# DT2119 Lab1: Feature Extraction

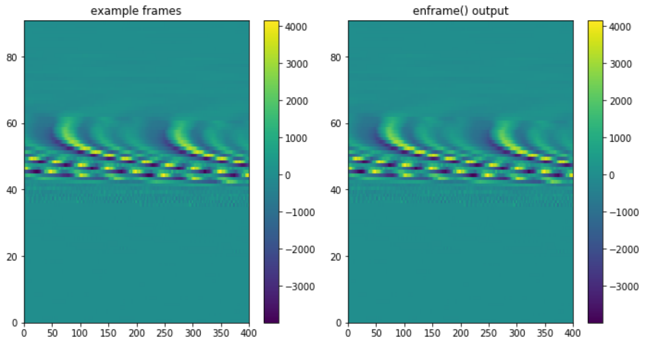
**Zhanpeng Xie, zxie@kth.se**

### 4.1 Enframe

Sampling Rate = 20000, Window Length = 20 ms, Shift Length = 10ms

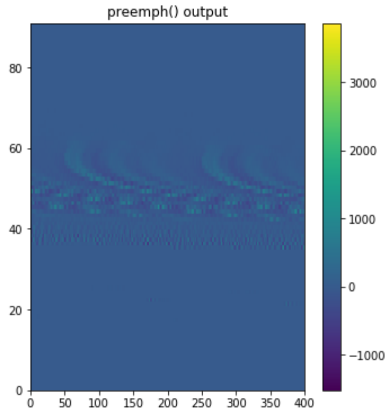
Window Length Samples = (20*10-3)/(1/20000) = 400

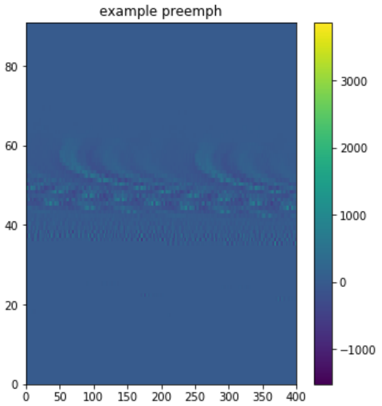
Shift Length Samples = (shift length / window length) * Window Length Samples = 200



The result of enframe() is the same as example['frames'].

### 4.2 Pre-emphasis





According to our course slides, the pre-emphasis filter is:

$$y[n] = x[n] - \alpha x[n-1], \quad \text{with } \alpha = 0.97$$

And lfilter() is implemented as:

$$\begin{aligned} y[n] &= \frac{1}{a_0}\Big(b_0 x[n] + b_1 x[n-1] + \cdots + b_P x[n-P] + \\ &\quad -a_1 y[n-1] - a_2 y[n-2] - \cdots + a_Q y[n-Q]\Big) \\ &= \frac{1}{a_0}\left(\sum_{i=0}^{P} b_i x[n-i] - \sum_{j=1}^{Q} a_j y[n-j]\right) \end{aligned}$$

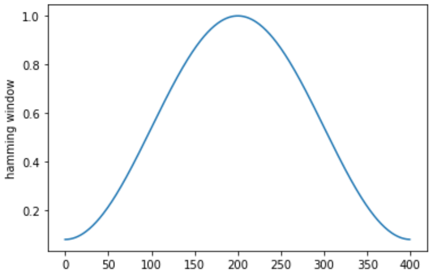
$$\begin{aligned} a &= [a_0, a_1, \ldots, a_Q] \\ b &= [b_0, b_1, \ldots, b_P] \end{aligned}$$

Therefore, a is set to be [1] and b is [-1, -preempcoeff]

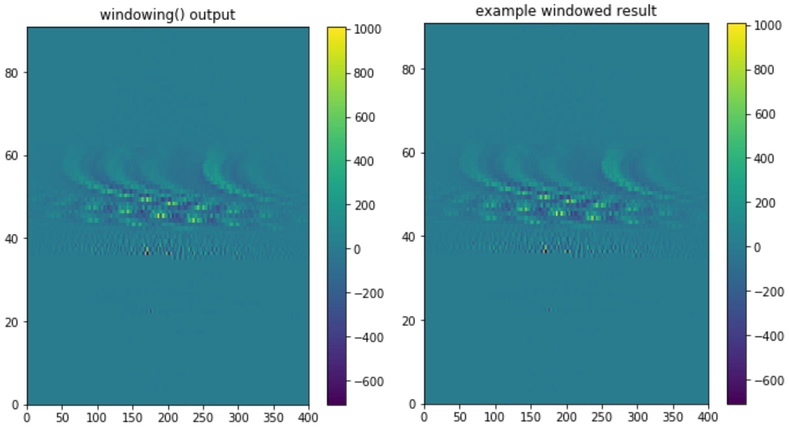
### 4.3 Hamming Window

Reason for using Hamming window: Selecting a sample from the whole speech signal is equivalent to apply a square windowing function to the signal. However, square windowing function in frequency domain has oscillations which means its frequency is influenced by other frequencies. Hamming window has much smoother bell shape in frequency domain and can avoid such problems.

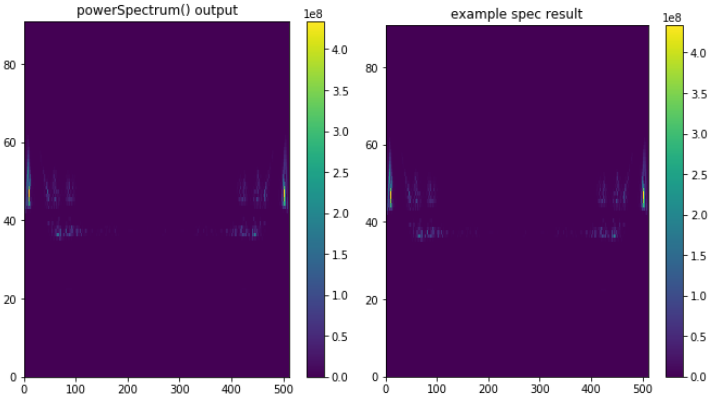
Hamming window:



Windowed results:



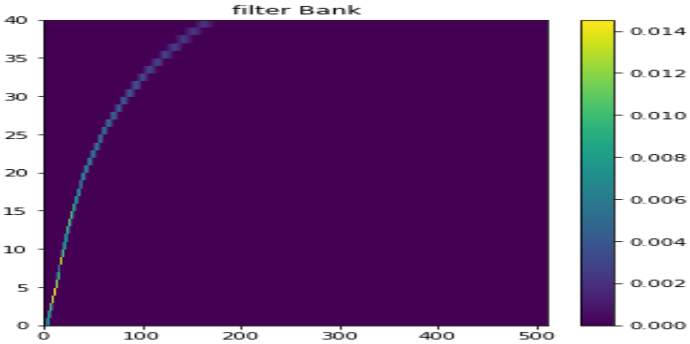
**4.4 Fast Fourier Transform**



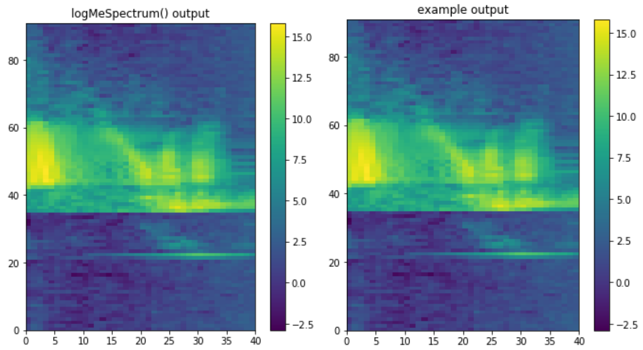
The signal’s sampling rate is 20KHz. Therefore, the $f_{max} = 10kHZ$.

4.5 Mel filterbank log spectrum

Most filters are focusing on the lower frequency part and filters for high frequency can sample a wider range of frequency but have much lower values.

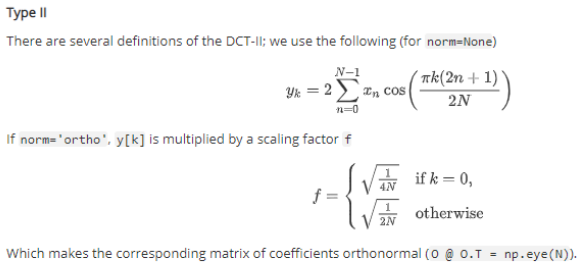


Result comparison:



### 4.6 Cosine Transform and liftering

Explanation for different results when using n=13:



**Type II**

There are several definitions of the DCT-II; we use the following (for `norm=None`)

$$y_k = 2\sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi k(2n+1)}{2N}\right)$$

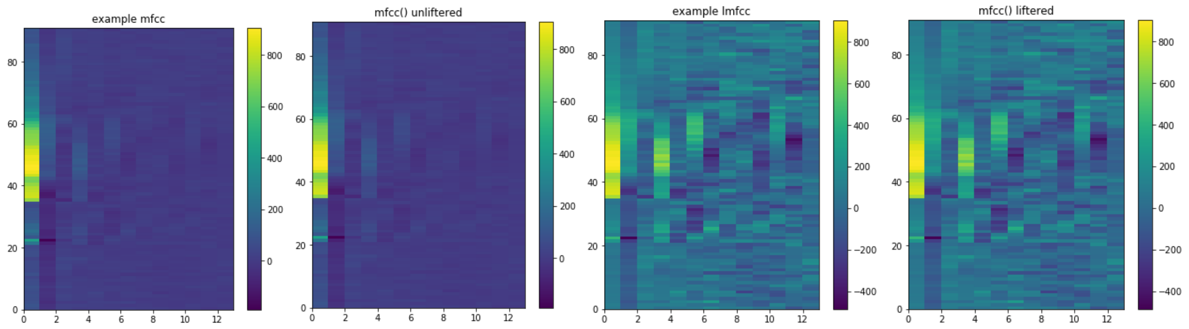
If `norm='ortho'`, `y[k]` is multiplied by a scaling factor `f`

$$f = \begin{cases} \sqrt{\frac{1}{4N}} & \text{if } k = 0, \\ \sqrt{\frac{1}{2N}} & \text{otherwise} \end{cases}$$

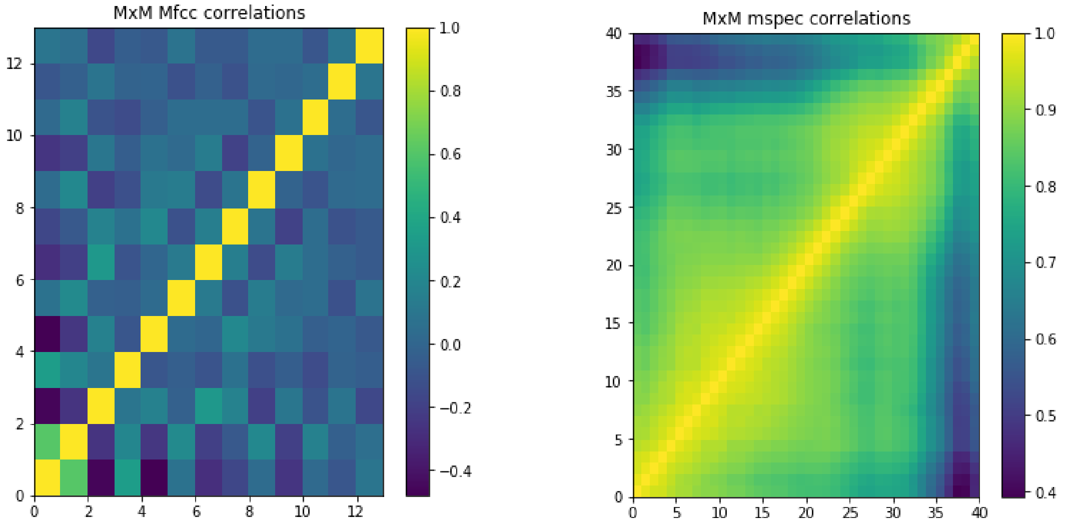
Which makes the corresponding matrix of coefficients orthonormal (`O @ O.T = np.eye(N)`).

The default value for n is equal to the number of filter bank channels. Therefore, $y_k$ corresponds to different cosine functions (different frequencies) depending on the value of n.

Results:

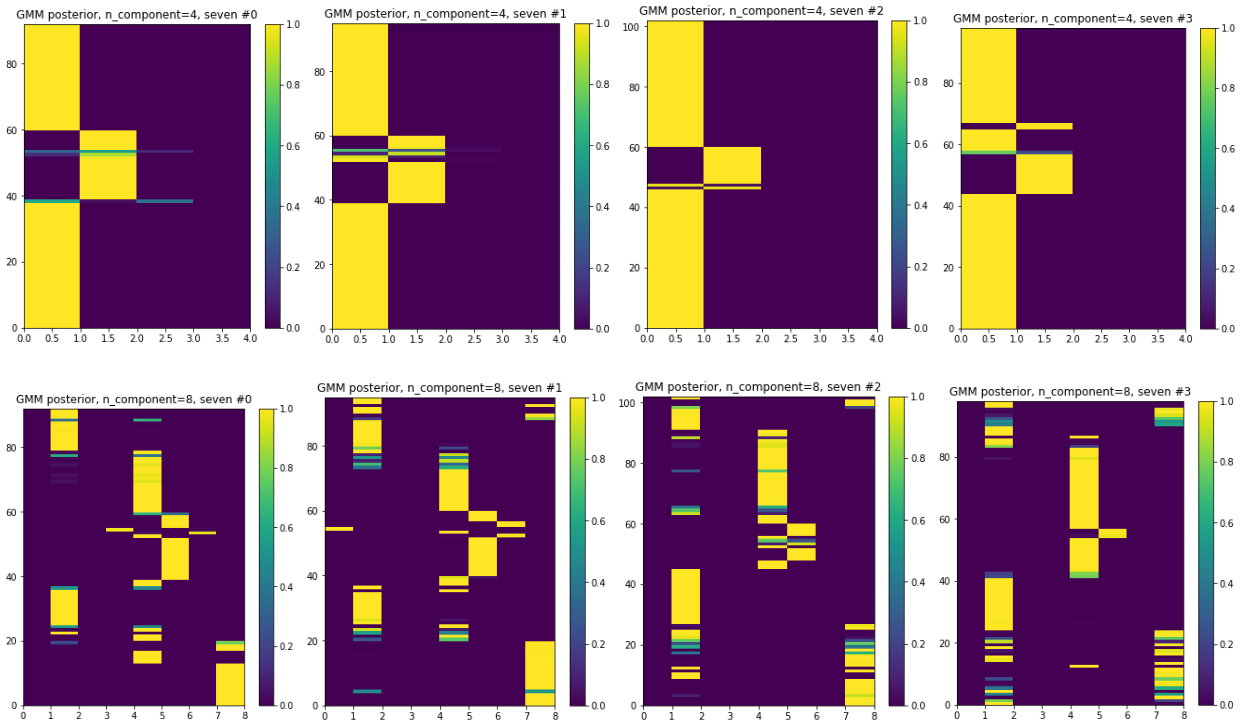


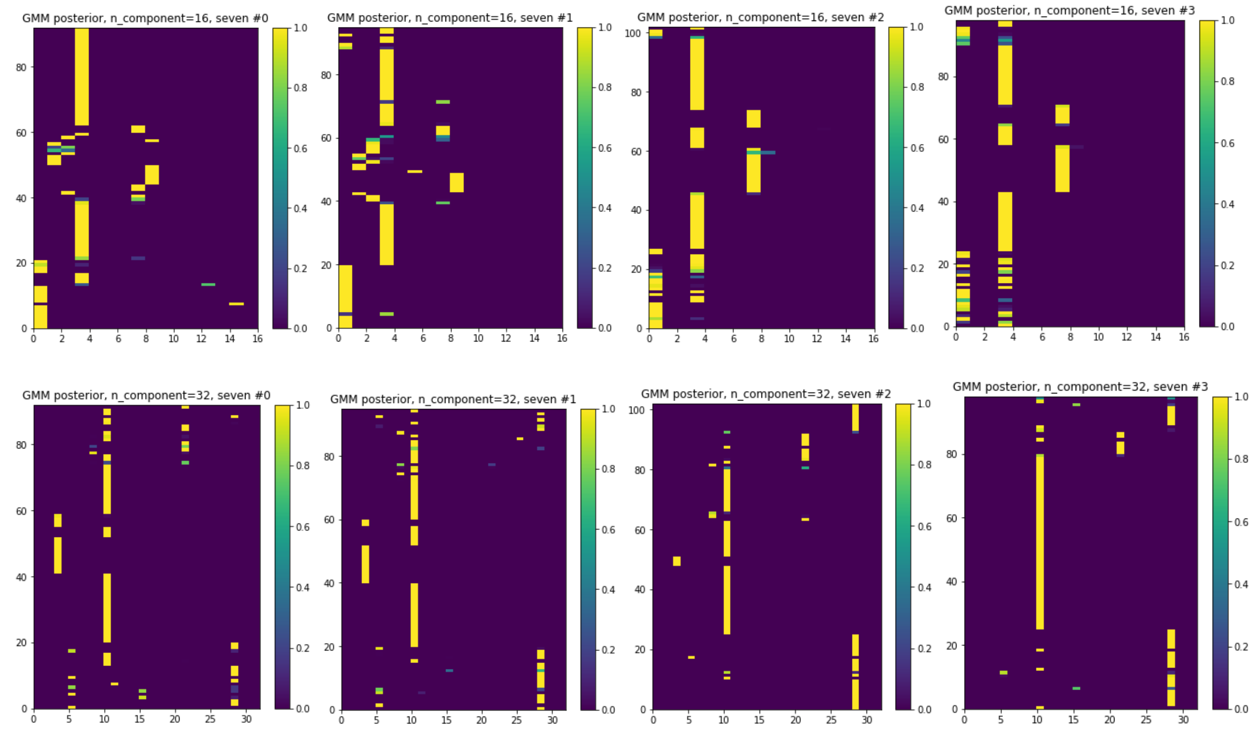
**5. Feature Correlations**



According to the Mfcc correlations matrix, some features are correlated like C0 and C1. But in general, the correlations among features are small which means the diagonal covariance matrices for Gaussian modelling can be approximated. The correlations matrix for Mel filterbank features have shown strong correlations among features.

**6. Explore Speech Segments with Clustering**

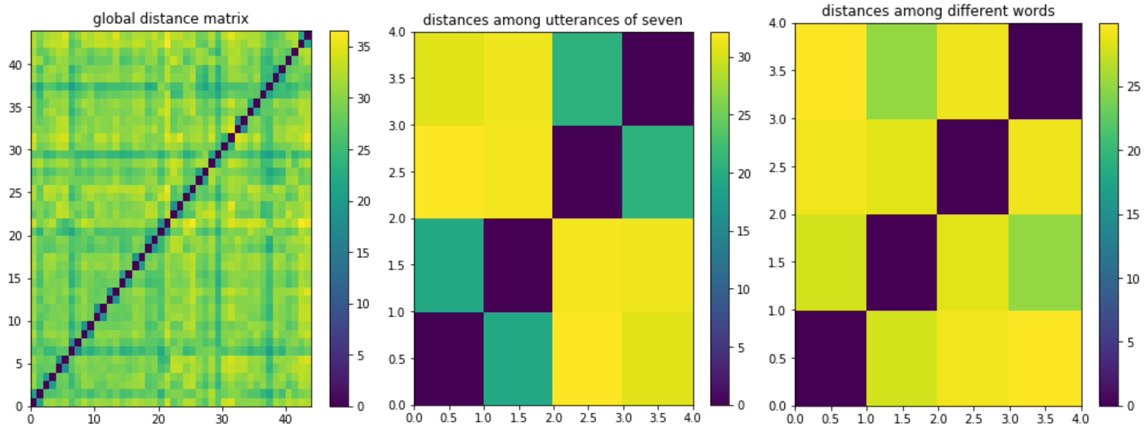




GMM posterior, n_component=16, seven #0
GMM posterior, n_component=16, seven #1
GMM posterior, n_component=16, seven #2
GMM posterior, n_component=16, seven #3
GMM posterior, n_component=32, seven #0
GMM posterior, n_component=32, seven #1
GMM posterior, n_component=32, seven #2
GMM posterior, n_component=32, seven #3

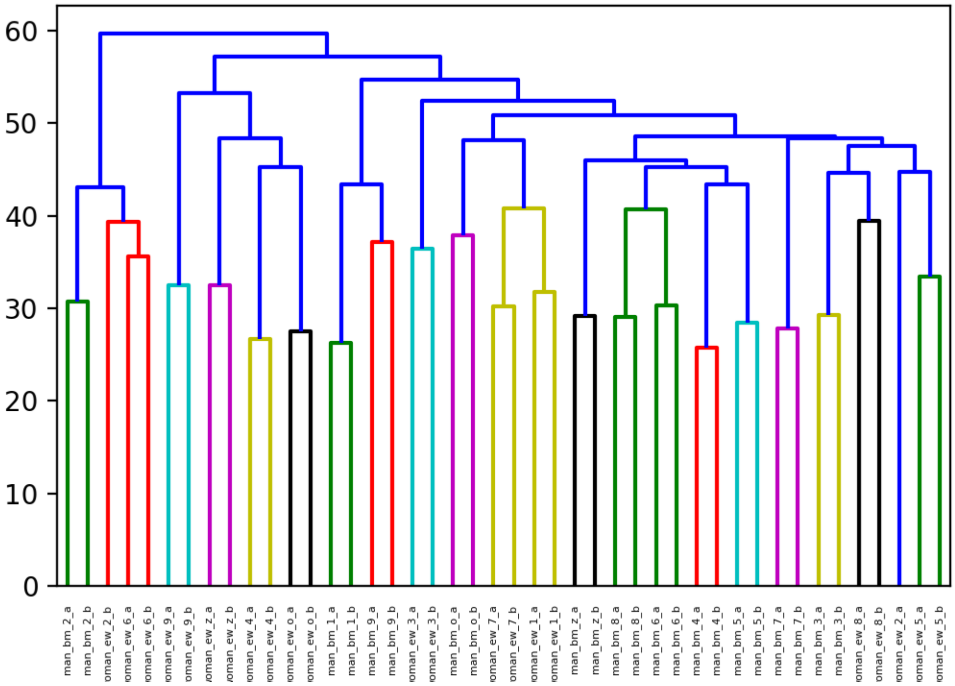
From the experiment results, with different number of components of GMM models, the posterior looks very similar when testing the GMM models with different utterance of the same word. The patterns of the posterior could indicate the class of the word. Therefore, I think the GMM models have learnt stable representations for the words.

**7. Comparing Utterances**



global distance matrix
distances among utterances of seven
distances among different words

The distance among different words are large in general. And the distances between the same word spoken by the same person are small. However, the distances between two same words spoken by different speakers are also large. Therefore, only consider the distances can't separate the digits well enough.



From the results above, the same word spoken by the same person are classified into the same leaf cluster in general. And words with similar pronunciations are also very closed to each other like ‘seven’ and ‘one’ are clustered into close clusters. Clusters that have higher depths seems to show less patterns. For example, ‘four’ and ‘five’ spoken by man are in close clusters but I don’t think they have similar pronunciations.