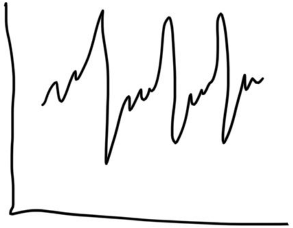
Unsupervised Learning with Speech

M.Tech. Artificial Intelligence, Second Year, NMIMS

By,

Bilal Hungund, Data Scientist, Halliburton

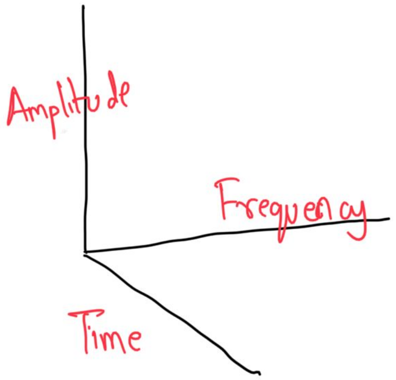
Sound Conversion:



$\longrightarrow$ Amplitude

Frequency

Time



$\Downarrow$

Helps for applying various transformation

$\rightarrow$ Frequency are key characteristics which composed sounds

$\rightarrow$ Chunk of information are encoded in the generation of frequency and their amplitude over time

Time and Frequency = Spectrogram Representation

Time Normalization: Template Matching

$\longrightarrow$ Deterministic Sequence Recognition

Mean Sequence of 12 MFCCs for two words

$$x = [x_1, x_2, x_3 \ldots\ldots x_{12}]$$

$$y = [y_1, y_2, y_3, \ldots\ldots y_{12}]$$

$$D(x, y) = \sqrt{\sum_i (x_i - y_i)^2} \Longrightarrow \text{Euclidean Distance}$$

→ When the words are uttering, the length of the utterance will change due to natural variations.

→ This makes the sequences have different lengths.

→ In order to handle time variation, we need two time normalization technique

→ Linear Time Warping

→ Dynamic Time Warping

Linear Time Warping

→ Linear Function used for twisting two template

→ Words such as

S p e e c h can be uttered as

S s p e e h h

As this is a linear function LTW will not help because the length of speech utterence will not be same.

→ Thus dynamic warping is required

Dynamic Time Warping

→ Helps to compare words with different sequences

→ The goal is to optimize the dynamic function and minimize the global distance error

Optimization Conditions

- Monotonic Condition
- Continuity Condition
- Boundary Condition
- Adjustment window condition
- Slope Constraint condition

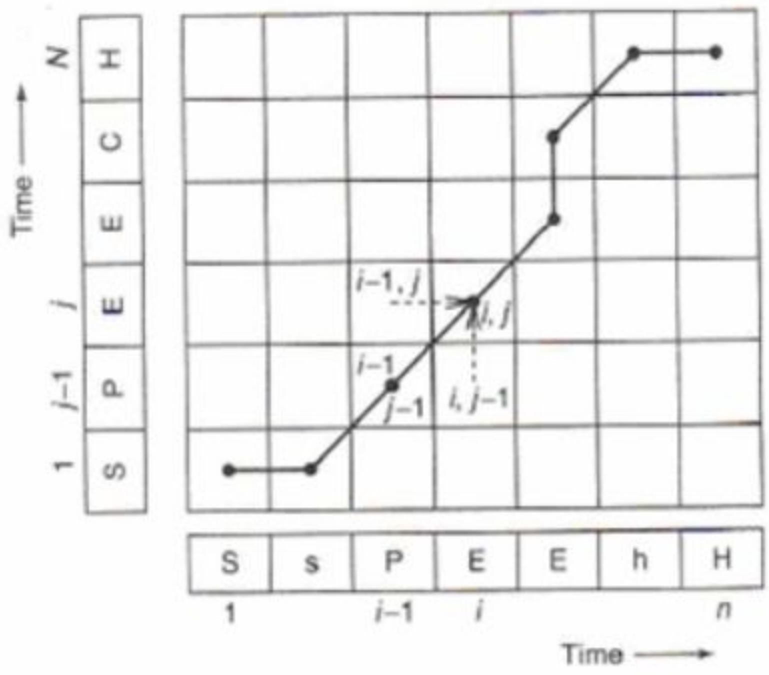
Algorithm for DTW:

→ Start with the first point in each template, compute the distance (this will be the cost function)

→ Move on to the next point in a shorter template and find its cost function with the next point in the longer template

→ Go on traversing in the forward direction for each point in the shorter template till both end points meet

$$D(i,j) = \min\left(D(i+1,j+1),\ D(i+1,j),\ D(i,j+1)\right) + d(i,j)$$



Time
N
H
C
E
j
E
j–1
P
1
S
i–1, j
i, j
i–1
j–1
i, j–1
S
s
P
E
E
h
H
1
i–1
i
n
Time