2. Isolated word speech recognition & HMMs

Task: Recognition of isolated spoken words from a vocabulary

Challenges:

- variable speed
- speaker independence
- prosody etc.

How do we (humans) hear?

audio signal → tympanic membrane → ossicles → cochlea: basilar membrane, inner & outer hair cells → auditory cortex

2A. Signal pre-processing

- Sample the pressure-time function $f(t)$ and digitise it
  highest frequency in speech signal < 10 KHz
  → Nyquist-Shannon theorem → sample with 20 KHz
- Apply digital Fourier transform with sliding window

$$c(w,t) = \int_{-\infty}^{\infty} W(t-t')\, f(t')\, e^{iwt'}\, dt'$$

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)

$$S(w,t) = 20 \log_{10} \| c(w,t) \|$$

discretise the w-domain into ~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)

Alorithm nearest neighbour classifier

We need a distance measure for sequences of feature vectors

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

$x_i, y_j \in \mathbb{R}^{20}$. Distance $D(x,y)=?$

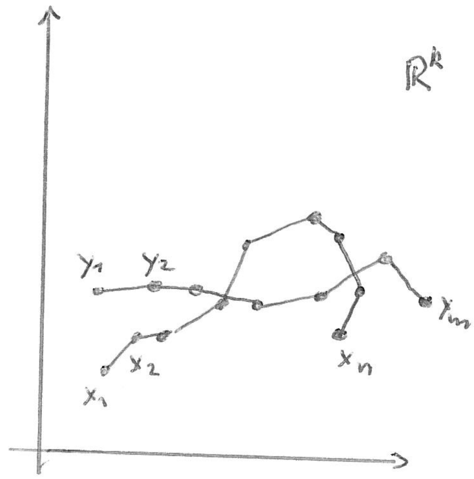
Monotonous matching (aka time warping)

matching $\tau=((i_1,j_1),(i_2,j_2),\dots,(i_\ell,j_\ell))\in T$ if

(1) $(i_1,j_1)=(1,1)$, $(i_\ell,j_\ell)=(n,m)$

(2) $i_{k-1}\le i_k\le i_{k-1}+1$

Similarly for j-s



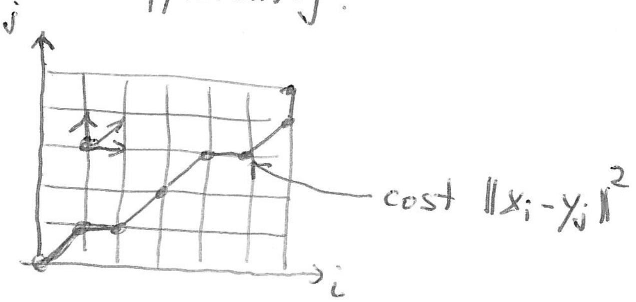
Distance for a fixed matching $\tau\in T$

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

Distance between sequences x and y

$$D(x,y)=\min_{\tau\in T} D(x,y;\tau)$$

How to compute it efficiently?



i.e. shortest path, here by dynamic programming with run-time complexity $\mathcal{O}(nm)$

Discussion model & algorithm

- Inference has high time complexity $\mathcal{O}(n^2 p)$, where p is the total number of prototypes
- Learning: how to choose optimal prototypes?

Better: Model each word (class) by an HMM

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

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

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

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

- fast inference (linear in n)
- feasible learning of model parameters