HW 1 - Written Assignment

Elvis Fernandez

February 2017

1 Laplace Smoothing

Please show that $Pr_L(y|x)$ is a valid probability distribution

Solution:

- $Pr_L(y|x) = \frac{\#(x,y)+1}{\#(x)+V'}$, where $V' = V + 1$

Usually when you have some model of probability of some event occurring over some sample space, Pr_{MLE} is sufficient enough to help you estimate parameters of a model. This computation has one significant drawback. And that is it will give a probability of zero to a lot of words. This is indeed troublesome to NLP applications because the idea behind a lot of NLP applications is to build a model from some training data in turn to be used on test data. For example, just because the sequence 'the book' is never seen in training does not mean that it is an unlikely sequence, and should be assigned a probability of zero, because 'the book' is a common sequence and possibly can be in the test data.

The idea of is Laplace smoothing to pretend that every bigram seen in training is seen one more time than it was actually seen.

Suppose these bigrams seen in training: the test, the book, the book, the tv, the tv, the tv

- $Pr_{mle}(test|the) = \frac{1}{6}$, $Pr_{mle}(book|the) = \frac{2}{6}$, $Pr_{mle}(test|the) = \frac{3}{6}$
- $Pr_L(test|the) = \frac{2}{10}$, $Pr_L(book|the) = \frac{3}{10}$, $Pr_L(test|the) = \frac{4}{10}$
- The idea behind laplace is to pretend we have seen every word in the vocabulary one more time, it essentially steals some probability of frequent words to less frequent or non-existent words.
- Since this probability is still a probability it needs to include the size of the vocabulary in the denominator to make sure the sum of all probabilities still adds to one.

2 Pr_{AD} Validity

Specify how to compute Pr_{AD}(y—x) for cases where (x, y) = 0, so that it is a valid probability distribution. You should provide that with your specification, Pr_{AD}(y—x) is indeed a valid probability distribution.

$Pr_{AD} = \alpha(x)\beta(y)$ if #(x, y) = 0

Where

$\alpha(x) = 1 - \sum Pr_{AD}(w|x)$ where w = $w : \#(x, w) > 0$

$\beta(y) = \dfrac{Pr_L(y)}{\sum Pr_L(w)}$ where w = $w : \#(x, w) = 0$

Beta is calculated by taking the unigram laplace probability of y and dividing it by the summation of all bigrams that (x, w) = 0, essentially all bigrams that didn't start with x, this is a very expensive operation and requieres calculating laplace for almost all of the vocabulary. A quick optimization mentioned in the notes and implemented in my code is to instead take advantage that this is a probability distribution and the summation is equal to one so instead calculate I calculated laplace for all bigrams containing x and then subtracting it from 1 to get my desired value.