2. Isolated word speech recognition & HHTS

Task: Recognition of isolated spoken words from a vocabulary

Challenges:
- variable speed
- speaker independence
- prosody etc.

How do we (humans) hear?
- audio signal $\rightarrow$ tympanic membrane $\rightarrow$ ossicles $\rightarrow$
- cochlea: basilar membrane, inner & outer hair cells $\rightarrow$
- auditory cortex

A. Signal pre-processing
- Sample the pressure-time function $f(t)$ and digitize it
- highest frequency in speech signal $\leq 10$ KHz
- $\rightarrow$ Nyquist-Shannon theorem $\rightarrow$ Sample with 20 KHz
- Apply digital Fourier transform with sliding window

$$C(W,t) = \int_{-\infty}^{\infty} W(t-t') f(t') e^{iW't'} dt'$$

Simplest window function $W(t) = \begin{cases} 1 & \text{if } |t| < 6 \\ 0 & \text{otherwise} \end{cases}$

width 6: lowest freq. vs. time resolution

- Energy in spectra (logarithmic, dB)

$$S(W,t) = 20 \log_{10} ||C(W,t)||$$

- Discretize the $W$-domain into $\sim 20$ frequency channels
- Possibly cluster spectral vectors
  - pro: small number of feature vectors
  - con: dominance of stationary speech parts
L3. Dynamic time warping & word recognition

Model: a set of prototypes (i.e., feature sequences) per word (class)

Algorithm: nearest neighbour classifier

We need a distance measure for sequences of feature vectors

prototype \( x = (x_1, \ldots, x_n) \), signal \( y = (y_1, \ldots, y_m) \)

\( x_i, y_j \in \mathbb{R}^{26} \). Distance \( D(x, y) = ? \)

Monotonous matching (aka time warping)

matching \( \tau = (i_1, j_1), (i_2, j_2), \ldots, (i_k, j_k) \) \( \in \mathcal{T} \) if

1. \( (i_1, j_1) = (1, 1) \), \( (i_k, j_k) = (n, m) \)
2. \( i_{k-1} \leq i_k \leq i_{k+1} \)

Similarly for \( j \)-s

Distance for a fixed matching \( \tau \in \mathcal{T} \)

\[
D(x, y; \tau) = \sum_{k=1}^{l} \| x_{i_k} - y_{j_k} \|_2^2
\]

Distance between sequences \( x \) and \( y \)

\[
D(x, y) = \min_{\tau \in \mathcal{T}} D(x, y; \tau)
\]

How to compute it efficiently?

\[
\text{i.e. shortest path, here by dynamic programming with }
\text{time complexity } O(nm)
\]
Discussion: model & algorithm

- Inference has high time complexity $O(n^3p)$, where $p$ is the total number of prototypes.
- Learning: how to choose optimal prototypes?

Better: Model each word (class) by an HMM

$x = (x_1, x_n)$ - sequence of features (= spectral vectors)

$s = (s_1, s_n)$ - sequence of hidden states (e.g. phonemes)

$$P(x,s) = p(s_1) \prod_{i=2}^{n} p(s_i|s_{i-1}) \prod_{i=1}^{n} p(x_i|s_i)$$

where $p(x_i|s_i)$ could be e.g. Gaussians

- Fast inference (linear in $n$)
- Feasible learning of model parameters