EE269
Signal Processing for Machine Learning

Cepstrum

Instructor : Mert Pilanci

Stanford University

October 11 2021

Linear systems and additive noise

- Linear systems, e.g., filters, can easily separate **additive noise** from useful information when we know the frequency range of the noise and information

$$y[n] = x[n] + w[n]$$

- In vector notation

$$Hy = Hx + Hw$$

Multiplicative or convolutive noise

- This is harder if the signal and noise are **convoluted**, e.g., in speech processing

$$y[n] = x[n] * w[n]$$

- $w[n]$ is the flowing air (noise source)
- $h[n]$ is the vocal tract (filter)

We can develop an operator that can separate convoluted components by **transforming convolution into addition**

Cepstrum

- Developed to separate convoluted signals

$$y[n] = x[n] * w[n]$$

Discrete Fourier Domain:

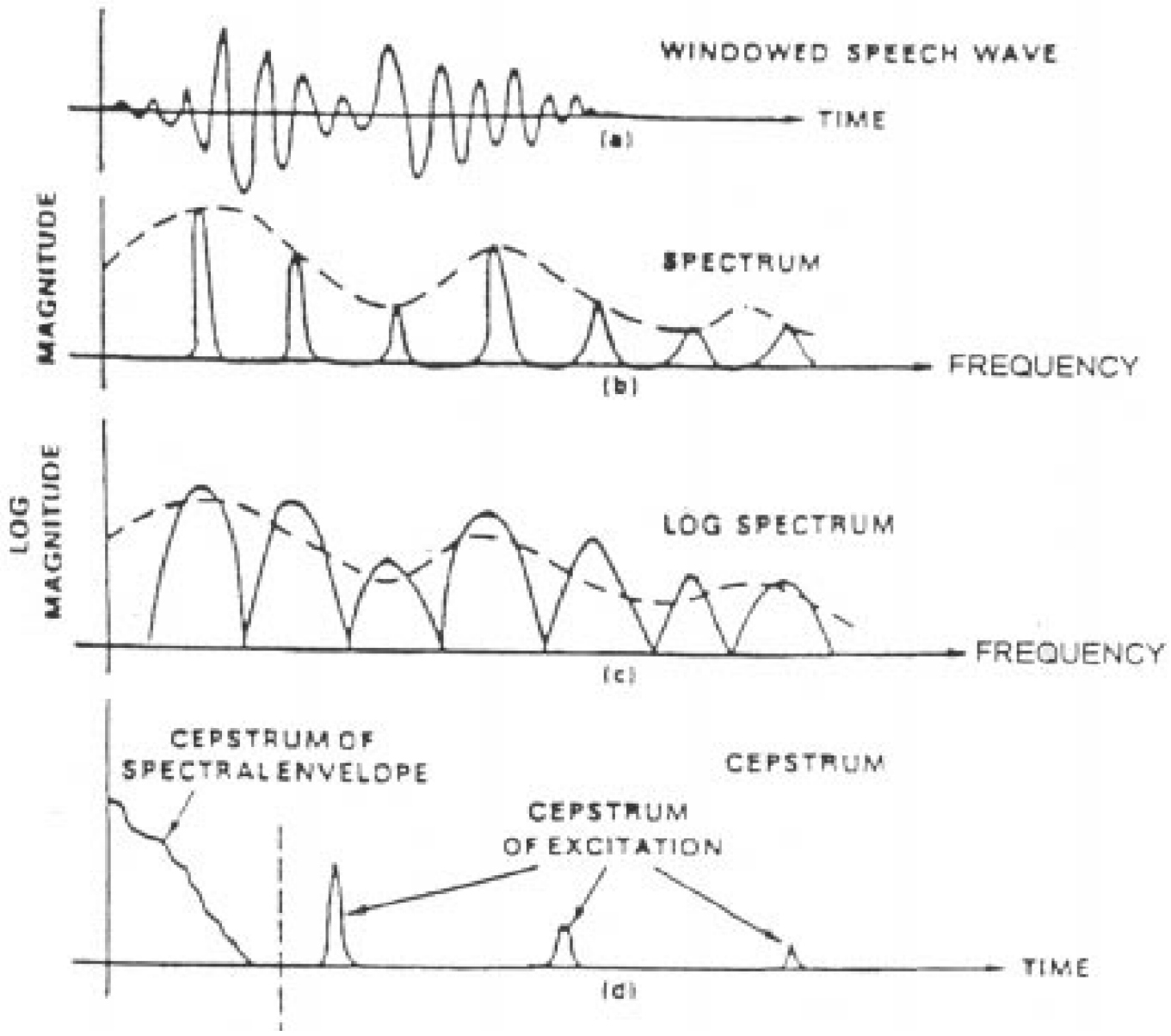
$$Y[k] = X[k]W[k]$$

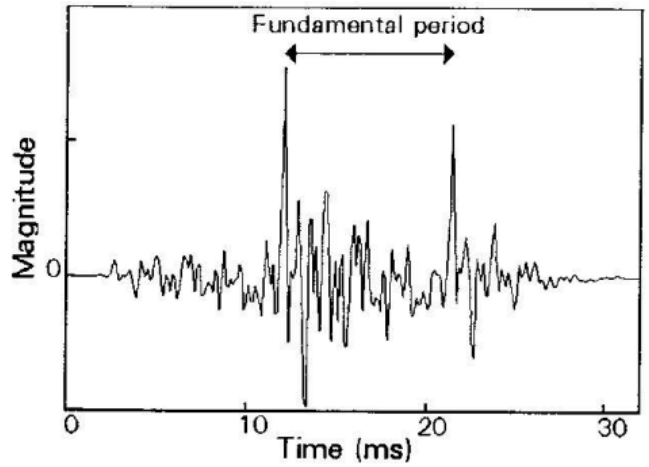
- Take logarithms

$$\log[Y[k]] = \log X[k] + \log W[k]$$

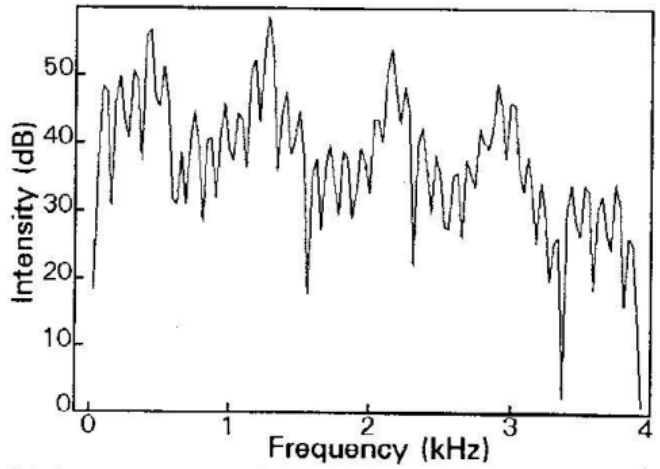
- we can apply a linear filter to $\log Y[k]$ to separate
- equivalently we can take DFT of $\log Y[k]$ and process in frequency domain

  cepstrum is the DFT (or DCT) of the log spectrum

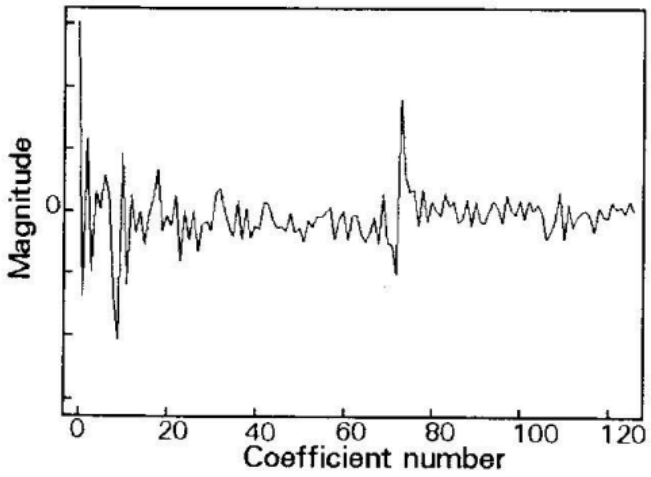

WINDOWED SPEECH WAVE
TIME
(a)
MAGNITUDE
SPECTRUM
FREQUENCY
(b)
LOG MAGNITUDE
LOG SPECTRUM
FREQUENCY
(c)
CEPSTRUM OF SPECTRAL ENVELOPE
CEPSTRUM
CEPSTRUM OF EXCITATION
TIME
(d)



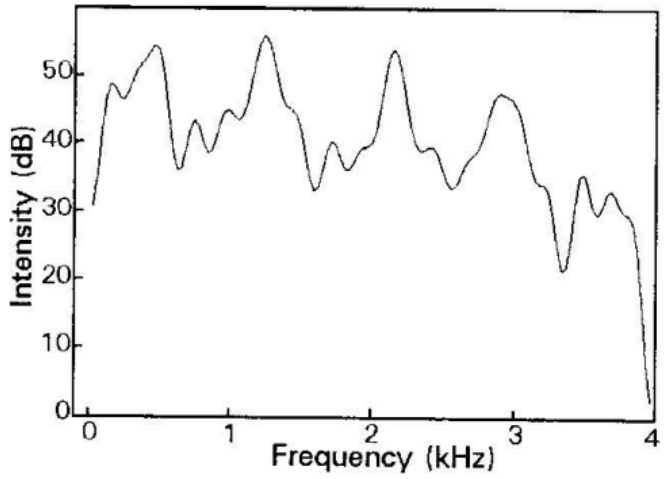
(a) Windowed speech waveform (32 ms at 8 kHz sampling rate).



(b) Log spectrum (from a Fourier transform).

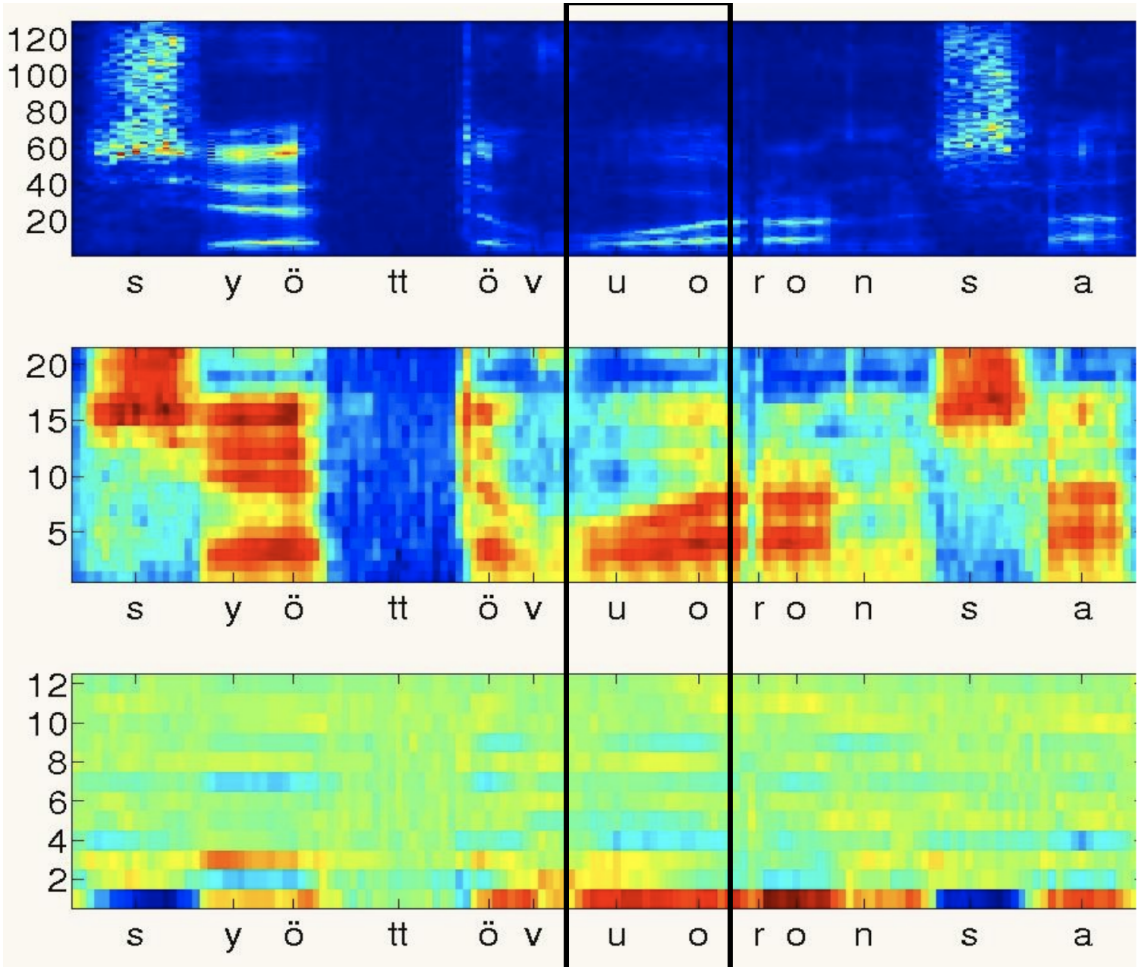


(c) Cepstrum computed from the log spectrum shown in (b).



(d) Log spectrum reconstructed from the first 40 cepstral coefficients in (c).

Figure 10.3 Analysing a section of speech waveform to obtain the cepstrum and then to reconstruct a cepstrally smoothed spectrum.



1. Frames:
short 10ms
windows
2. FFT:
power spectrum
spectrogram

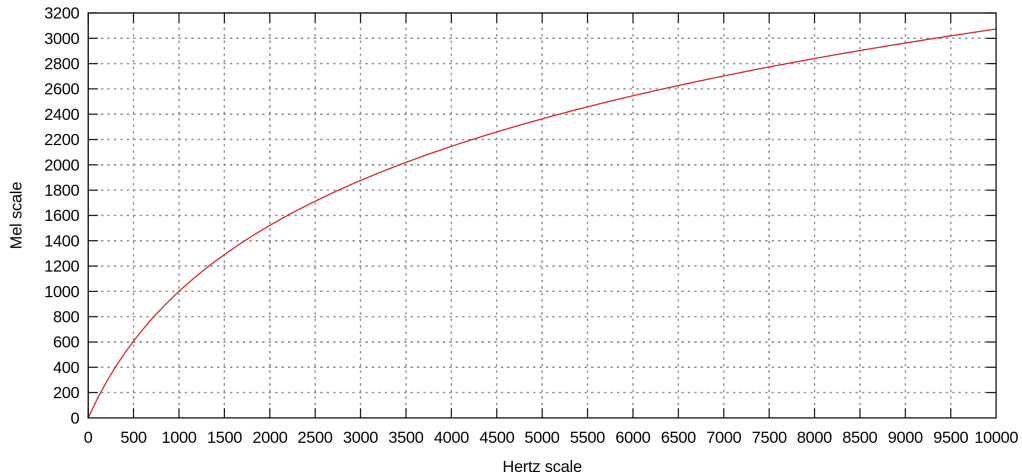
3. Filtering:
mel filter
motivated by
human ear
"essential data"

4. Features:
DCT transform
mel cepstrum
MFCC
-less features
-less correlation

Application: Mel-frequency spectrum

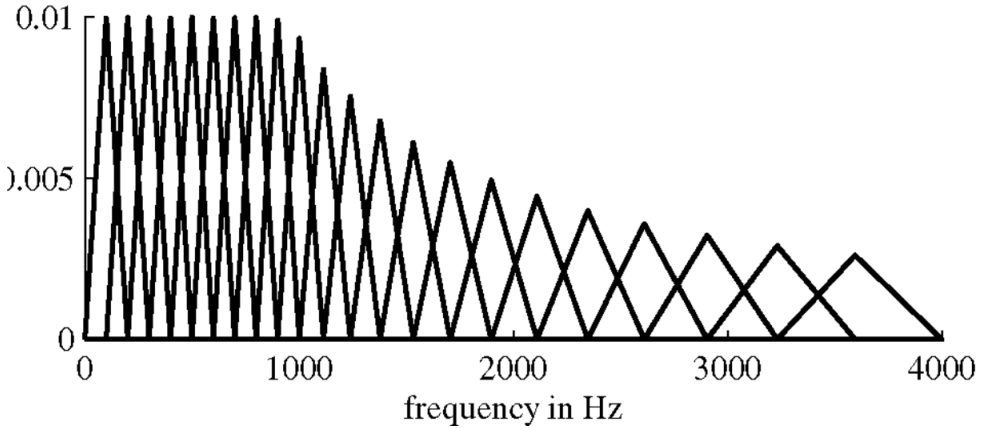
- perceptual scale of pitches
- 1 mels $=$ 1000 Hz
- a formula to convert f hertz into m mels

$$m = 2595 \log_{10}\left(1 + \frac{f}{700}\right)$$



Application: Mel-frequency spectrum

- weighted DFT magnitude
- mel-frequency spectrum $MF[r]$ is defined as

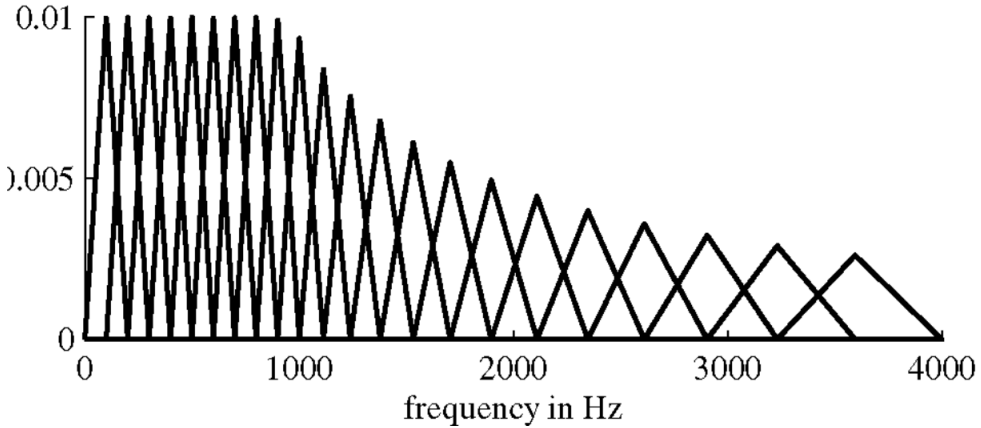


$$MF[r] = \sum_k |V_r[k]X[k]|^2$$

- $V_r[k]$ is the triangular weighting function for the rth filter.
- bandwidths are constant for center frequencies ¡ 1kHz and then increase exponentially
- identical to convolutions with 22 filters

Application: Mel-frequency spectrum

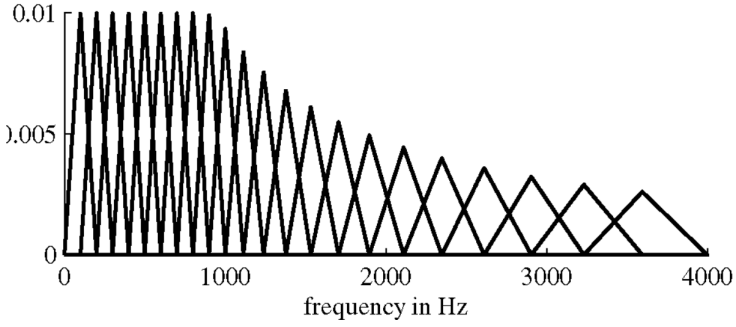
- weighted DFT magnitude
- mel-frequency spectrum $MF[r]$ is defined as



$$MF[r] = \sum_k |V_r[k]X[k]|^2$$

- $V_r[k]$ is the triangular weighting function for the rth filter.
- bandwidths are constant for center frequencies ¡ 1kHz and then increase exponentially
- identical to convolutions with 22 filters

Application: Mel-frequency spectrum



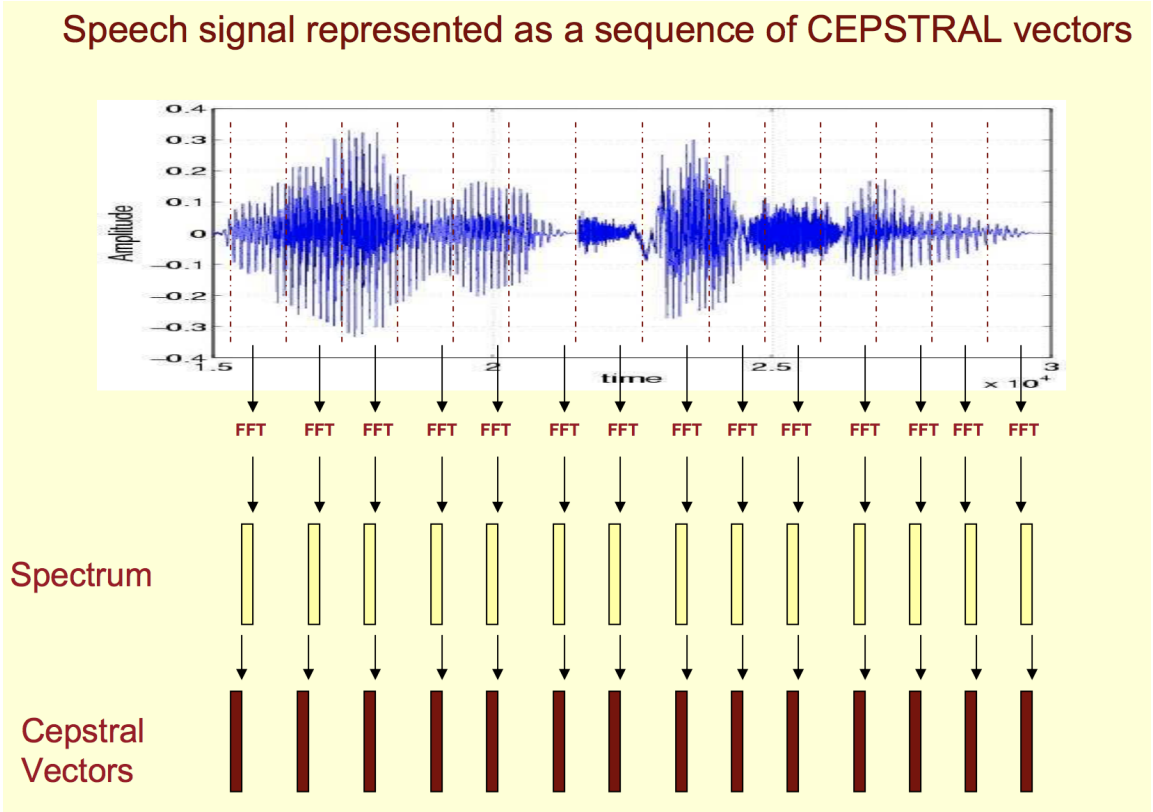
$$\mathrm{MF}[r] = \sum_k |V_r[k]X[k]|^2$$

- Mel Frequency Cepstral Coefficient (MFCC)

$$\mathrm{MFCC}[m] = \sum_{r=1}^{R} \log(\mathrm{MF}[r]) \cos\left[\frac{2\pi}{R}\left(r + \frac{1}{2}\right)m\right] \qquad (1)$$

- i.e., inner-product with cosines $\mathrm{MFCC}[m] = \langle \log \mathrm{MF}[r], c_m[r] \rangle$

Application: Speaker Identification



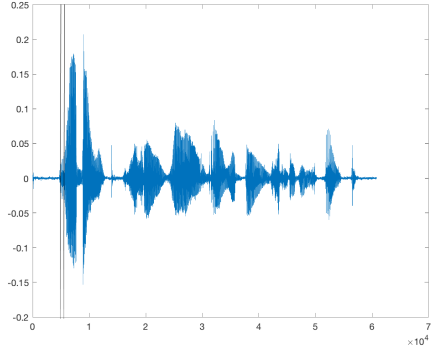
- train a k-Nearest Neighbor classifier to classify frames

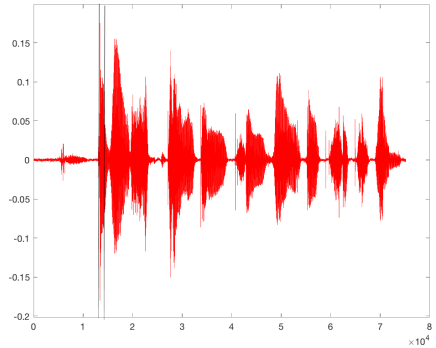
Application: Speaker Identification

- AN4 dataset (CMU): 5 male and 5 female subjects speaking words and numbers
- collect the training samples into frames of 30 ms with an overlap of 75%
- calculate MFCC
- train a k-Nearest Neighbor classifier on the frames
- for a given test signal, predictions are made every frame
- most frequently occurring label is declared as the speaker

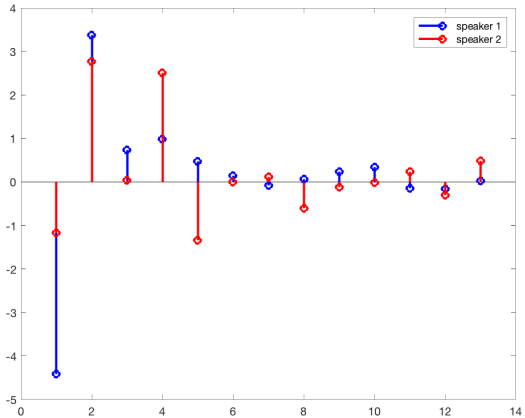
Application: Speaker Identification

speaker 1 (blue) and speaker 2 (red) time domain signals

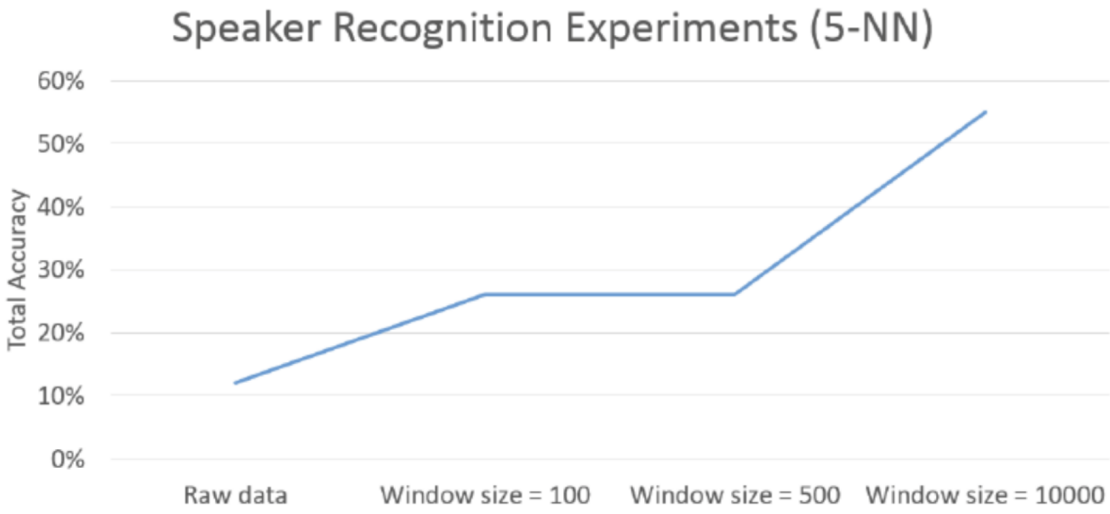




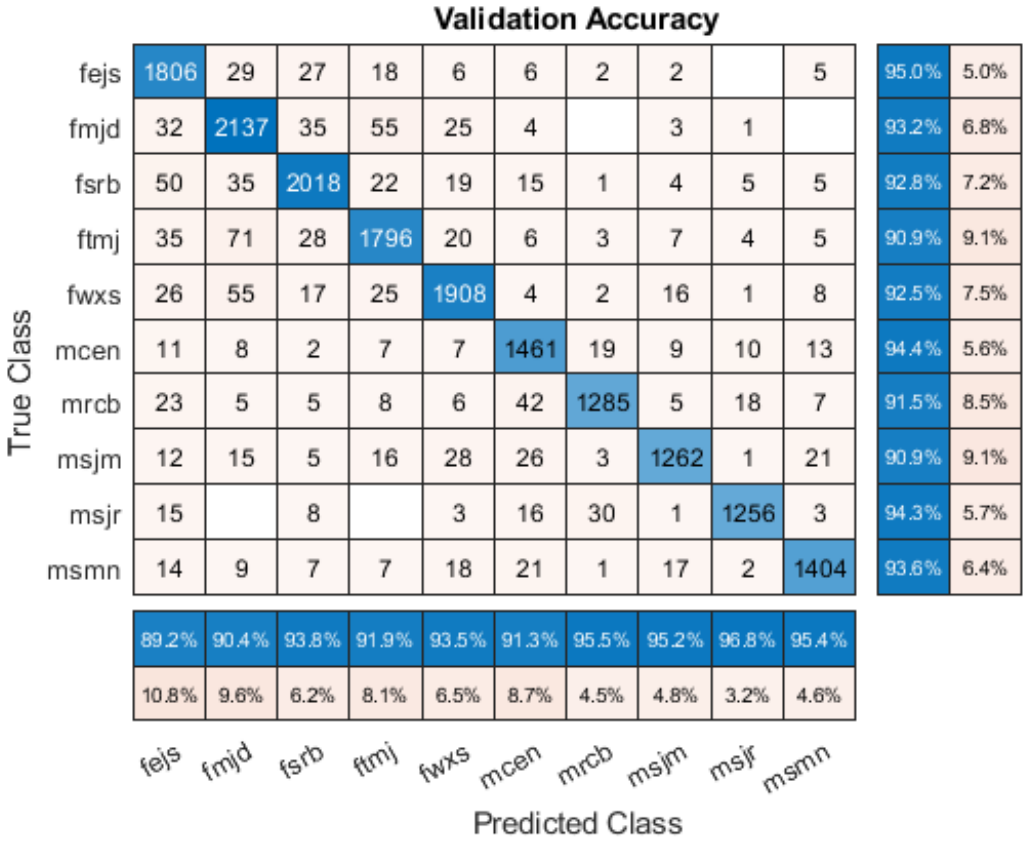
frame based MFCC features



Application: Speaker Identification



Application: Speaker Identification



- average accuracy is 92.93%