ECE 537 **HW #7 – Version 1.01 November 13, 2018** **Fall 2018**

Univ. of Illinois **Due Dec. 12 (or Reading Day)** **Prof. Allen**

Topic of this homework: Information theory, language models, entropy and the EM algorithm.
Deliverables: A detailed summary that demonstrates your understanding of the topics. I want a full discussion of each topic.

1. The entropy of speech

   (a) Given the following table of consonant and vowel syllables, what is the entropy of English, in bits/syllable?

Syllable Type	Occurrence (%)
V	9.7
VC	20.3
CV	21.8
CVC	33.5
VCC	2.8
CCV	0.8
CVCC	7.8
CCVC	2.8
CCVCC	0.5
	100

   (b) If the average syllable duration is 200 [ms], what is the bit rate for speech?

2. Write a program to sort all the words in Shakespeare's play *As you like it.*[1]

   Option: You may alternatively work with *Green Eggs and Ham.*

   Materials:
   https://jontalle.web.engr.illinois.edu/uploads/537.F18/Assignments/HW07.pdf

   Useful Matlab commands for this exercise `fgetl, k=find(string==' '), and sort`. I have provided an example Matlab script `srt.m` to read in the text file, and do some limited processing on it.

   (a) Sort all the words from "As you like it" by Shakespeare. Remove the words that are all upper case, as a first pass, so as to remove the names and directives. All white space and new lines are treated as delimiters. Remove punctuation and special characters such as –, (), !, etc and periods (I did this for you, and the output is in the URL:
   https://jontalle.web.engr.illinois.edu/uploads/537.F18/M/files/AULI+GEnH.tgz
   (and tared-up in file AULI+GEnH.tgz).

   (b) Compute first order statistics (relative frequencies).

   (c) list the top 100 most likely words.

   (d) Plot a Zipf's Law type plot (See Shannon 1950 paper for the definition of this Law)

   (e) Compute second order statistics (frequency of word order pairs).

   (f) list the top 20 most likely word pairs.

   (g) Write a program to generate "fake Shakespeare" sentences by using the 2d order statistics to output sentences. Don't worry about starting or stopping the sentences, with real starting and stopping words. Start the sentences with the top 10 most frequently labeled pairs.

[1]Credit: I downloaded the play from http://www.gutenberg.org/.

i. How do you handle the case of no data in the 2d order matrix? Hint: Make up a rule. Can you think of more than one resolution to this problem of insufficient statistics in the 2d order estimates?

ii. How do you handle "start" and "end" with this scheme?

iii. Can you figure out how to generate rhymes at the ends of phrases?

(h) Run your program to look at frequency, on your fake sentences, and compute the second order frequency tables for your output, to show that it has the same statistics as what you started with.

3. Using the EM algorithm, write a program to separate the the data provided in the Matlab mat file, into two Gaussian distributions. Report the mean and variance of the two data sets.

   I would prefer you write you own code but there is a simple way to do it using the matlab function `gmdistribution`.

   http://jontalle.web.engr.illinois.edu/uploads/537.F18/Assignments/HW8.F06

4. This next problem is simply too hard for most people, but I give a lot of points if you even attempt this one, and get something that makes any sense at all.

   Write a program to separate the speech and noise using the STFT and the EM algorithm. The general outline is to start from speech contaminated by noise, and to remove the noise in narrow frequency bands, via the STFT and the EM algorithm

   The first step is to add noise to speech at a know SNR. The second step is to apply the STFT to the signal, splitting it into frequency bands as a function of time. The third step is to remove the noise in each band, by estimating the noise and speech distributions over time, and by the use of the EM algorithm in each band, split the signal into a speech and noise STFTs. Finally by the use of Over-Lap and Add (OLA), reconstruct the speech and noise. Finally do this at several SNRs and report your result. At a +10 dB SNR, their should be functionally no degradation.

   When applying the EM algorithm, you must model the two distributions (narrow band speech and noise, in each frequency band) as zero-mean Gaussian distributions having a standard deviation (SD) σ_n and $\sigma_s(t)$. For the case of the noise, the SD is a constant (it depends on the SNR. For the case of the speech, the SD will depend on time, making this component a bit more tricky to estimate. It could be that there is a better distribution to assume, such as a mixed Gaussian model, or a Laplace model.

   When applying the EM method, you need to start by guessing the initial parametric unknown values (i.e., σ_n and $\sigma_s(t)$). Given the guess, compute the ratio of the two parametric distributions, and find the points where they are equal. These points tell you your initial guess as to how to separate the samples into speech and noise. Threshold the values into the two groups, depending on what region the sample is in. Then recompute the means (still zero) and SDs for the two groups, but now based on your initial guess. Use these new estimates of the STDs to re-estimate the distributions, and obtain new thresholds. Iterate the process for a few times until the answer stops changing (until the solution converges).

   (a) First use the speech file from the previous assignment and add noise to it at a 0 dB SNR. To do this, normalize both the speech and noise samples so that their RMS level is one (compute the RMS of the sample by matlab's `std()` command, and divide the signals by their RMS level.

   (b) Listen to the samples to make sure they sound reasonable (that the noise is about as intense as the speech.

(c) Compute the STFT of the signal, and apply the EM algorithm in each band. Here you must work over a few frames of time, in each frequency band, so that you can take into account the time varying nature of the speech sample. One way to decide how many frames you need to take is to look at the RMS level of the speech as a function of the number of frames. Since I have not actually tried this step, it may be that you just make a decision as to where the speech is zero, and where it is non-zero, and split these regions in a bi-modal fashion.