Albanian j. agric. sci. 2013;12 (2): 305-308 *Agricultural University of Tirana*

RESEARCH ARTICLE (Open Access)

Speech acustic analysis and mfcc extraction

OLTI QIRICI*, RIDI QIRICI

Faculty of Natural Sciences, University of Tirana

Abstract

Hereafter it will be shown a comparative approach toward the speech analysis, between different methods of characteristics extraction which would lead to a better understanding and simpler implementation of a automatic speech recognition system. Different methods are shown till now and all these methods try to rich in a better description of the signal information to be simple enough for implementation and big enough to really show the signal content. Hereby will be shown side by side some of these methods with a better view on the MFCC method which seems to be also the most used till know.

Keywords: MFCC, Cepstrum, frame, windowing, DFT, ASR, LFCC

1. Introduction

Speech Recognition is one of the main activities of our times in Artificial Intelligence which still doesn't meet the results of 100% successful attempt even in recognizing single words. The main reason of such results mainly relates to different articulation of the same word by different persons and differences in the speaking apparatus in each individual. But maybe also relates to techniques which are adapted in the information technologies by other natural sciences.

At an article written by Bogart D. et al. where first stated the principles of speech recognition by starting from usage of Cepstral analysis already used in seismic and geophysical calculations. From there the same idea of using a frequency (quefrency) and spectral (Cepstral analysis was propagated in speech recognition.

In this paper we'll try to describe and summarize the main idea of MFCC extraction which in turn is the feeding information for model training in the process of speech recognition.

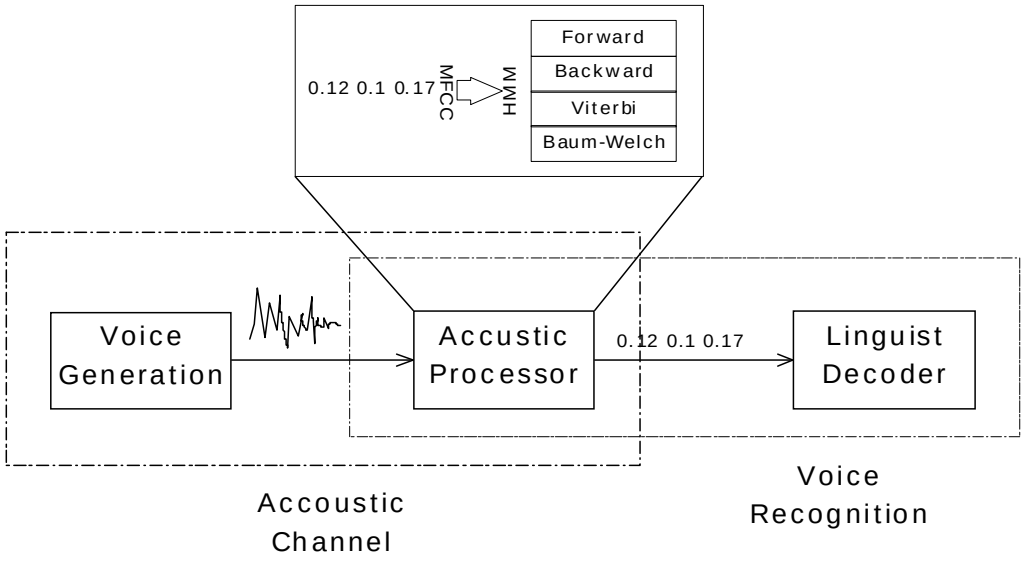

Figure 1: Speech Recognition Process Schema

As a key feature of such process it tries to gather the best information from an audio or voice source which will be periodically sampled and will be produced a array for each extraction segment which will feed the remaining part of the recognizer. The reason of MFCC model choice is purely casual and is one of the three main feature extraction techniques which are: i) MFCC – Mel-Frequency Cepstral Coefficients, ii) PLP – Perceptual Linear Prediction, iii) LPC – Linear Prediction Coefficients, iv) LFCC – Linear Frequency Cepstral Coefficients etc. As results show: "... beyond the common usage of MFCC and PLP spectra, there is no predominant feature choice in the speaker diarization research community." [2]

Correspondence: Olti Wirici, Affiliation University of Tirana; Email: oltiqiric@yahoo.com
(Accepted for publication 30 April 2013)
***ISSN**: 2218-2020, © Agricultural University of Tirana*

So to us is irrelevant the discussion on the choice of MFCC (even if in fact MFCC is the most used method in voice features extraction). Following we will gather and explain mathematical wise and computer science wise the process of MFCC extraction.

2. MFCC Extraction headlines

The MFCC feature is a function which transforms information on audio files or microphone reading to a vector of values easily includable in the learning mechanisms.

The process of saving data to audio files will be first of all a sampling process, data from input (microphone) will be periodically chosen and their values will represent the signal in a file. Since data on signals are parts of a continuous field, values before to be saved in files (files only save discrete values – as integer numbers are) have to be quantized so for a range of values around a significant numbers all these values have to be specified by the representative value. Even if these data aren't saved in files, the same logic is applied for direct listening on microphone data.

Generally the sampling rate of these signals is of 8,000 Hz (telephone speech bandwidth) or 16,000 Hz (microphone bandwidth). Even though is commonly known that:

"Most information in human speech is in frequencies below 10,000 Hz, so a 20,000 Hz sampling rate would be necessary for complete accuracy." [3]

So the most probable sampling rate in automatic speech recognition would be the 16,000 Hz sampling rate.

According to Jurafsky D., and Martin J. [3] the MFCC features extraction passes through seven steps also specified as: i) Preemphasis, ii) Windowing, iii) DFT, iv) Mel Filter Bank and Log, v) The Cepstrum, vi) Deltas and vii) Energy.

Other authors divide the Preemphasis phase from the MFCC process extraction as a preparatory phase and not part of the MFCC phase, anyway let us analyse these phases in the following sessions.

3. Preemphasis and windowing

The idea of preemphasis is related to gathering information on high frequency energy. This energy is very important to better identify the spoken character. The preemphasis can be specified by a high order, high pass filter and can be mathematically identified by:

$$y[i] = x[i] - \alpha * x[i-1] \text{ where } 0.9 \leq \alpha \leq 1.0 \text{ and } i \in [, n-1] \quad [3.1]$$

If we would try to code, by taking in consideration that the whole information we have is included in an array, let say smpl[1000000] we would do the following:

```
const int a = 0.95;  // the middle of the segment
for (int i = 1; i < smpl.length; i++)
preemphasis[i] = smpl[i] – a*smpl[i-1];
```

Afterwards we can start with the windowing process. The windowing process is mainly related to dividing the whole set of data sampled into specific frames. These frames can be even overlapped (frame shift is the time period between two consecutive frames). Since the framing process should include a number of frames while the speech is continuing, a better structure to store such information would be a matrix. Let us suppose we have L columns in the matrix (of course we are using a default value since the system doesn't know in advance the length of such text). Let us have an β sampling length frame and the λ frame ($0 \leq \lambda < L$) we would have:

$$x_\lambda[i] = x[i + \beta * \lambda],\ i \in [, n-1] \quad [3.2]$$

so programmatically speaking for the first time we have to take the first β samples and then follow up by shifting each time. In order to develop a windowing process we have also to use a characteristic known as windows shape. Actually there are two most known shapes in windowing which are i) rectangular, ii) hamming shape. Following will be introduced the mathematical description of these shapes:

$$\text{Rectangular} \rightarrow w[i] = \{1 \text{ if } 0 \leq i < L,\ 0 \text{ if otherwise}\} \quad [3.3]$$

$$\text{Hamming} \rightarrow w[i] = \{0.54 - 0.46 \cos(2\pi i/L) \text{ if } 0 \leq i < L,\ 0 \text{ if otherwise}\} \quad [3.4]$$

In order to follow up with the window programming we will use this formula:

$y[i] = w[i] * s[i]$ where $s[i]$ is the value of signal at time i but which recently changed through the preemphasis process.

```
const double PI = 3.14159;
int L=16;
int windowing[L] [B] = {0};
// L frames and B samples per frame
char shape = 'r';
// 'r' – rectangle, 'h' – hamming
lambda = 0;
while (lambda<L){
    for (int i = 0; i<B; i++){
```

```
    if (shape == 'r')
    windowing[lambda][i] = preemphasis[i + B*lambda];
    else
    windowing[lambda][i] = preemphasis[i + B*lambda]*( 0.54 – 0.46 cos (2*PI*(i+ B*lambda) /L));
    }
    lambda++;
    }
```

4. Discrete Fourier Transform and the Cepstrum

"The tool for extracting spectral information for discrete frequency bands for a discrete time (sampled) signal is the discrete Fourier transform or DFT" [3]

The DFT process will be fed with the results coming from the previous step (from windowing) and for each window will be provided a complex number representing the magnitude and phase of a each signal.

In order to specify the DFT for a frequency set we can use the formula (for each frequency n ex. n=1, 2, ... , 24):

$$X[k] = \Sigma_{i=0}^{N-1} x[i]*e^{-j2\pi ki/N} \qquad [4.1]$$

But this calculation would be time consuming. So in these cases, if the number of samples for frame is a power of 2 value it can be used for such calculations a better approach that is the FFT or Fast Fourier Transform.

In order to follow up let us argument regarding the Cepstrum. We are trying to overpass for now the Mel Filter Bank and Log (point no. 4) in order to return again at this concept in a while.

"For discrete time signal… the cepstrum of a signal is the inverse discrete-time Fourier transform (IDTFT) of the logarithm of the magnitude of the discrete-time Fourier transform (DTFT) of the signal." [5]

And in this case for IDTFT (shortly IDFT) we will have the following formula:

$$c[n] = \Sigma_{i=0}^{N-1} log(\mid \Sigma_{i=0}^{N-1} x[i]*e^{-j2\pi ni/N} \mid) \, e^{j2\pi ni/N} \qquad [4.2]$$

In both these formulas the Euler formula could be specified (in all these formulas j is the imaginary part of the complex number):

$$e^{j\theta} = cos\ \theta + j\ sin\ \theta \qquad [4.3]$$

which would lead in the fact that:

$e^{j2\pi ni/N} = cos\ (2\pi ni/N) + j\ sin\ (2\pi ni/N)$ and $e^{j2\pi ni/N} = cos\ (-2\pi ni/N) + j\ sin\ (-2\pi ni/N)$ [4.4]

Till here all the possible literature was converging in the same formulas but, the formulas of DFT and IDFT found above (taken from the interpretation as in Jurafsky's book) are different from the formula's found in Rabiner's book. As Rabiner is underlining the exact formula to calculate the MFCC is:

$$mfcc[n] = (1/N)* \Sigma_{i=1}^{N} log(1/(\Sigma_{k=Ln}^{Un| } V_n[k]|^2) * \Sigma_{k=Ln}^{Un} | V_n[k]*X_m[k]|^2) \, cos(2\pi ni/N + \pi ni/N) \qquad [4.5]$$

where Ln, Un are respectivelly Lower and Upper frequencies for a representative frequency n. Regarding the $V_n[k]$ here expressed the formula to calculate would be:

$$V_n[k] = 1127\ ln\ (1 + k/700) \qquad [4.6]$$

Actually even if it seems we are using "Magic" values this would be the calculation method for the skipped step called above Mel Filter Bank and Log.

"A mel is a unit of pitch defined such that pairs of sound which are perpetually equidistant in pitch are separated by an equal number of mel" [3]

As we demonstrated above there are different ways to calculate the MFCC values each of which has it's own calculation difficulty.

Let us try to build a coding structure for the DFT transformation:

In order to specify all above code, here is shown the Cepstrum code built in MATLAB taken from [4]:

```
mfcc = (abs (ifft (abs (fft (hamming (length (segment)).* segment)))));
```

Calculations on this phase would be very long so we are over passing the conversion of the above formulas in code. In fact this depends on us but in different implementations there is different number of coefficients in the MFCC. In Sphinx-III this number is 40 mel filters to produce 14 mel coefficients, in Jurafsky's book the number of mel coefficients is 12 and in other implementations this number could be different.

5. Deltas and Energy

Form previous extraction we found 12 coefficients. Now we have to add a thirteenth called the energy coefficient. This coefficient is calculated through the formula below and represents the power in time through a frame:

$$E = \Sigma_{i=t1}^{t2} x^2[i] \qquad [5.1]$$

And in order to conclude with calculations we have to conclude with deltas and double deltas representing specifically velocity and acceleration in passing from frame to frame in a window. So to calculate these values we have to calculate the difference of the Cepstral values and the difference of these differences. The same has to be applied in the energy coefficients.

The general formulas would be for a particular i formula starting from 0 to 12:

Δ(i) = (mfcc[i+1] - mfcc[i-1]) / 2 [5.2]

And

Δ Δ (i) = (Δ (i+1) - Δ (i-1)) / 2 [5.3]

The frames are concatenated with one another so the last frame of the previous window could be the i-1 index of the new window. The first frame can have 0 as the i-1 index.

6. Conclusions

The calculation of MFCC is very heavy and sometimes differs from a source to another. The time consuming in this code calculation is very big and in ASR it takes the vast majority of time in calculation. All this calculation is done frame by frame so every time we try to calculate values for a frame so we can recognize the letter or sound behind this we have to do this calculation, meanwhile the same calculation has to persist in time and techniques are trying to help decrease amount of calculations (as LFCC). But none of the other calculation techniques have better effectively in speech recognition. Still perhaps the technique needs improvement because still ASR precision is far from outstanding.

7. References

1. Bogert B, Healy M J R, Tukey J W: **The quefrency alanysis of Time Series for Echoes**, Wiley, 1963: 209-243.
2. Gold B, Morgan N, Ellis D: **Speech and Audio Signal Processing, Second Edition**, Wiley – Interscience, 2011: 300-305, 644-653.
3. Jurafsky D, Martin J H: **Speech and Language Processing, Second Edition**, Pearson Prentice Hall, 2009: 295-302, 230-241.
4. McLoughlin I: **Applied speech and audio processing, Second Edition**, Cambridge University Press, 2009: 25-30.
5. Rabiner L, Schafer R: **Theory and applications of digital speech processing, First Edition**, Pearson Prentice Hall, 2011: 413-480.