomial Naive Bayes change anything?
4. Why do you add |V| to the denominator of add-1 smoothing, instead of just counting the words in one class?
5. What would the answer to question 2 be without add-1 smoothing?
6. Can you think of any other features (or preprocessing) that you could add that might be useful in predicting sentiment? (This will come in handy for next HW!).
7. Naive Bayes treats words as if they are independent conditioned upon the class (that is why we multiply the individual probabilities). For which (if any) of the new features you suggested does this independence assumption roughly hold?

**Part 2: Challenge Problems**

1. Go to the Sentiment demo at http://nlp.stanford.edu:8080/sentiment/rntnDemo.html. Come up with 5 sentences that the classifier gets wrong. Can you figure out what is causing the errors?
2. It is sometimes the case that more complex features (like trigrams or bigrams) perform better than simple features (like unigrams) on the training set, but perform worse than simple features on the test set. This is a particular case of the phenomenon called `overfitting' in machine learning. Discuss why this might be. Can you create a tiny training set with 2 3-word documents and a test set with one document for which this overfitting situation holds?
3. Binary multinomial NB seems to work better on some problems than full count NB, but full count works better on others. For what kinds of problems might binary NB be better, and why? (There is no known right answer to this question, but I'd like you to think about the possibilities.)