CFCS1: Practical 8
Language Modelling and Entropy

Peter Bell

March 6, 2008

Introduction

In this practical you will study the distribution of letters (alphabetic characters) in the English language. Data on the occurrence of each letter have been obtained from the comprehensive English dictionary found in the file `/usr/share/dict/words` (using only words containing entirely characters and no numerals). You will use the data to create two probabilistic models.

Load the file `lab8_data.mat` into Matlab. This creates the following variables:

- `chars`, an array containing all the letters in order, and a 27th symbol, `<b>`, signifying the gaps between words – this should be treated like any other character.
- `unigram_counts`, a vector containing the number of occurrences in the dictionary of each letter. For example, `counts(1)` is the number of times 'a' occurs in the dictionary.
- `bigram_counts`, a matrix containing counts of pairs of adjacent letters in the dictionary. For example, `bigram_counts(1,2)` is the number of times 'a' is followed by 'b' in the dictionary.

1 Analysis

1. In Matlab, list the letters ordered by how frequently they occur in the dictionary of English.

2. What is the average length of a word in the dictionary?

3. Compute the probability of observing each letter (including word breaks), assuming successive letters in a word are independent.

4. Calculate the entropy of the distribution. What is the expected number of bits per letter needed to code an English word?

5. What assumptions have you made in Question 4 that mean that your answer is unlikely to be true in practice for coding English text?

2 Bigram models

Your probabilities in the previous section constitute a *unigram* language model. You will now create a *bigram* language model. This makes the Markov assumption – that the probability of a letter depends on the preceding letter, but, given the preceding letter, is independent of all other letters.

1. Use the data in `bigram_counts` to compute the full set of bigram probabilities for letters, $p(L_i\,|L_{i-1})$, where L_i is any letter and L_{i-1} is the preceding letter.

2. Use the original unigram model to compute the probability of observing the word 'enjoyment', and of observing the fake word 'eejmnnoty'. (It is helpful to work using logs).

3. Now use the bigram model to calculate the same probabilities. Comment on your findings.