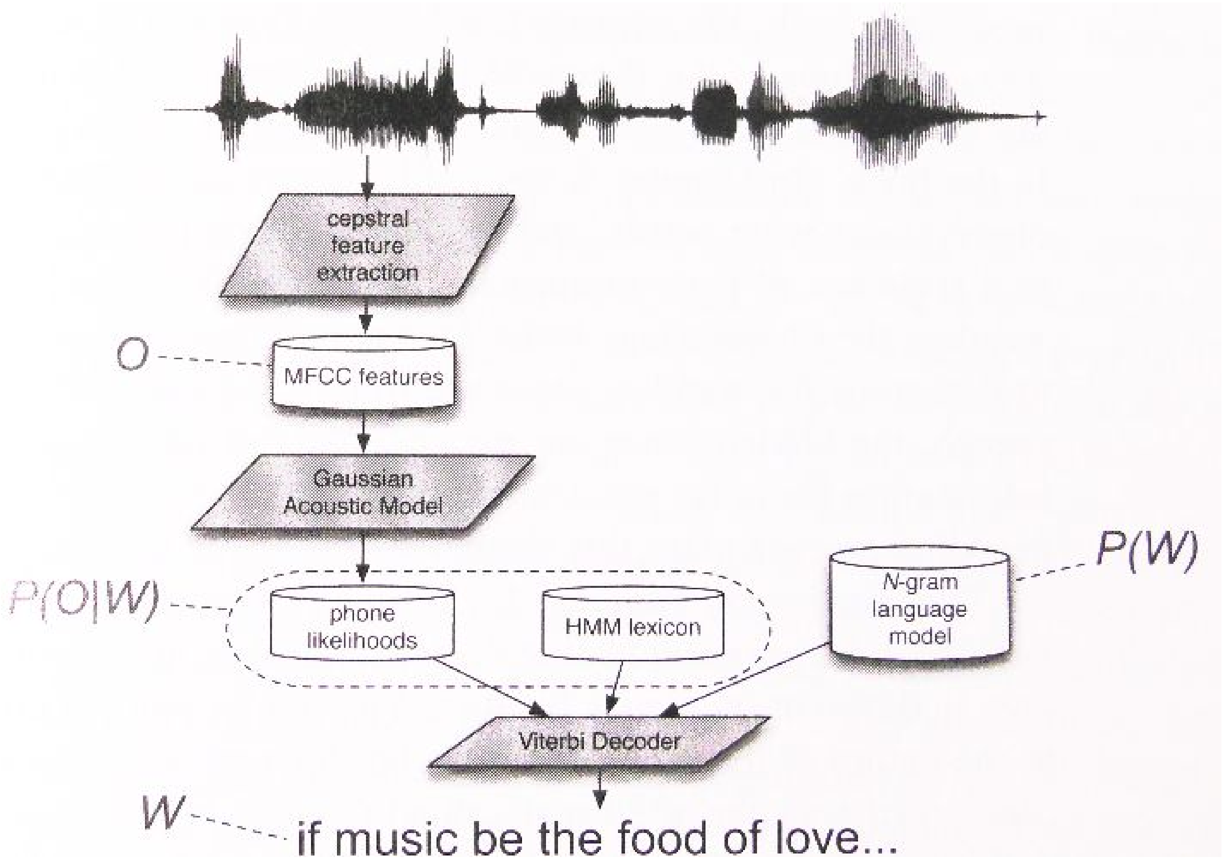
Speech Processing and Recognition

Part I



Maryam Najafi[1]

02.15.2017

ECE Dept., Florida Institute of Technology

[1] mnajafi2012@my.fit.edu

OVERVIEW

The objective is to recognize the most likely sentence out of all sentences in the language L given some acoustic input O (observations).

Because the speech is variable, an acoustic input utterance never exactly matches the probabilistic model. Therefore, this gives a rise to Bayesian Inference. We candidate a source sentence (modified by passing through the noisy channel) and we give it to a noisy channel. The goal is to find how the channel distorts the input utterance so then we can find the correct source sentence. We search through the Instance space (all possible sentences) to give each of them a probability for matching the source sentence. Obviously, we end up with the source sentence itself with the highest probability. Since we cannot afford searching throughout the Instance space, the problem becomes the **decoding** or **search** to consider those sentences that have a good chance of matching the source sentence (input). In other words, we are making our Instance space BIASED.

BAYESIAN INFERENCE

1. The observation O (input in the Feature space) breaks down into time-basis intervals o1, o2, ..., ot.
2. The instance space W (words or possible hypothesized sentences) breaks down into words w1, w2, ..., wn.
3. The language model follows the **Bayesian rule**.We have the a-priori (P(w)) which is calculated by the **language model (LM)** (N-gram grammars). This probability indicates the likelihood of a sentence in the language. Briefly, the probability of a given set of words is a sentence of English language for instance is calculated applying the chain rule of probability as:

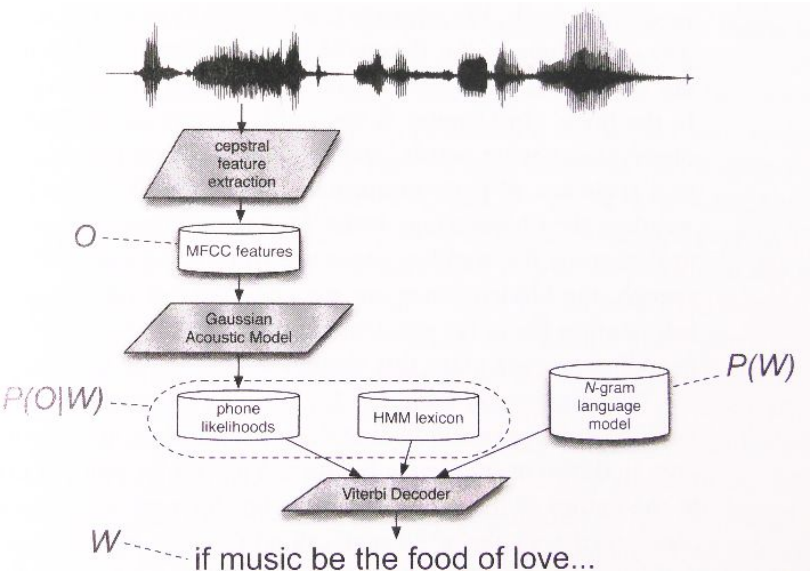
$$P(w_1^n)=p(w_1)\,p(w_2|w_1)\,p(w_3|w_1^2)\ldots p(w_n|w_1^{n-1})\approx\prod_{k=1}^{n}p(w_k|w_{k-N+1}^{k-1})$$

4. Moreover, we have class conditional probability (P(O|W)) calculated by the **acoustic model (AM)**. Accordingly, we can calculate (the estimated W) which is equivalent to the a-posteriori (P(W|O)) for W using the following equation:

$$\hat{W} = \underset{W\in L}{argmax}\, P(W|O) = \underset{W\in L}{argmax}\, \overset{likelihood}{P(O|W)}\, \overset{apriori}{P(W)}$$

Question: Is this just an assumption to consider an arbitrary (or Gaussian Mixture Model classifier) distribution for the class conditional distribution of f(O|W)? The answer is we should employ HMM to build an AM that computes the likelihood P(O|W). And in the HMM the model that is being used is assumed to be Gaussian.

5. Thus, we need both AM and LM models to operationalize the resulted probabilistic model in a search algorithm. We pick our **recognizer** using LM and AM.
6. This schematic illustrates the architecture of a speech recognizer.[1]



7. Three stages are shown above:

- Signal Processing (Feature Extraction) stage (output: MFCC's)
- Acoustic Modeling stage (output: a sequence of acoustic likelihoods for each feature vector)
- Decoding stage
  (**inputs:**
  - HMM dict. of word pronunciations (a word is an HMM and a phone or subphone is it's state).*p(O|W)* **1**
  Gaussian estimator (for the Maximum Likelihood Estimation, MLE) governs the output likelihood function for each state.
  - The sequence of acoustic likelihood for each time frame. *p(O|W)* ***2***
  - Language model (P(w)), the same a-priori probability.

P(O|W) is the same observation likelihood. (Recall: p(W|O) = p(O|W) p(W))

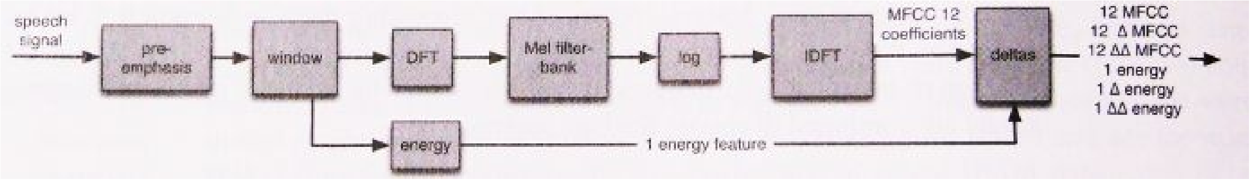
Output:

- The most likely seq. of words. W given by the employed Viterbi algorithm for decoding HMMs (e.g. if music be the food of love ...)

8. **Viterbi** algorithm is being used to decode HMMs and **Baum-Welch (Forward-Backward)** algorithm to train HMMs.
9. Let's say an HMM's emission state represents a phone (e.g. ay). Phones pronunciations could be different, depending on the number of its HMM's emission state's repetition. The spectrum of "ay" phone shows two different non-steady changes for F1 and F2. (Figure 9.5 in the book) Therefore, a middle state is deemed for each state. In total, each HMM emission state comprises with a transition-in, steady-state, and transition-out state. Ultimately, an HMM contains a bunch of these three-state emission states, plus two non-emitting states at either side.
10. Simply it's like Q combined with A means **lexicon**: a set of pronunciations for words.
11. We already have LM according which we computed P(w). How do we create the Gaussian Acoustic Model from the MFCC features? Using HMMs!

MFCC Feature Extraction

Now it's time to first introduce the actual acoustic observations (MFCC feature vectors, Mel Frequency Cepstral Coefficients), then continue with computing their likelihoods.



1. **Digitizing:** (Analog-to-Digital)
2. **Sampling:** (The **Nyquist frquency** is the maximum frq. for a given sampling rate. The Nyquist frq. is half the sampling rate. Since most info in human speech is in frq. below 10MHZ, a 20MHZ sampling rate would be enough for complete accuracy).
3. **Quantization:** Represent real-valued numbers as integers (8bit or 16bit)
4. **Pre-emphasis:** Now let's do some preprocessing on the input audio. **Spectral tilt** causes problems for high frequencies. Boost high frequencies for higher formants because they decay in time due to glottal pulse.

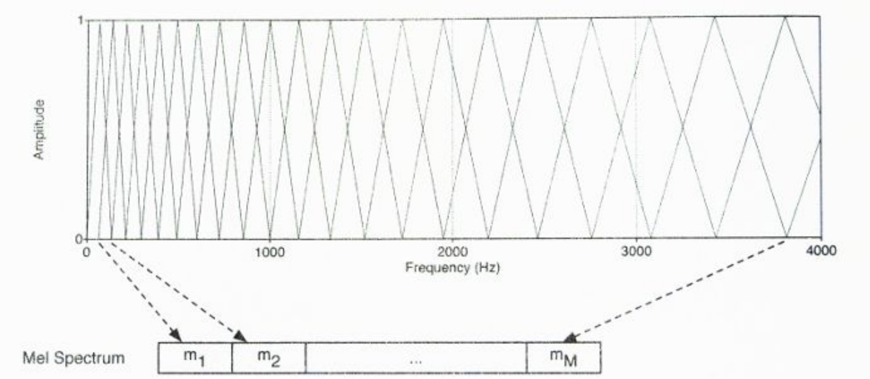
5. **Windowing:** We don't want the entire input audio be processed in whole because speech is a **non-stationary** signal (as brain waves). The statistical parameters in speech is not constant during time. Thus, let's segment the input audio into little windows (roughly stationary) in which the statistical parameters are considered as constant as possible. The window is non-zero inside and zero otherwise. Different filters such as hamming, Gaussian, triangular, rectangular*, and etc could be applied for this stage in order to window the waveform. They could affect differently depending on the **offset**, **frame size (e.g. 25 ms)**, **frame shift (e.g. 10 ms)**, and it's **shape**. Name your window *w*[n] at time n. Note that the windowing is being done in the time domain.

$$\mathbf{y[n] = s[n].\, \mathit{w}[n]}$$

; where s stands for input speech signal and y represents the windowed signal at time n using window w[n].

*. The rectangular windowing causes discontinuity therefore, problems for the Fourier analysis.It abruptly cuts off the signal at its boundaries.

6. **Discrete Fourier Transform:** Derive from your windowed signal the information of the energy of frequency bands by taking DFT of each window. Obviously the windowed signal x[n] .. x[m] is the input and output is a set of the complex numbers representing the magnitude (real part) and phase (imaginary part). Plot the magnitude against the frequency visualizes the **spectrum**.
7. **Mel Filter Bank and Log:** Human is less sensitive toward high frequencies in the range of roughly 1000HZ and above. Thereby, we consider this property of the human hearing in our model by employing **mel** scale. A mel is a unit of pitch. The objective in this step is to wrap up the frequencies resulted from the FFT onto the mel scale. Eventually the result would be linearly spaced for frequencies below 1000HZ (10 filters) and logarithmically spaced for the rest (above 1000HZ). **mel(*f*) = 1127 ln(1 + *f*/700)**. The human response to lower frequencies is relatively linear while it's logarithmic to the higher frequencies. Humans can sense the difference of amplitude better in low amplitudes than in high amplitudes. We take the log of each mel spectrum value to ignore the effect of speaker's mouth moving closer or further from the microphone. Employing the logarithm of mel spectrum values makes our model less sensitive to the changes of the amplitude.



8. **The Cepstrum: Inverse Discrete Fourier Transform:** The iFFT and delta's computations are discussed in part II.

This documents will be continued in part II, in which I will discuss how to include the AM or the same observations' probability (p(O|W) likelihood) in conjugation of the a-priori to calculate the a-posteriori. Then the report follows up describing the search and decoding algorithms.

REFERENCES

1. Martin, James H., and Daniel Jurafsky. "Speech and language processing." International Edition 710 (2000).