CUSTOMIZED LIBRARY IMPLEMENTATION FOR REAL-TIME SPEECH PROCESSING - MFCC

Akhilesh Rao

Ankit Dani

NYU Tandon School of Engineering

Authors note

This report was prepared for DSP LAB I, Fall 2015

Under the guidance of Professor Ivan Selesnick

INDEX

INTRODUCTION

Speech recognition continues to this day to be a major area of research and commercial importance. With the rise of mobile applications and Internet of Things (IoT), the importance of fast and energy-efficient speech recognition algorithms will only continue to grow in the coming decades.

Speaker recognition is the process of identifying a person on the basis of speech alone. It is a known fact that speech is a speaker dependent feature that enables us to recognize friends over the phone. During the years ahead, it is hoped that speaker recognition will make it possible to verify the identity of persons accessing systems; allow automated control of services by voice, such as banking transactions; and also control the flow of private and confidential data. While fingerprints and retinal scans are more reliable means of identification, speech can be seen as a non-evasive biometric that can be collected with or without the person's knowledge or even transmitted over long distances via telephone.

Speech is a complicated signal produced as a result of several transformations occurring at several different levels: semantic, linguistic, articulatory, and acoustic. Differences in these transformations appear as differences in the acoustic properties of the speech signal. Speaker-related differences are a result of a combination of anatomical differences inherent in the vocal tract and the learned speaking habits of different individuals. In speaker recognition, all these differences can be used to discriminate between speakers.

Speaker recognition allows for a secure method of authenticating speakers. During the enrollment phase, the speaker recognition system generates a speaker model based on the speaker's characteristics. For speech/speaker recognition, the most commonly used acoustic features are mel-scale frequency cepstral coefficient (MFCC for short).

MFCC takes human perception sensitivity with respect to frequencies into consideration, and therefore are best for speech/speaker recognition. The MFCC features are extracted from the speaker phonemes in the pre-segmented speech sentences. Now days Prosodic features are currently used in most emotion recognition algorithms Prosodic features are relatively simple in their structures and known for their effectiveness in some speech recognition tasks. There are various ways of generating prosodic syllable contour features that have recently been applied to enhance systems for speaker recognition. Here we will explain the step-by-step computation of MFCC.

THE MFCC USED IN SPEECH RECOGNITION

Generally speaking, a conventional automatic speech recognition (ASR) system can be organized in two blocks: the feature extraction and the modeling stage. In practice, the modeling stage is subdivided in acoustical and language modeling, both based on HMMs as described in Figure 1.

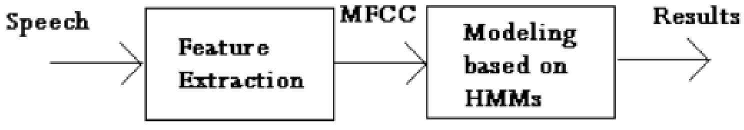

Figure 1- Simple representation of a conventional ASR.

The feature extraction is usually a non-invertible (lossy) transformation, as the MFCC described pictorially in Figure 2. Making an analogy with filter banks, such transformation does not lead to perfect reconstruction, i.e., given only the features it is not possible to reconstruct the original speech used to generate those features. Computational complexity and robustness are two primary reasons to allow loosing information. Increasing the accuracy of the parametric representation by increasing the number of parameters leads to an increase of complexity and eventually does not lead to a better result due to robustness issues. The greater the number of parameters in a model, the greater should be the training sequence. Here one should notice that the goal is not speech compression but using features suitable for speech recognition.

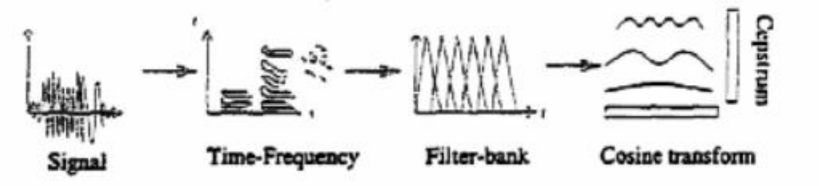

Figure 2- Pictorial representation of Mel-frequency Cepstrum (MFCC) calculation

CALCULATING THE MFCCs

1. Frame the signal into short frames.
2. For each frame calculate the Periodogram estimate of the power spectrum.
3. Apply the Mel filterbank to the power spectra, sum the energy in each filter.
4. Take the logarithm of all filterbank energies.
5. Take the DCT of the log filterbank energies.
6. Keep DCT coefficients 2-13, discard the rest.

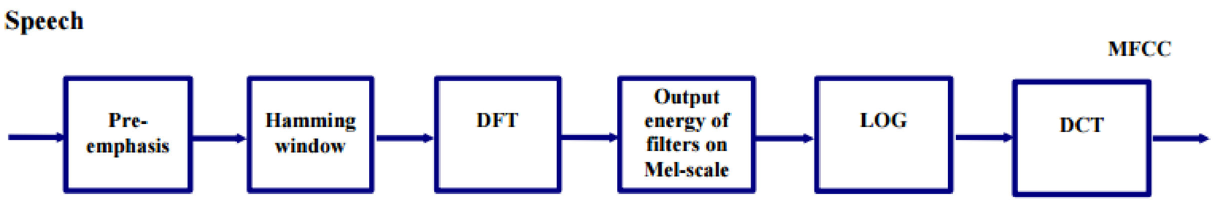

Figure 3 - MFCC calculation

(i). Frame the signal into short frames?

An audio signal is constantly changing, so to simplify things we assume that on short time scales the audio signal doesn't change much (when we say it doesn't change, we mean statistically i.e. statistically stationary, obviously the samples are constantly changing on even short time scales). This is why we frame the signal into 20-40ms frames. If the frame is much shorter we don't have enough samples to get a reliable spectral estimate, if it is longer the signal changes too much throughout the frame.

(ii). Periodogram estimate of the power spectrum for each frame?

The next step is to calculate the power spectrum of each frame. This is motivated by the human cochlea (an organ in the ear) which vibrates at different spots depending on the frequency of the incoming sounds. Depending on the location in the cochlea that vibrates (which wobbles small hairs), different nerves fire informing the brain that certain frequencies are present. Our periodogram estimate performs a similar job for us, identifying which frequencies are present in the frame.

(iii). Application of Mel-filterbank to the power spectra?

The periodogram spectral estimate still contains a lot of information not required for Automatic Speech Recognition (ASR). In particular the cochlea cannot discern the difference between two closely spaced frequencies. This effect becomes more pronounced as the frequencies increase. For this reason we take clumps of periodogram bins and sum them up to get an idea of how much energy exists in various frequency regions. This is performed by our Mel filterbank: the first filter is very narrow and gives an indication of how much energy exists near 0 Hertz. As the frequencies get higher our filters get wider as we become less concerned about variations. We are only interested in roughly how much energy occurs at each spot. The Mel scale tells us exactly how to space our filterbanks and how wide to make them. See below for how to calculate the spacing.

(iv). Logarithm of all filterbank energies?

Once we have the filterbank energies, we take the logarithm of them. This is also motivated by human hearing: we don't hear loudness on a linear scale. Generally to double the perceived volume of a sound we need to put 8 times as much energy into it. This means that large variations in energy may not sound all that different if the sound is loud to begin with. This compression operation makes our features match more closely what humans actually hear. Why the logarithm and not a cube root? The logarithm allows us to use cepstral mean subtraction, which is a channel normalization technique.

(v). DCT of the log filterbank energies?

The final step is to compute the DCT of the log filterbank energies.

There are 2 main reasons this is performed. Because our filterbanks are all overlapping, the filterbank energies are quite correlated with each other. The DCT decorrelates the energies which means diagonal covariance matrices can be used to model the features in e.g. a HMM classifier. But notice that only 12 of the 26 DCT coefficients are kept. This is because the higher DCT coefficients represent fast changes in the filterbank energies and it turns out that these fast changes actually degrade ASR performance, so we get a small improvement by dropping them.

WHAT IS THE MEL SCALE?

The Mel scale relates perceived frequency, or pitch, of a pure tone to its actual measured frequency. Humans are much better at discerning small changes in pitch at low frequencies than they are at high frequencies. Incorporating this scale makes our features match more closely what humans hear.

The formula for converting from frequency to Mel scale is:

$$M(f) = 1125\ \ln(1 + \frac{f}{700})$$

The formula for converting from Mel to frequency scale is:

$$M^{-1}(m) = 700(e^{\frac{m}{1125}} - 1)$$

COMPUTING THE MEL FILTERBANK?

To get the filterbanks we first have to choose a lower and upper frequency. Good values are 300Hz for the lower and 8000Hz for the upper frequency. Of course if the speech is sampled at 8000Hz our upper frequency is limited to 4000Hz. Then follow these steps:

Using equation 1, convert the upper and lower frequencies to Mels.

For example if we want do n filterbanks, we need (n+2) points. This means we need n additional points spaced linearly.

Now use equation 2 to convert these back to Hertz. We will notice that our start- and end-points are at the frequencies we wanted. We don't have the frequency resolution required to put filters at the exact points calculated above, so we need to round those frequencies to the nearest FFT bin. This process does not affect the accuracy of the features. To convert the frequencies to fft bin numbers we need to know the FFT size and the sample rate,

f(i) = floor((nfft+1)*h(i)/samplerate)

Now we create our filterbanks. The first filterbank will start at the first point, reach its peak at the second point, then return to zero at the 3rd point. The second filterbank will start at the 2nd point, reach its max at the 3rd, then be zero at the 4th etc. A formula for calculating these is as follows:

$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

Where M is the number of filters we want, and f() is the list of M+2 Mel-spaced frequencies.

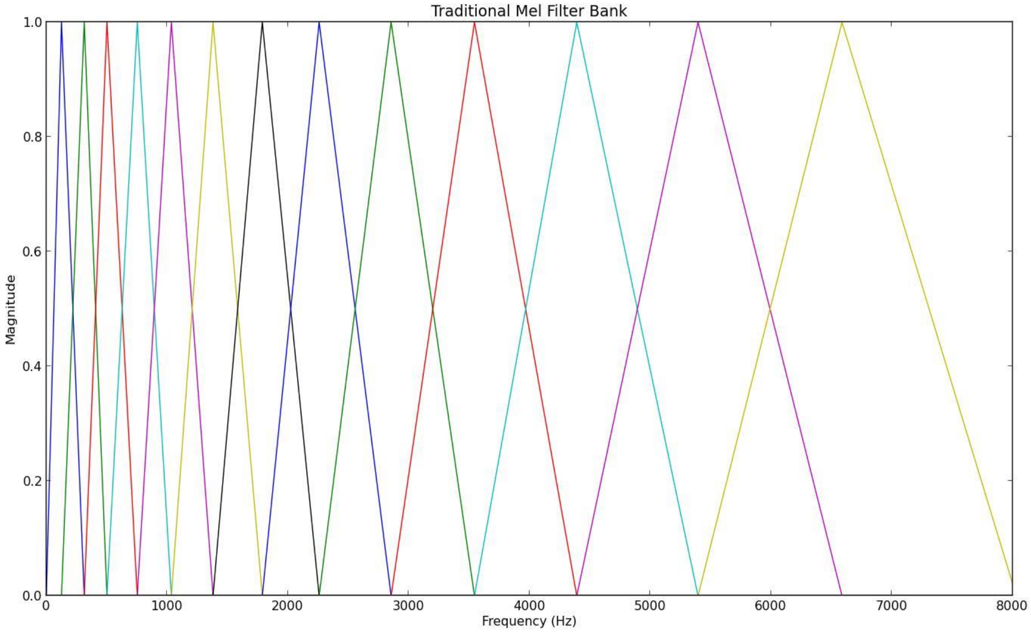

Figure 4 - Mel-Filter Bank

SAMPLE OUTPUTS

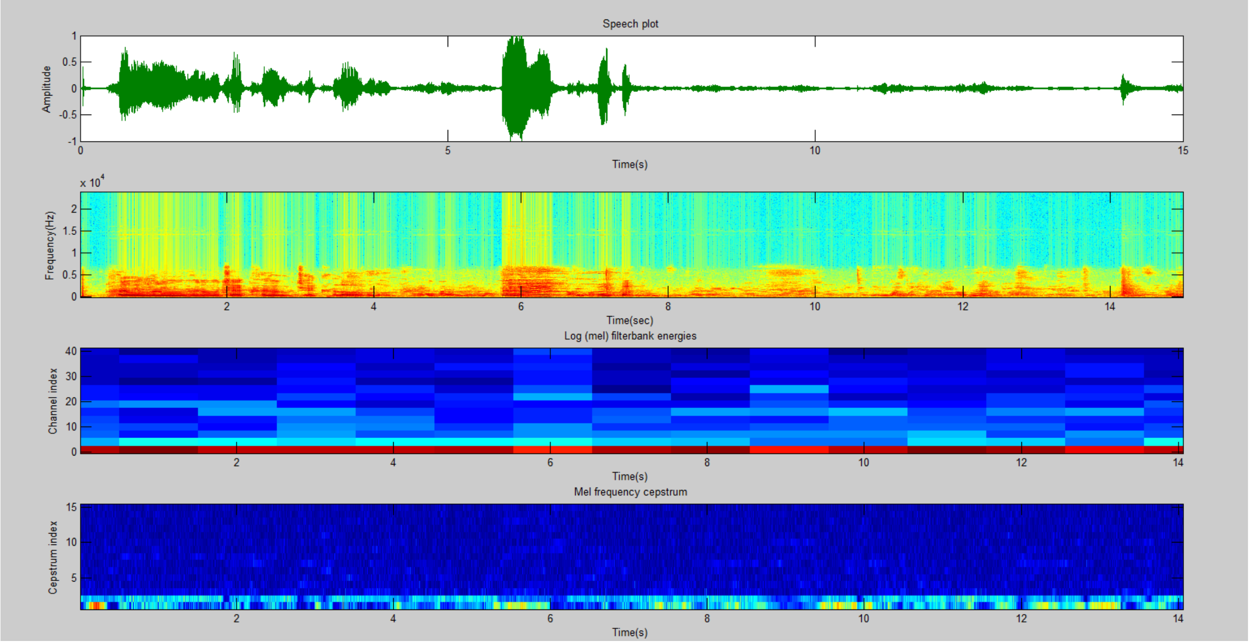

Fig 5 - Plot

PROBLEMS

The reason we cannot get the MFCC plots from time 0-15, although we implement the recording for 15 seconds, is because we only consider the MFCC's from index 2 onwards. The reason we drop the earlier value is because we find it degrades the performance and hence, the plot of the recording. This inturn sets of a chain reaction during the training process and reduces the efficiency of the recognition system.

APPLICATIONS

We can further implement these properties to train data sets and then test it in order to use it for voice or instrument recognition. Applications such as Shazam used for song identification often use this algorithm. Our code bank can be implanted for this purpose.

CONCLUSION

In this report, we try to cover up the basic implementation of MFCC. This work reflects the results obtained in the evaluation of MFCC features, in this study we present the feature extraction techniques for speaker recognition is discussed MFCC and it can be used for speech recognition or voice identification

References

[1] http://www.speech.cs.cmu.edu/15-492/slides/03_mfcc.pdf

[2] http://practicalcryptography.com/

[3] Martin, A. and Przybocki, M., "The NIST 1999 Speaker Recognition Evaluation—An Overview", Digital Signal Processing, Vol. 10, Num. 1-3. January/April/July 2000

[4] John G. Proakis and Dimitris G. Manolakis, "Digital Signal Processing", New Delhi: Prentice Hall of India. 2002.