SPEECH MODELS: MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

Michele Rossi

rossi@dei.unipd.it

Dept. of Information Engineering

University of Padova, IT

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Outline

- MFCCs: what are they for?
- The 6 steps to compute them (intuition)
- The Mel scale for human perceived frequency
- The 6 steps in greater detail

MFCC – what are they for?

- First step in any speech recognition system is
  - To extract features: should be good for identifying the linguistic content (neglecting background noise, emotion, etc.)
- Sounds generated by humans
  - Are filtered by the shape of the vocal tract, including tongue, teeth, etc.
  - This shape determines which sound comes out
  - If we determine this shape accurately, this should give a good representation of the phoneme that is being produced
- Shape of the vocal tract
  - Manifest itself in the envelope of the *short-term power spectrum*
  - The job of MFCCs is to accurately represent this envelope [1]

[1] S. Davis, P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sequences," *IEEE Transactions on Acoustic, Speech and Signal Processing*, Vol. 28, No. 4, Aug. 1980.

Their computation involves 6 steps

1. Frame the signal into short frames
2. For each frame: calculate its power spectrum
3. Apply the Mel filterbank to the power spectrum
   - 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 others

Step 1 - framing

- An audio signal is constantly changing (non-stationary)
- It is difficult to deal with non-stationary data

- To simplify things
  - We assume that on short time scales the signal does not change much
  - “Doe not change” means it is statistically stationary

- This is why we frame the signal into 20-40ms time windows
  - Standard value is 25ms
  - Shorter values: not enough samples to get a reliable spectral estimate
  - Larger values: the signal (statistics) changes too much in the frame

Step 2 – power spectrum

- Compute the power spectrum of a frame
  - This is motivated by the human cochlea (organ in the ear)
  - Which vibrates at different spots depending on the frequency of the incoming sound waves
  - Depending on the location of the cochlea that vibrates, different nerves fires informing the brain that certain frequencies are present
- The periodogram estimate of the power spectrum
  - Does a similar job
  - Informing us which frequencies are present in the current frame

Step 3 – filterbank

- The periodogram estimate of the power spectrum
  - Still contains a lot of information that is not required by Autonomous Speech Recognition (ASR) systems
  - In fact, the cochlea cannot discern the difference between two closely spaced frequencies
  - This effect becomes more apparent as frequencies increase
- Hence:
  - We take clumps of periodogram bins and sum them up to get an *idea of how much energy there exists in each region*
  - This is performed using a Mel filterbank
    - First Mel filter is very narrow: to get an idea of how much energy there is around 0 Hertz
    - Filters get wider as frequencies get higher: we become less concerned about variations (Mel scale tells us exactly how to space filterbanks)

Step 4 – logarithm

- Once we have the filterbank energies
  - We take their logarithm
  - This is also motivated by human hearing: we do not 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 that different* if the sound is loud
- Hence
  - This compression operation (logarithm) makes our features match more closely to what humans actually hear

Steps 5 and 6 – DCT

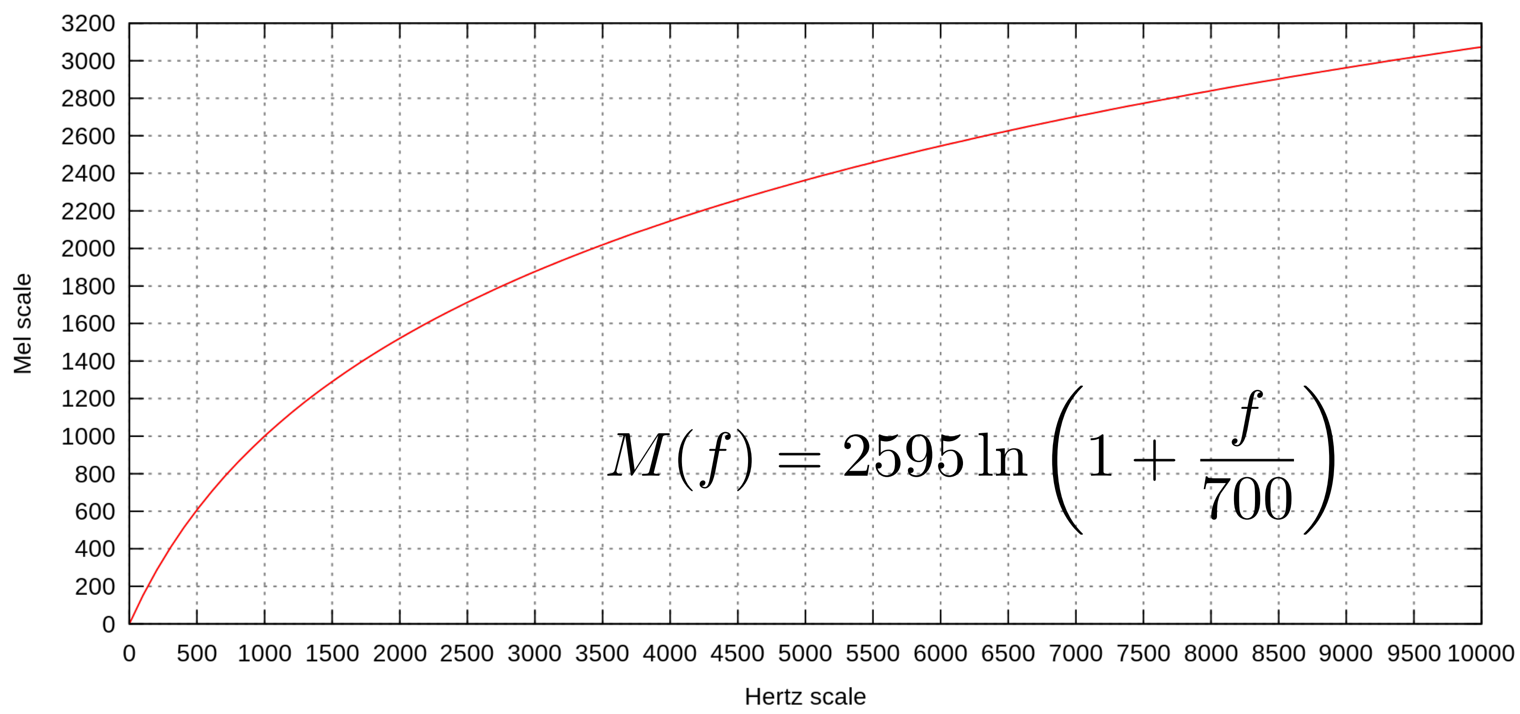
- Discrete Cosine Transform (DCT)
  - There are two main reasons for applying it

  - First: filterbanks are quite overlapping (meaning that filterbank energies are quite correlated with each other)
    - DCT decorrelated filterbank energies
    - This means that diagonal covariance matrices can be used to model the features with GMM and/or GMM/HMM systems (very desirable)
    - Also (energy compaction property of DCT) we obtain a compact representation in the first DCT coefficients

  - Second: discard some information for improved performance
    - Only 12 of the 26 DCT coefficients are kept
    - Higher DCT coefficients represent fast changes in the filterbank energies
    - These fast changes degrade ASR performance
    - A (small) improvement is achieved by dropping them

The Mel scale - M(f)

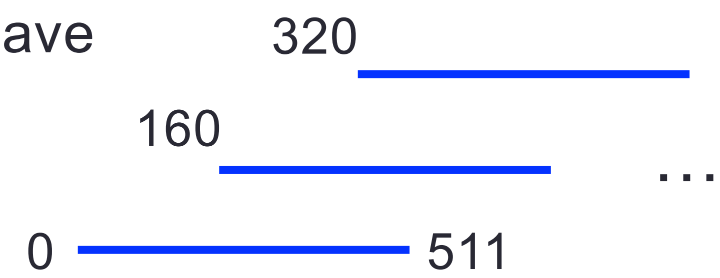
- Relates the *human perceived frequency* (or pitch) of a pure tone to its *actual measured frequency*
- Humans are much better at discerning small changes in pitch at low frequencies that they are at high frequencies
- Incorporating it makes ASR match more closely what humans hear



STEPS 1-6 IN MORE DETAIL

Step 1 - framing

- Frame the audio signal into [20,40] ms frames (standard value is 25 ms)
- For a 20.480 kHz signal, with 25 ms we have
  - 20,480 samples per second
  - 0.025 * 20,480 = 512 samples per frame



- Frame step is usually, e.g., 10 ms (160 samples)
  - This allows for some overlapping between subsequent frames
  - First frame (of 512 samples) starts at sample 0
  - Second frame (still of 512 samples) start at sample 160, etc.
  - Keep framing until the end of the speech signal
  - If the signal does not divide into an even number of frames, pad it with zeros so it does
- Notation
  - $s(n)$ is the audio signal (time domain signal)
  - $s_i(n)$ is frame i with n ranging from 0 to 511 in our example

Step 2 – power spectrum (1/4)

- For each frame $s_i(n)\,,\, n = 0, 1, \ldots, N-1\,(N = 512)$
  - Calculate the complex DFT (Discrete Fourier Transform)

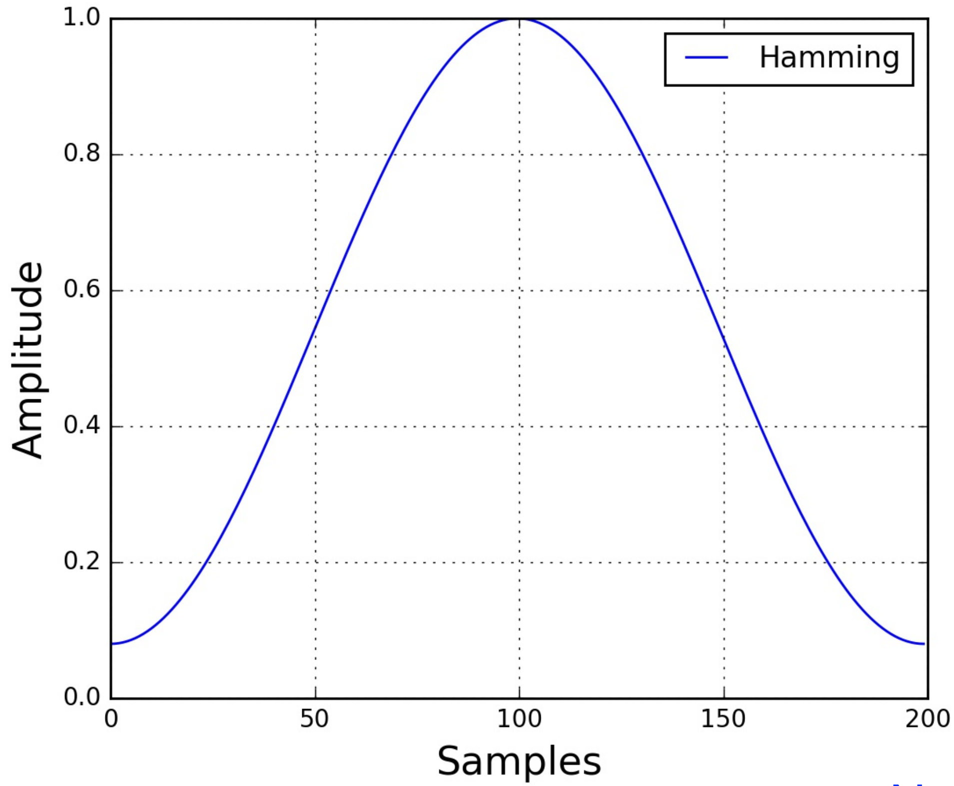
$$S_i(k) = \sum_{n=0}^{N-1} s_i(n) w(n) e^{-j2\pi kn/N}\ ,\ k = 0, 1, \ldots, N-1$$

  - DFT
    - Converts a time sequence of N equally spaced samples into an N-long complex sequence, which is a complex-valued function of frequency
    - The DFT (with w(n)=1) completely describes the discrete Fourier transform onto an N-periodic time sequence
    - When applying DFT, we are implicitly applying it to an infinitely repeating (periodic) signal. However, if this is not the case, e.g., first and last samples of $s_i(n)$ do not match, this is *interpreted as a discontinuity* in the signal and *generates a lot of high frequency response in the transformed signal*, that we do not want → use of "windowing" w(n)

Step 2 – power spectrum (2/4)

- Often used: Hamming window - w(n)

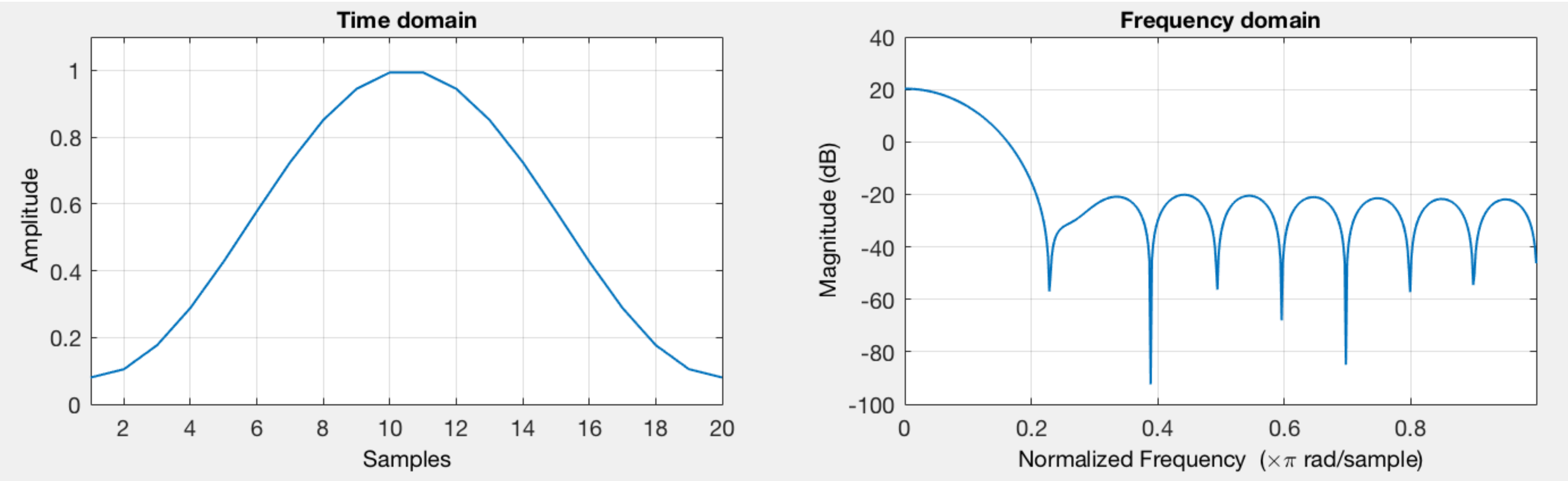
$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right), \; n = 0, \ldots, N-1$$



N-1

Step 2 – power spectrum (3/4)

- Hamming window – transformation (with N=20)



  - The center of the main lobe of a smoothing window occurs at each frequency component of the input signal (of course, freq. plot is symmetric around zero)
  - Center lobe width (across negative & positive freqs.) is $4\Omega_N=4(2\pi/N)$
  - Ideally, only the main lobe should be emphasized while the side lobes should be zeroed (this prevents energy leakage across subsequent frequencies)

Step 2 – power spectrum (4/4)

- Note
  - Among the N=512 complex DCT values
  - Only the first 1+N/2 values are significant, the remaining ones are complex conjugates of the first 1+N/2 values (this is how DCT works)
  - Hence, the first 1+N/2=257 elements *are kept*, while the remaining ones (N/2-1 values) *are discarded*
- Compute the periodogram estimate of the power spectrum:

$$P_i(k) = \frac{|S_i(k)|^2}{N}, \; k = 0, \ldots, N/2$$

Remember that: k=0 is the DC component (frequency f=0). Then, with DCT the step size if F_s/N (F_s is the *sampling frequency*). Hence, k=1 corresponds to frequency $f_1=F_s/N$ and element k=N/2 corresponds to frequency $f_{N/2}=F_s/2$ (this is the, so called, *Nyquist critical frequency*, i.e., the highest that can be represented by sampling at frequency F_s)

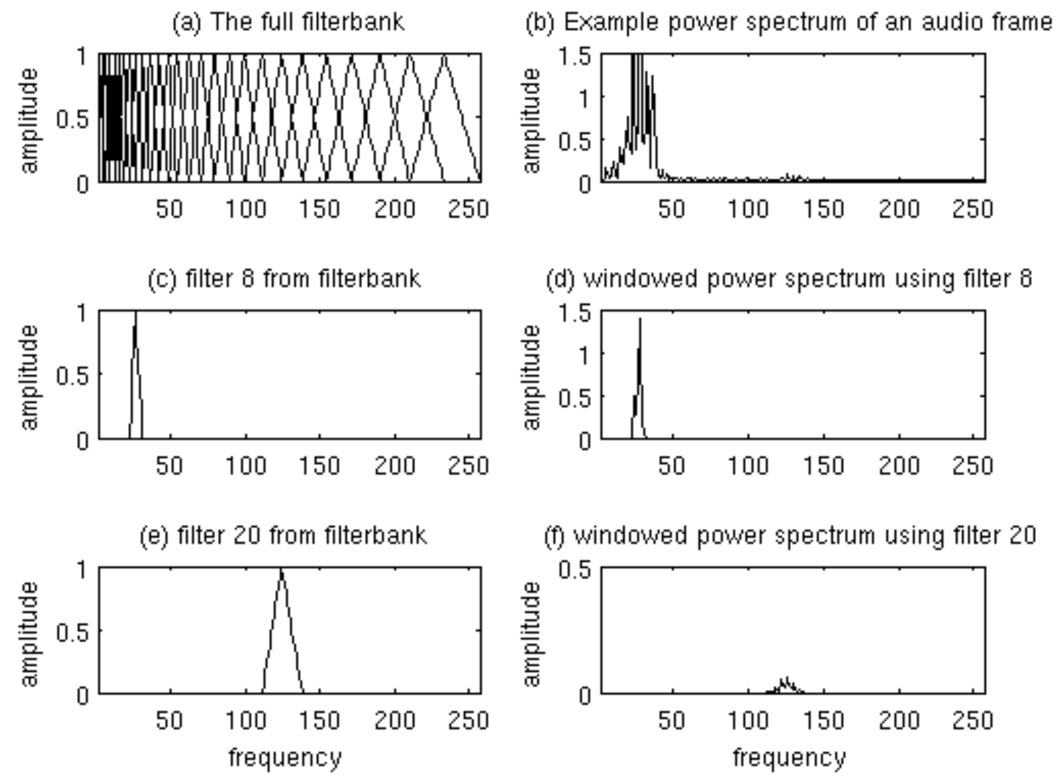
Step 3 – filterbanks (1/3)

- Compute Mel-spaced filterbanks (through Mel's equation)
  - They amounts to 20-40 triangular filters
  - Each filter (in our case) has 257 values (to match the output of step 2) and it is multiplied by the entire power spectrum from step 2

full filterbank (left)
full power spectrum (right)

filter 8 from filterbank (left)
output when this filter is applied (right)

filter 20 from filterbank (left)
output when this filter is applied (right)



Step 3 – filterbanks (2/3)

- Obtain N_{fb} Mel-spaced triangular filters
  - Pick the number of filters (usually N_{fb}=26)
  - Set the maximum frequency as $F_{max}=F_s/2$
  - Set the minimum frequency as, e.g., F_{min}=300 Hz (user defined)
  - Compute N_{fb} center frequencies f(1), f(2), …, f(N_{fb})
    - Linearly spaced in Mel's domain
    - Non-linearly spaced in frequency domain [Hz]
- Each filter m
  - Is centered at frequency f(m)
  - Is equal to 1 for f(m) and decreasing linearly otherwise
  - Is zero at f(m-1), f(m+1) and outside [f(m-1), f(m+1)]
  - N_{fb} filters require N_{fb}+2 thresholds

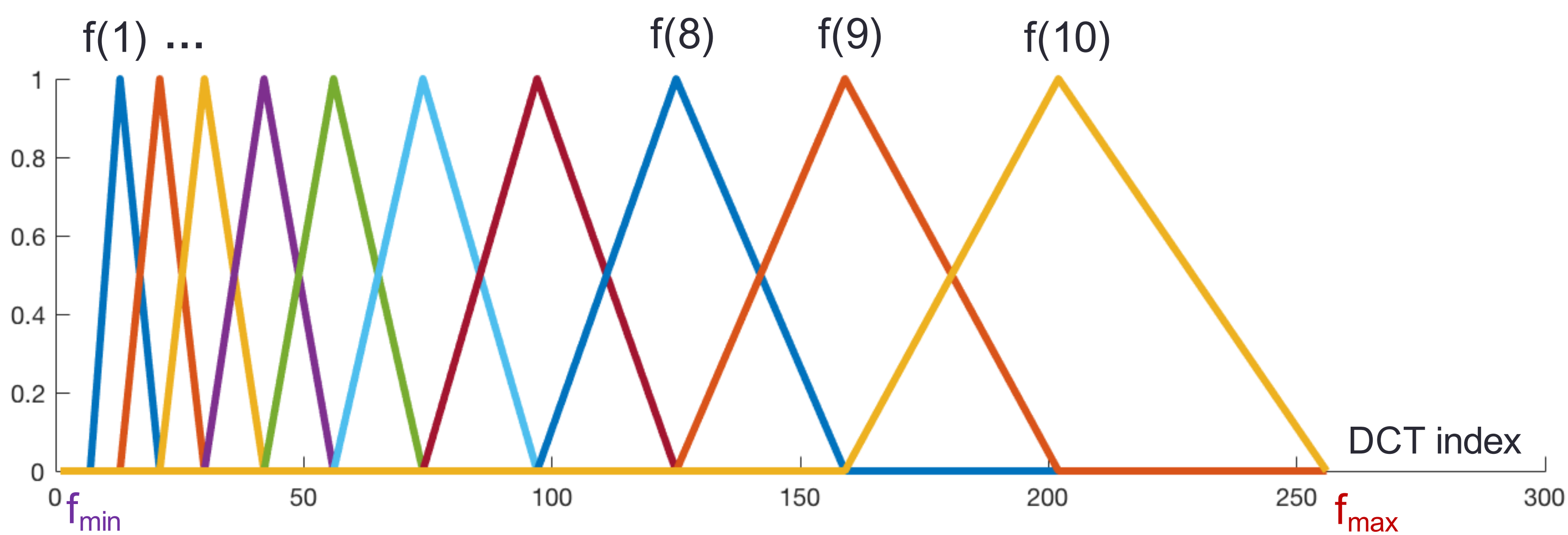
Step 3 – filterbanks (3/3)

- Example
  - F_s= 20.480 kHz, N=512, N_{fb}=10, f_{max}=Fs/2=10,240, f_{min}=300 Hz
- Frequencies [Hz]

300=f_{min} 543=f(1) 845=f(2) 1,220=f(3) 1,687=f(4) 2,267=f(5) 2,988=f(6) 3,883=f(7) 4,997=f(8) 6,381=f(9) 8,102=f(10) 10,240=f_{max}

- Corresponding DCT indices

[7 13 21 30 42 56 74 97 125 159 202 256]



Step 4 – logarithms

- We now have N_{fb} Mel-spaced triangular filters
- For each filter m=1,2,…, N_{fb}
  1. Multiply filter m by the full power spectrum from step 2 (the resulting energy trace will be mostly 0 (besides around f(m), the filter acts as a mask, centered at f(m))
  2. Add up the non-zero coefficients in this trace → E_m
  3. Take the logarithm: $\log(E_m)$

- This gives us N_{fb} real values:
- $\log(E_1)$, $\log(E_2)$, …, $\log(E_M)$, with $M = N_{fb}$
  - One for each Mel's frequency band
  - They tell how much energy there is within each band

Steps 5 and 6

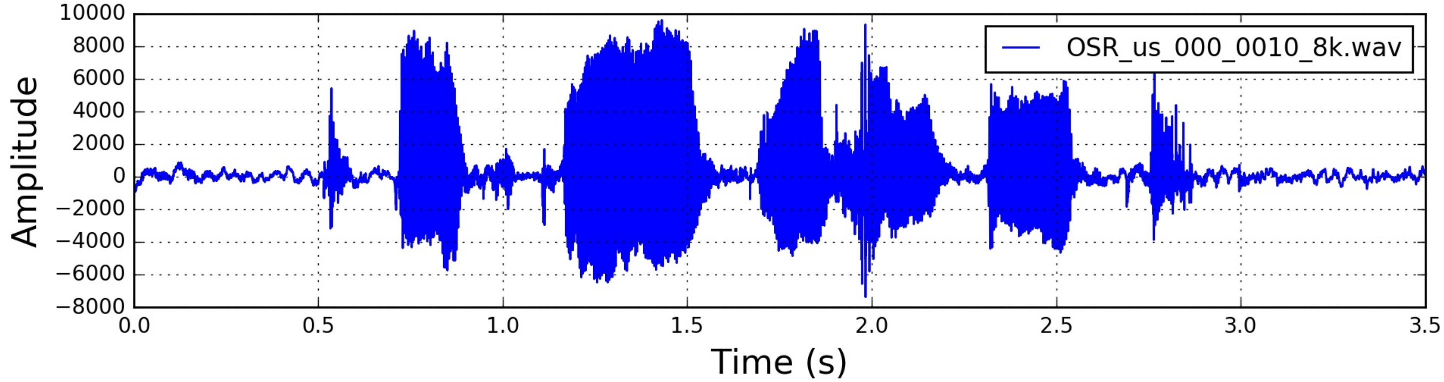
- Step 5 – for each frame
  - Take the DCT (Discrete Cosine Transform) of the vector containing the logarithms of the energy within each band
  - The result of the DCT is a vector of N_{fb} cepstral coefficients

- Step 6 – for each frame
  - For ASR, the cepstral coefficients no. 2,3, …,13 are kept
  - The remaining ones are discarded
  - Note that: the first DCT coefficient is the sum of all the log-energies computed at the previous step (by the very def. of DCT) – thus, it is an overall measure of signal loudness and is not very informative - it is often discarded for *speech recognition* or *speaker id applications* where the system has to be robust to loudness variations

- Final result of this processing is
  - 12 cepstral coefficients for each frame

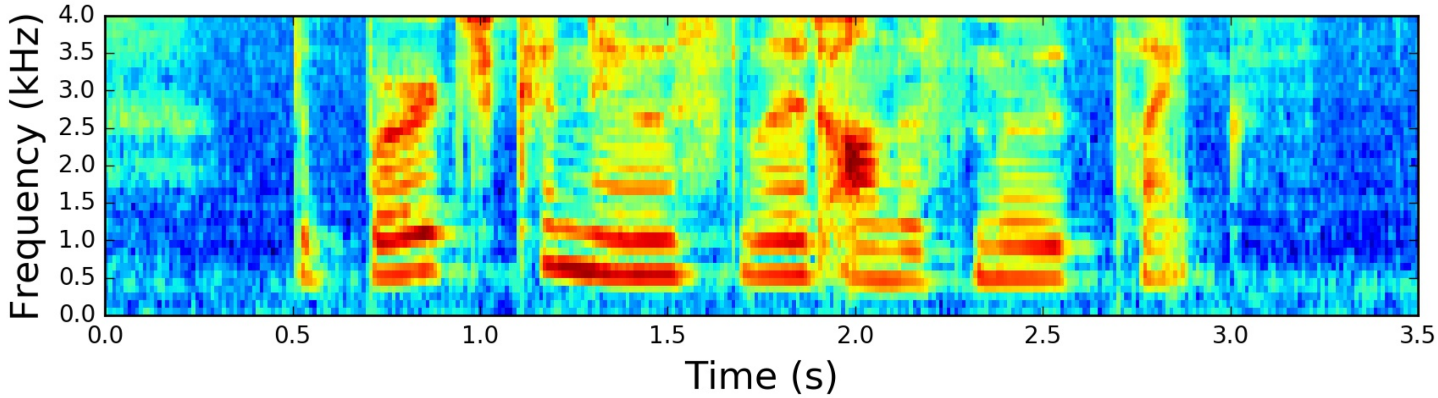
Some visual insight (1/2)

F_s = 8 kHz time signal



Signal in the Time Domain

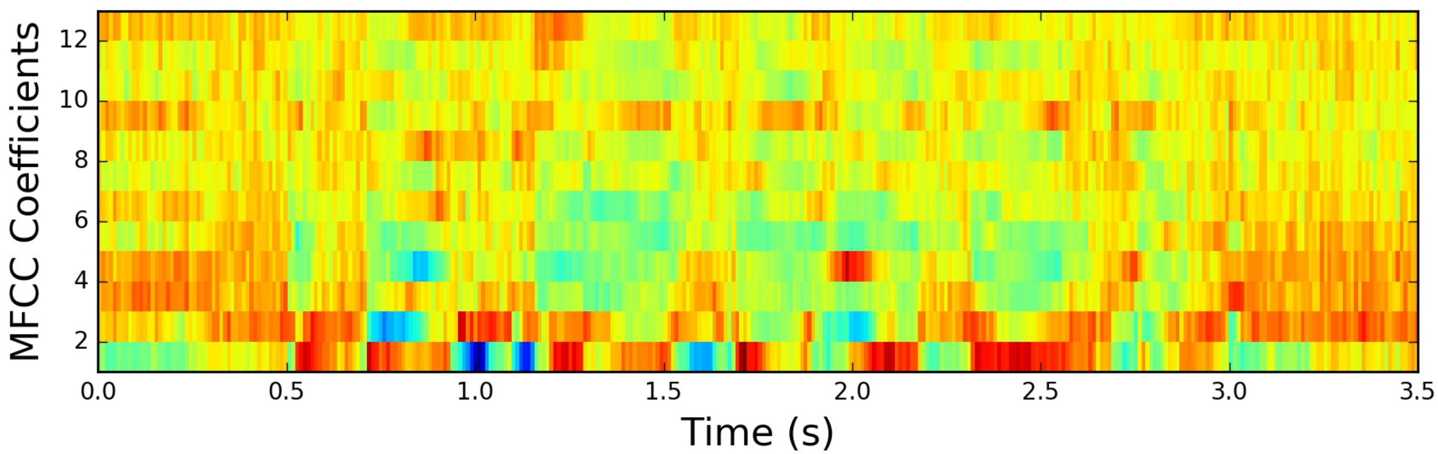
After applying the filterbank to the power spectrum (periodogram):



Spectrogram of the Signal

Some visual insight (2/2)

Mel's Frequency Cepstral Coefficients (MFCCs):



MFCCs

Phyton library to extract MFCCs

https://github.com/jameslyons/python_speech_features

APPENDIX A

Matlab code to compute filterbanks

Filterbanks Matlab code (1/2)

```
Fs=20480;                  % sampling rate
nDFT=512;                  % number of DFT samples
Fnyq=Fs/2;                 % Nyquist frequency

Nfb=10;                    % number of filterbanks to generate
Fmin=300;                  % min frequency of filterbanks [Hz]
Fmax=Fnyq;                 % max frequency of filterbanks [Hz]

FminMel=1125*log(1+(Fmin/700));   % min Mel's frequency
FmaxMel=1125*log(1+(Fmax/700)); % Max Mel's frequency

Nthresh=Nfb+2;             % number of required frequency thresholds

step=(FmaxMel-FminMel)/(Nthresh-1); % step size (linearly partition Mel's space)

% allocate memory for frequency thresholds
melvec=zeros(1,Nthresh);
freqvec=zeros(1,Nthresh);
f=zeros(1,Nthresh);

% for each frequency threshold do
for i = 0:(Nthresh-1)
   melvec(i+1) = FminMel+i*step;                    % assign Mel thresholds (linearly)
   freqvec(i+1) = 700*(exp(melvec(i+1)/1125)-1);    % compute corresponding frequencies (Mel’s transformation)
   f(i+1) = floor((nDFT+1)*freqvec(i+1)/Fs);        % express frequency thresholds in terms of DFT indeces
end
```

Filterbanks Matlab code (2/2)

```
H=zeros(Nfb,Nthresh); % filterbank matrix, here we store the Nfb filters

% for each filter m do (m is the filterbank index)
for m = 2:(Nfb+1)
    for k=1:f(Nthresh)
        if      ((k>=f(m-1)) && (k<=f(m))) H(m-1,k) = (k-f(m-1))/(f(m)-f(m-1)); % increase
        elseif  ((k>=f(m)) && (k<=f(m+1))) H(m-1,k) = (f(m+1)-k)/(f(m+1)-f(m)); % decrease
        else    H(m-1,k) = 0;
        end
    end
end

clf
figure(1);
hold on
for m = 1:Nfb
 % read Filterbank matrix by row, each row contains a filter
 plot(H(m,:),'LineWidth',3)
end
```

SPEECH MODELS: MEL FREQUENCY CEPSTRAL COEFFICIENTS (MFCC)

Michele Rossi
rossi@dei.unipd.it

Dept. of Information Engineering
University of Padova, IT