Speech Recognition

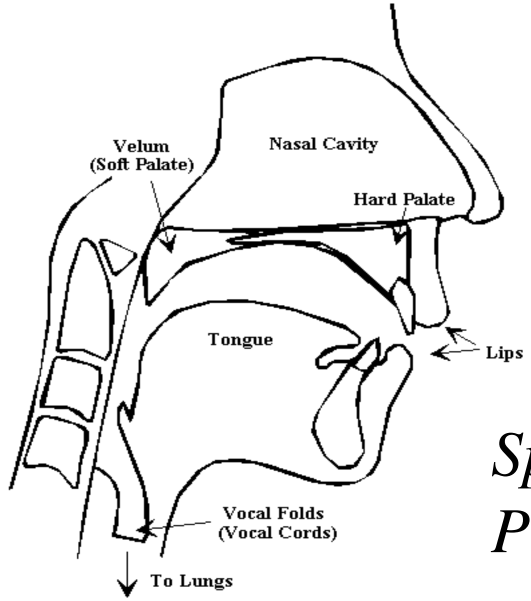
UNIVERSITY OF WASHINGTON
ELECTRICAL ENGINEERING

Speech Recognition

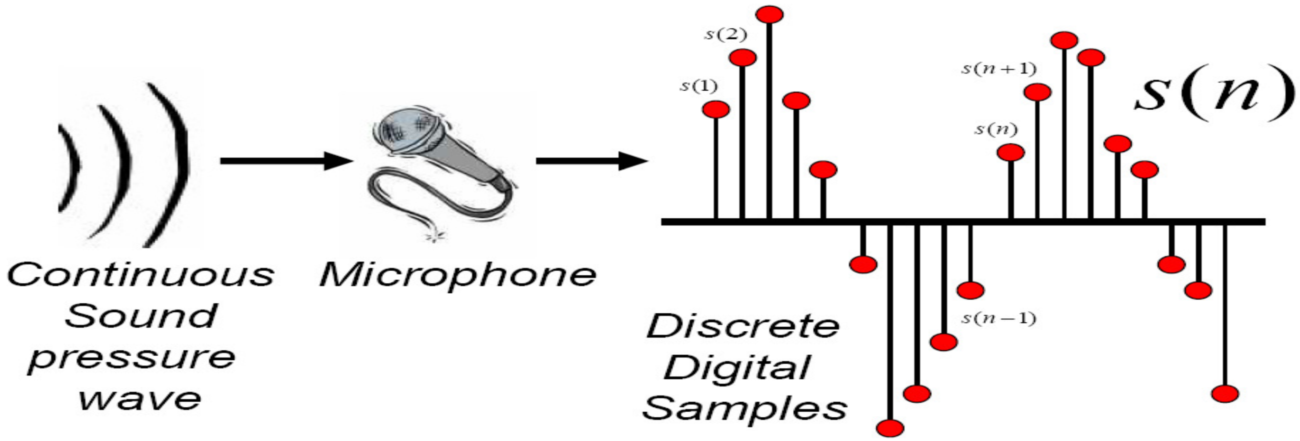
- Fundamentals of Digital Speech Processing
- Mel-Frequency Cepstral Coefficients (MFCCs)
- Speech Recognition by Dynamic Time Warping
- Speech Recognition by Hidden Markov Models

Human Speech Production

- All speech sounds are formed by blowing air from the lungs through the vocal tract.



Speech Production

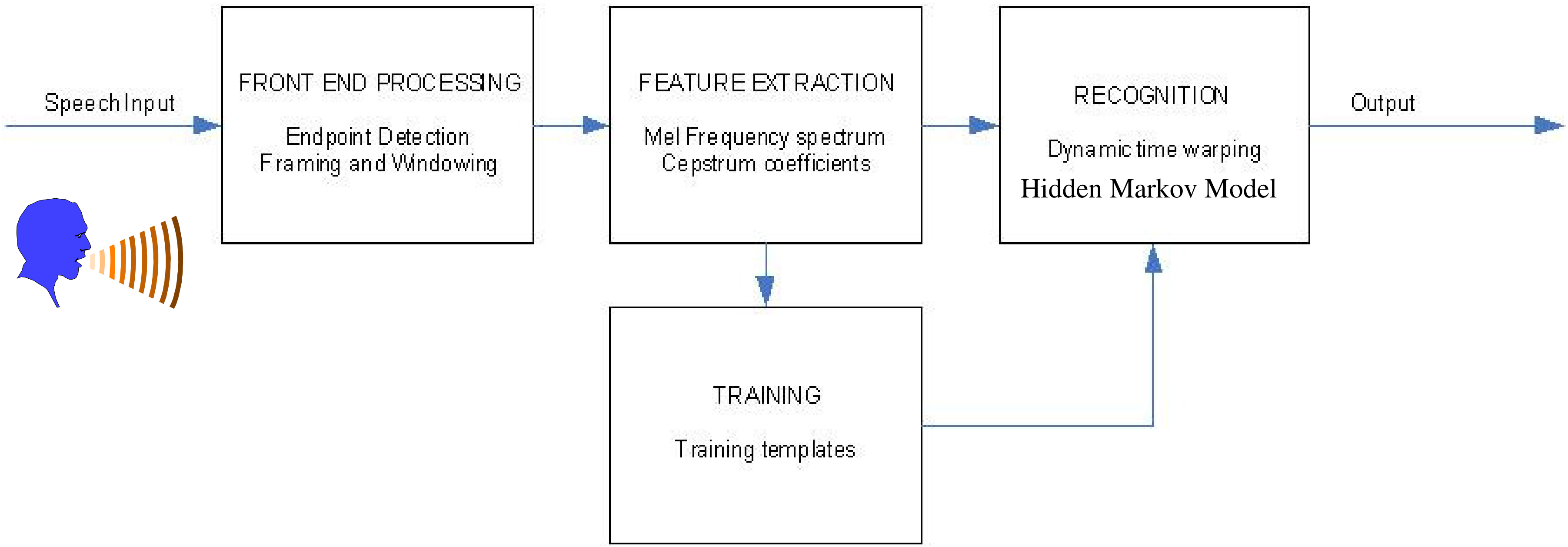


8KHz or 16KHz

Automatic Speech Recognition (ASR) Paradigms

- Continuous vs. Isolated
- Large (>1000) vs. Small Vocabularies (< 100)
- Speaker Dependent vs. Speaker Independent
- Speech Recognition vs. Speaker Recognition
- Speaker Recognition vs. Speaker Verification
- Context Dependent vs. Context Independent Verification
- Key Word Spotting
- SubWord Speech Units and Modeling
- Statistical Language Modeling and Perplexity
- Robust Speech Recognition and Adaptation

A Typical ASR System

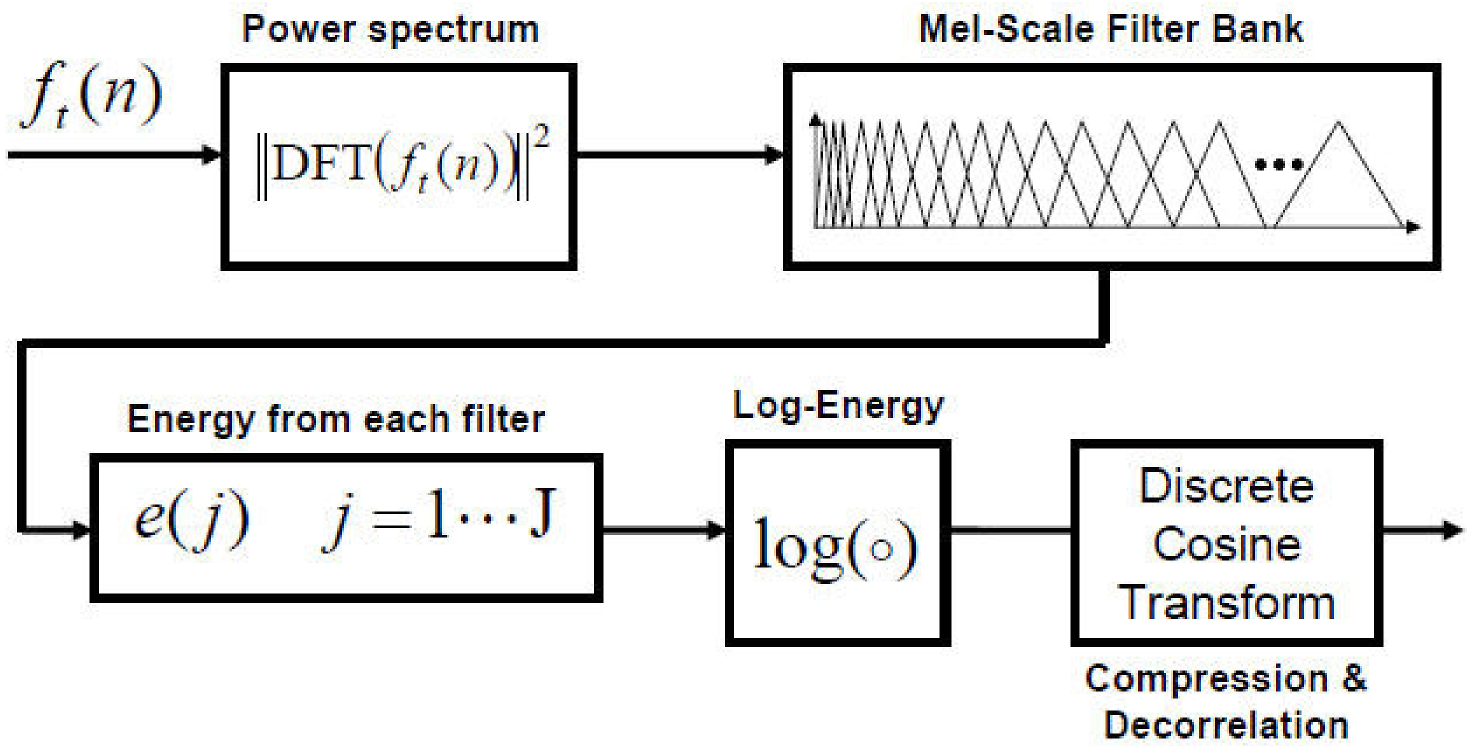


Why ASR is Difficult?

- Speech signals are **continuous.**
  - No explicit markers to indicate end of one sound and start of the next.
- Speech signals are **highly variable**.
  - Not only differences as a result of different people/sex saying the same word/sentence, but also differences with the **same** person saying the **same** word/sentence at different times.
- Speech is **ambiguous**.
  - There is no acoustic difference between **to**, **two** and **too**.
- Speech is **contaminated.**
  - -- Usually a speech signal occurs in an environment where there is some degree of reverberation, or competing acoustic noises

MFCC (see speaker_recognition project)

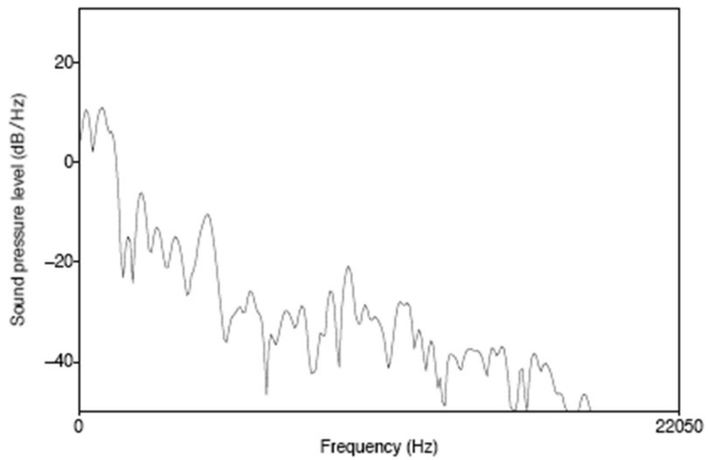
- Mel-Frequency Cepstral Coefficient (MFCC)
  - Most widely used spectral representation in ASR

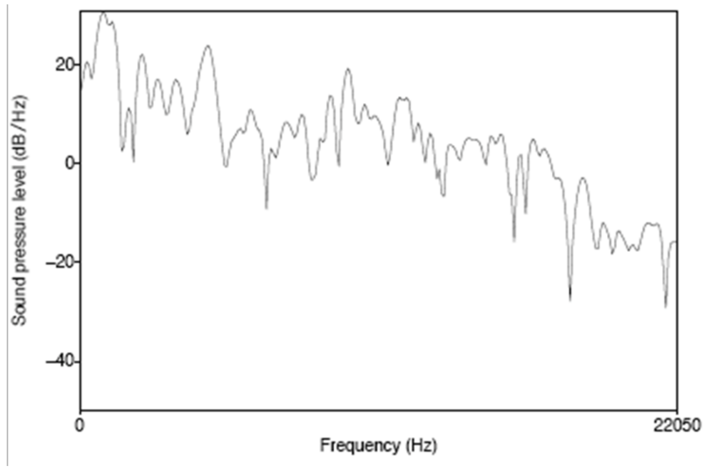


Pre-Emphasis before DFT

- Pre-emphasis: remove the lowpass lip radiation effect and boost the energy in the high frequencies
- Boosting high-frequency energy gives more info to Acoustic Model – better recognition

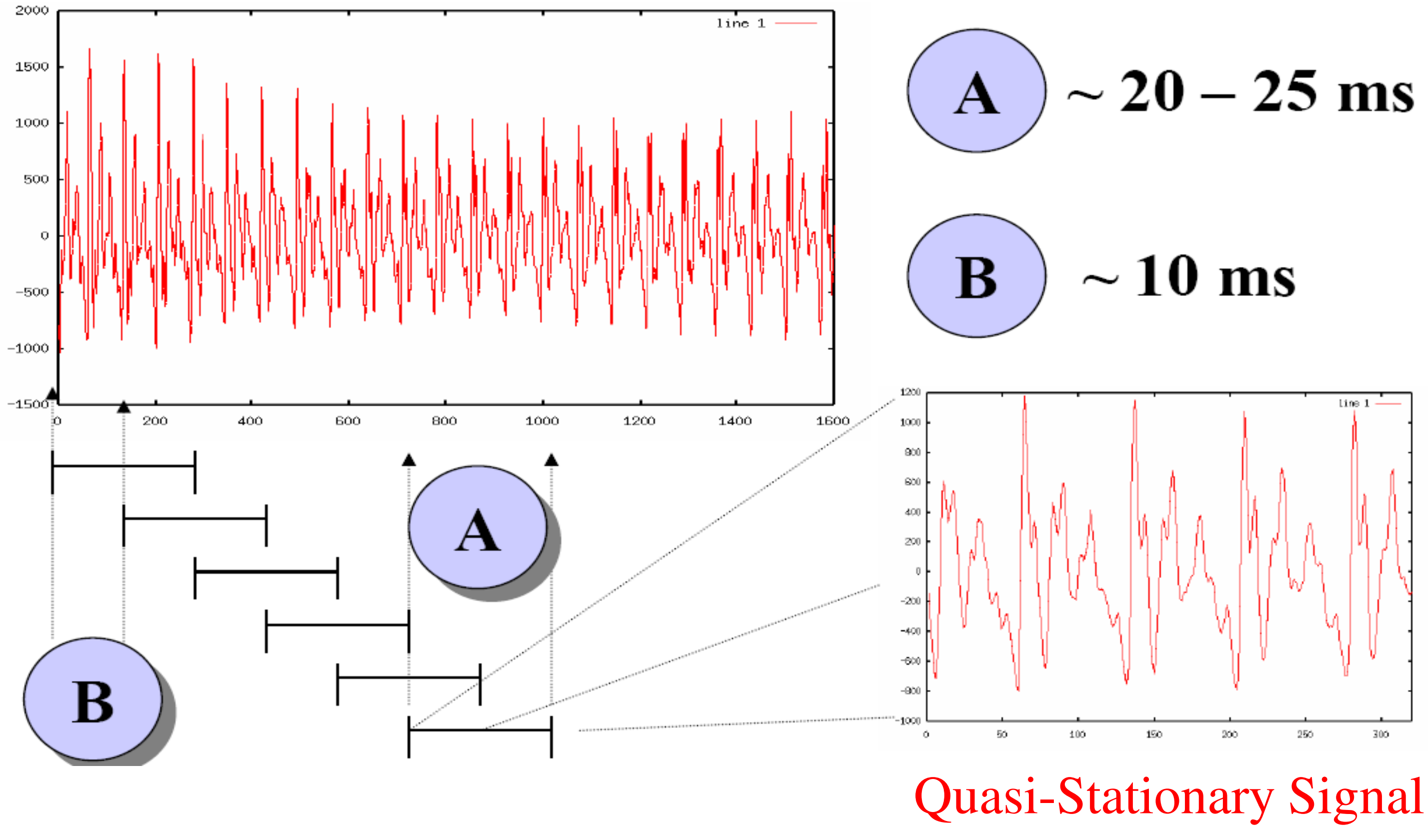
$$1 - \alpha z^{-1}, \quad \alpha = 0.97$$



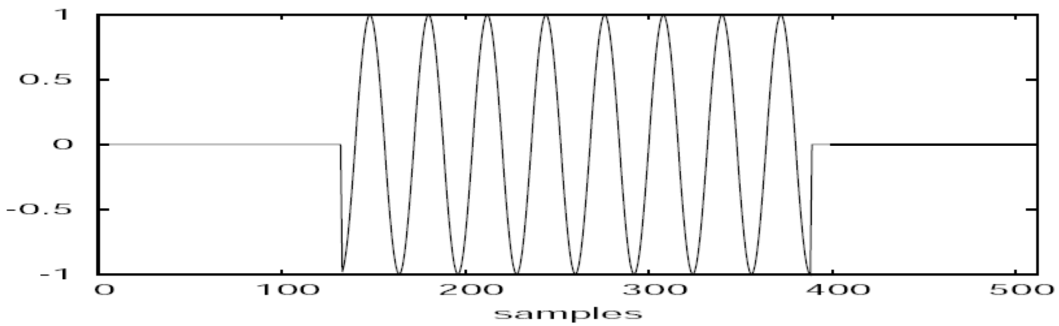


Windowing (Framing)

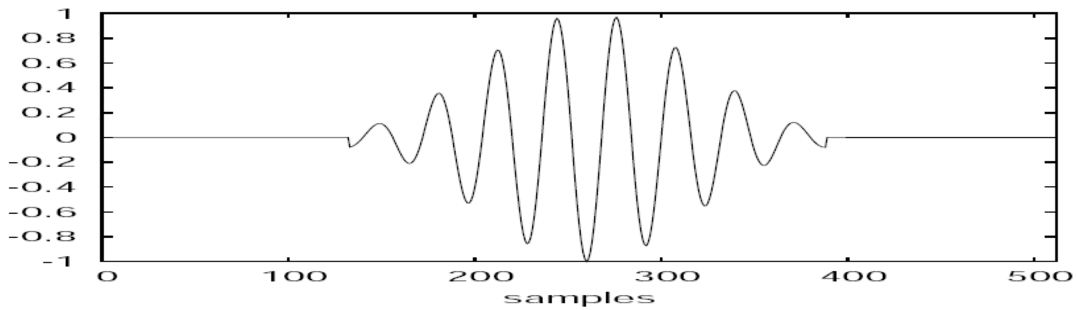
- Apply Hamming window, duration 200 samples (25 msec) every 10 ms (100-Hz frame rate)



Hamming Windowed Frames



(a) Rectangular window

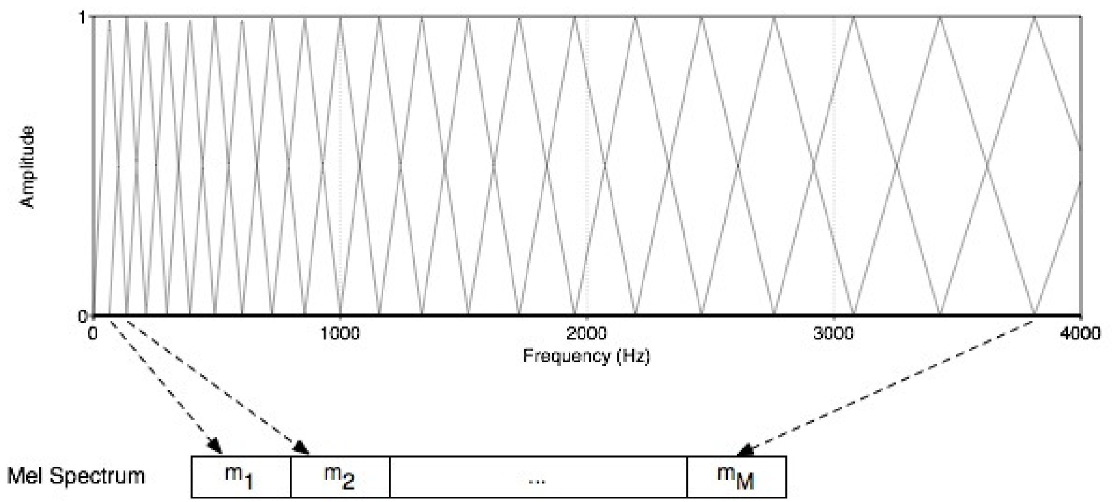


(c) Hamming window

Mel-Scale & Mel Filter Bank

- Human hearing is not equally sensitive to all frequency bands, less sensitive at higher frequencies
- Human perception of frequency is non-linear
- Mel-scale is approximately linear below 1 KHz and logarithmic above 1 KHz

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



Mel-filter Bank Processing

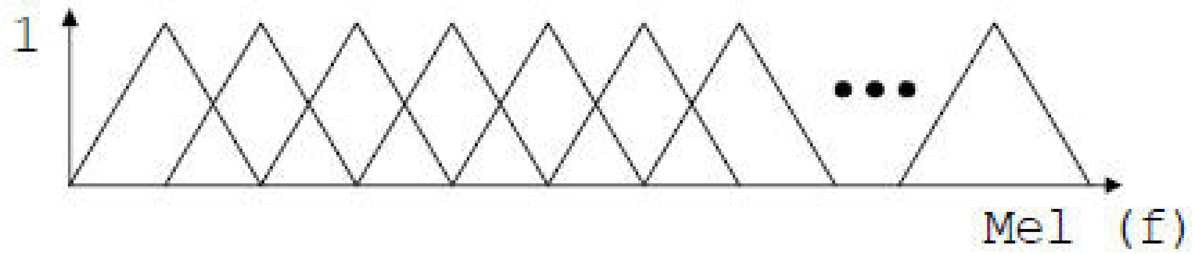
- (20-24) triangular shaped filters spaced evenly along the Mel Frequency Scale with 50% overlap
- Energy from each filter is computed (N = DFT size, P = #filters) at time t:

$$e[j][t] = \sum_{k=0}^{N-1} H_j[k] \cdot \left\| \widetilde{S}_t[k] \right\|^2 \quad \text{for } j = 1 \ldots P$$

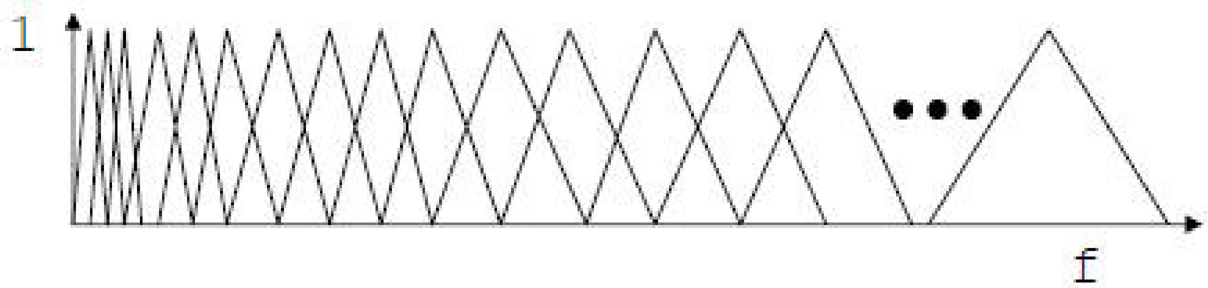
Triangular Filter *Signal Power Spectrum*

Equally Spaced Filters

- Equally spaced filters along the Mel-frequency scale with 50% overlap



- Analogous to non-uniformly spaced filters along linear frequency scale:



DCT to Approximate IDFT

- Compute Log-Energies from each of P filters
- Apply Discrete Cosine Transform (DCT)

$$MFCC[i][t] = \sqrt{\frac{2}{P}} \sum_{j=1}^{P} \left\{ (\log e[j][t]) \cdot \cos\left(\frac{\pi i}{P} (j - 0.5) \right) \right\}$$

- DCT: (1) improves diagonal covariance assumption, (2) compresses features
- Typically 12-14 MFCC features are extracted (higher order MFCCs useful for speaker-ID)

Distance Measure in the Pattern Matching

- Given one frame of speech (testing):

$$\vec{O} = (O_1, O_2, O_3, \ldots, O_{12}),$$

- and another frame of speech (template):

$$\vec{E} = (E_1, E_2, E_3, \ldots, E_{12}).$$

- the distance between these two frames of speech is (can be weighted if not cepstral coeff.):

$$d = dist(\vec{O}, \vec{E}) = \sum_{m=1}^{12} (O_m - E_m)^2$$

- Note that time domain comparison is meaningless, especially human ears are insensitive to time delay and slight vocal tract variations, which result in big changes in time domain.

Endpoint Detection

- To determine the **beginning** and the **end** of an isolated utterance (isolated word, isolated sentence, etc).
- Two factors: **energy** (sum of the magnitude in each frame) and the zero crossing rate (**ZCR**).
- For either "energy" or "ZCR", there is an associated "possible threshold" P_T , and "word start threshold" WS_T .
- The P_T is "set" when the energy is just above background noise level (may be exceeded by spurious noise). The WS_T is "set" when the system is sure a word is spoken.
- Two additional thresholds: minimum word length threshold, ML_T (i.e., minimum number of frames per word), and minimum silence duration threshold MD_T , (i.e., minimum number of frames after the end of a word).

Start of A Word

- The condition: $frame-energy > WS_T^{(E)}$, or ZCR $> WS_T^{(Z)}$
- Once $P_T^{(E)}$ or $P_T^{(Z)}$ is exceeded, we then have to continue search forward until the corresponding WS_T is exceeded. If before WS_T is exceeded, P_T fails itself, then we have to restart the search.
- Once a valid word is detected, we start the search of the long-enough silence end MD_T. Once the end is identified, the word length must exceed the ML_T to be qualified as a word.

Pattern Matching

1. Dynamic Time Warping (DTW)

2. Hidden Markov Modelling (HMM)

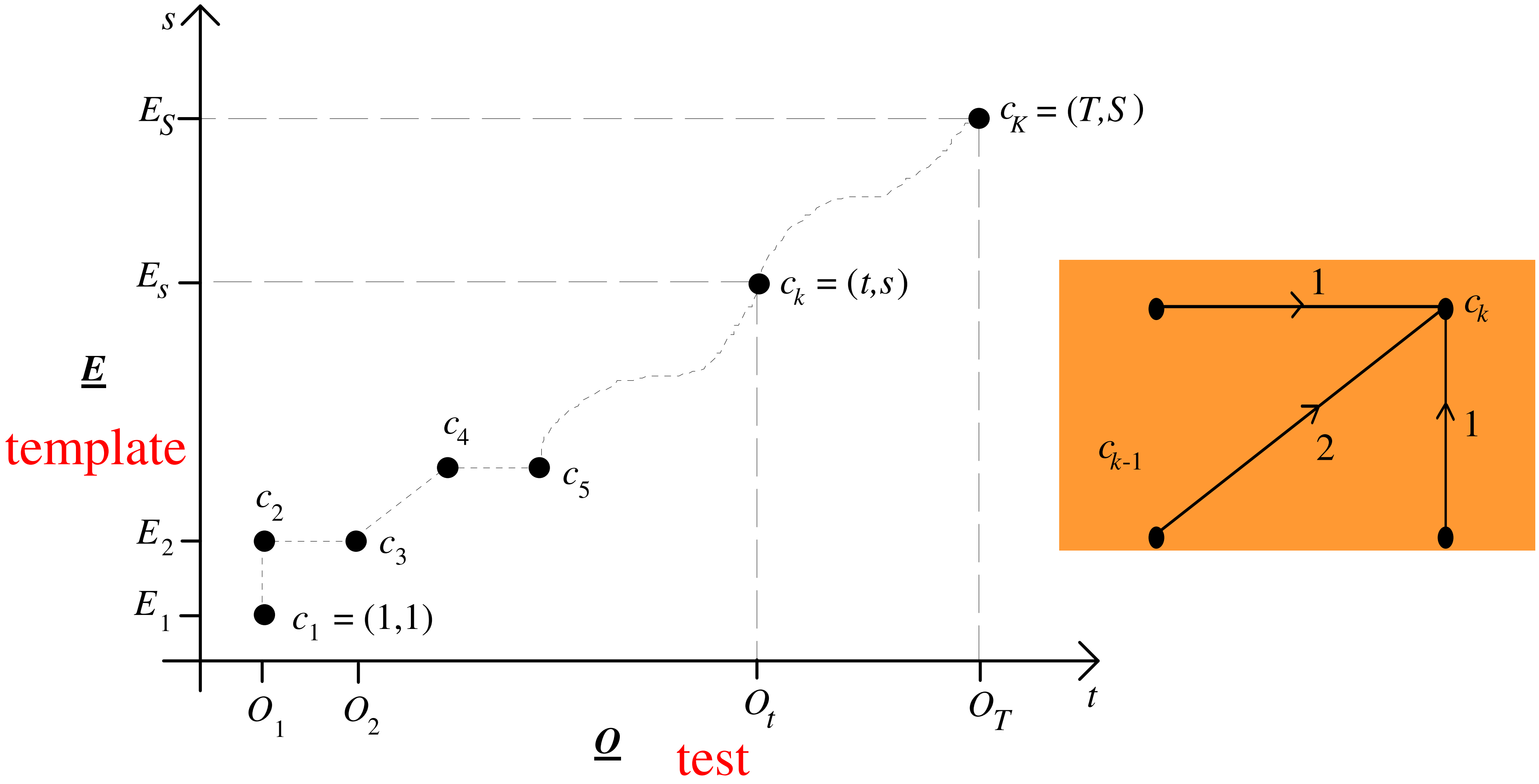
3. Others

DTW Basics

1	4	3	1	2	4
3	2	1	5	7	5
3	1	3	5	5	6
3	5	8	2	4	5
1	4	3	6	7	8
2	1	7	5	8	6

Template MFCC

Test MFCC



The weighted sum of distances along the function $\underline{c}$ is

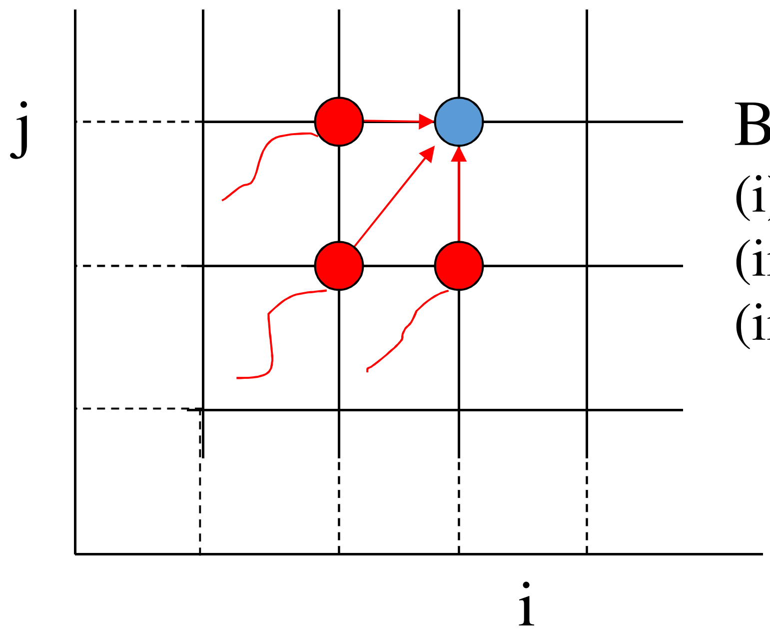
$$\sum_{k=1}^{K} d(c_k)w_k \quad \text{where} \quad d(c_k) = d(O_{t(k)}, E_{s(k)})$$

It remains to find the warping function which minimizes

$$\sum_k d(c_k)w_k$$

A direct search is too slow.

We use the method known as Dynamic Programming.



Best path to (i,j) is the minimum of

(i) best path to (i-1,j) +dist from (i-1,j) to (i,j)

(ii) best path to (i-1,j-1) +dist from (i-1,j-1) to (i,j)

(iii) best path to (i,j-1) +dist from (i,j-1) to (i,j)

Sakoe and Chiba algorithm

Dynamic programming solution to speech pattern matching problem:

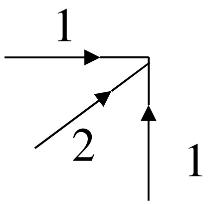
$$f(1,1) = 2d(1,1) \quad (\text{assuming } c_0 = (0,0) \text{ so } w_1 = 2)$$

$$f(t,1) = f(t-1,1) + d(t,1) \quad (\text{for } 2 \le t \le T)$$

$$f(1,s) = f(1,s-1) + d(1,s) \quad (\text{for } 2 \le s \le S)$$

$$f(t,s) = \min \begin{cases} f(t-1,s) + d(t,s) & (\text{right}) \\ f(t-1,s-1) + 2d(t,s) & (\text{diagonal}) \\ f(t,s-1) + d(t,s) & (\text{up}) \end{cases} \quad (\text{for } 2 \le t \le T,\ 2 \le s \le S)$$

$$D(\underline{\mathbf{O}}, \underline{\mathbf{E}}) = f(T,S)$$



f(t,s) is the shortest distance to the point (t,s)

d(t,s) is distance between feature vectors O_t and E_s

Example

The bold numbers represent distances between feature vectors for input data and template, i.e. $d(t,s)$, the numbers in parentheses are the cumulative distances $f(t,s)$, and the arrows indicate the best path.

	7 (16)	→	**8** (24)		**6** (24)		**4** (26)
	↑				↑	↗	
	4 (9)	→	**9** (18)		**5** (18)	→	**7**(25)
s	↑				↑		
	3 (5)	→	**7** (12)		**3** (13)	→	**7** (20)
	↑			↗			
	1 (2)	→	**5** (7)	→	**5** (12)	→	**9** (21)
				t			

Minimum overall distance is 26. The optimum path can be found by tracing the arrows back from the top right corner.

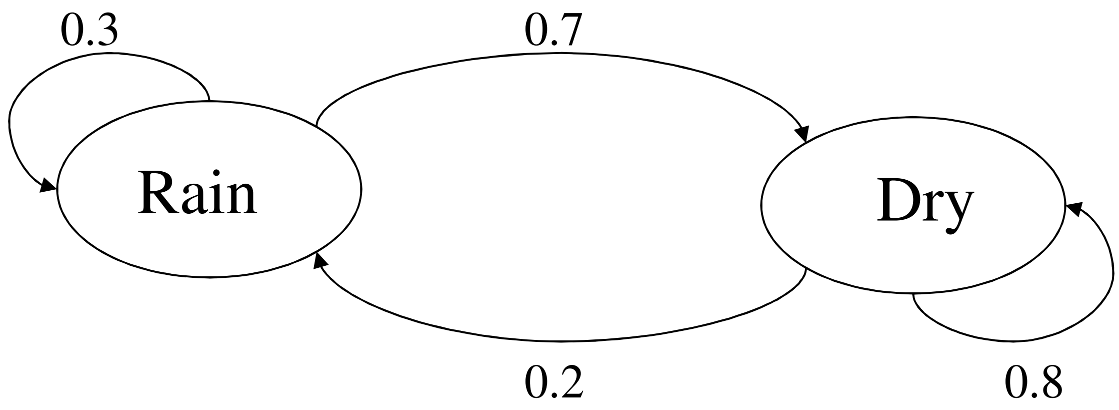
Markov Models

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states (observations) : $s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$
- Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

- To define Markov model, the following probabilities have to be specified: transition probabilities $a_{ij} = P(s_i \mid s_j)$ and initial probabilities $\pi_i = P(s_i)$

Example of Markov Model



- Two states : 'Rain' and 'Dry'.
- Transition probabilities: P('Rain'|'Rain')=0.3 , P('Dry'|'Rain')=0.7 , P('Rain'|'Dry')=0.2, P('Dry'|'Dry')=0.8
- Initial probabilities: say P('Rain')=0.4 , P('Dry')=0.6 .

Calculation of Sequence Probability

- By Markov chain property, probability of state sequence can be found by the formula:

$$P(s_{i1}, s_{i2}, \ldots, s_{ik}) = P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1})$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{i1}, s_{i2}, \ldots, s_{ik-1}) = \ldots$$

$$= P(s_{ik} \mid s_{ik-1}) P(s_{ik-1} \mid s_{ik-2}) \ldots P(s_{i2} \mid s_{i1}) P(s_{i1})$$

- Suppose we want to calculate a probability of a sequence of states (observations) in our example,

{'Dry','Dry','Rain', 'Rain'}.
P({'Dry','Dry','Rain', 'Rain'}) =
P('Dry') P('Dry'|'Dry') P('Rain'|'Dry')
P('Rain'|'Rain') = 0.6*0.8*0.2*0.3

Hidden Markov Model (HMM)

- Set of states: $\{s_1, s_2, \ldots, s_N\}$
- Process moves from one state to another generating a sequence of states :

$$s_{i1}, s_{i2}, \ldots, s_{ik}, \ldots$$

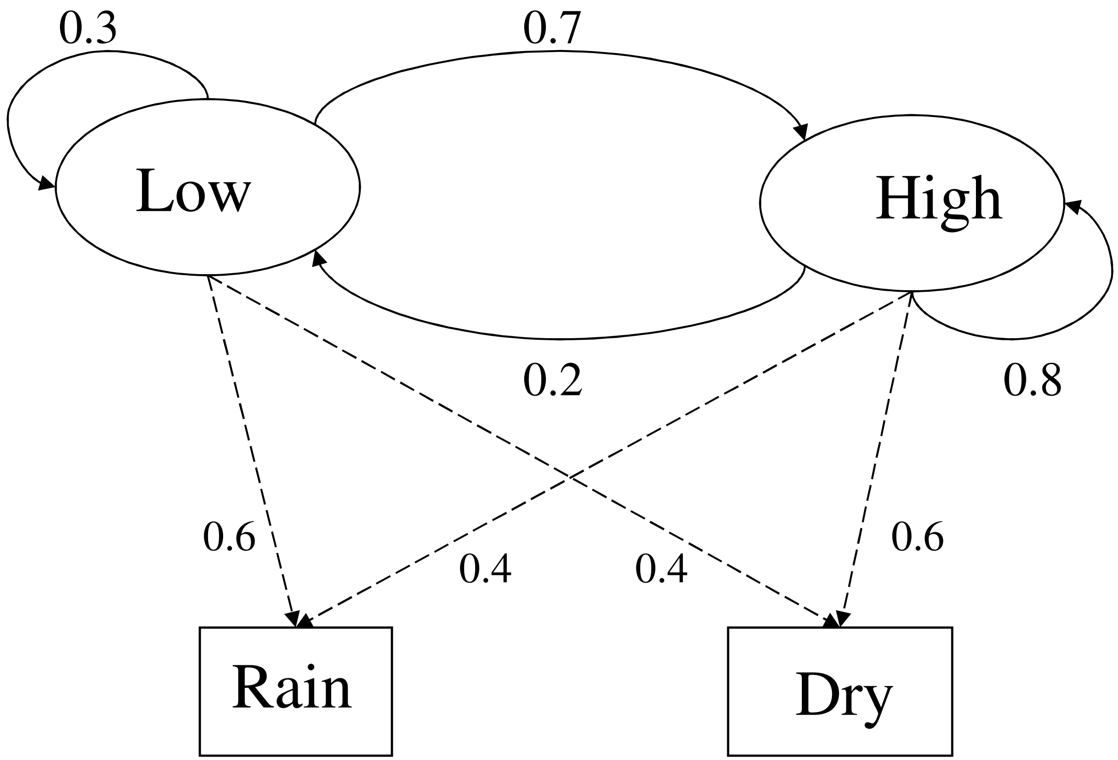
- Markov chain property: probability of each subsequent state depends only on what was the previous state:

$$P(s_{ik} \mid s_{i1}, s_{i2}, \ldots, s_{ik-1}) = P(s_{ik} \mid s_{ik-1})$$

- States are not visible, but each state randomly generates one of M observations (or visible states) $\{v_1, v_2, \ldots, v_M\}$

- To define hidden Markov model, the following probabilities have to be specified: matrix of transition probabilities $A=(a_{ij})$, $a_{ij}= P(s_i \mid s_j)$, matrix of observation probabilities $B=(b_i (v_m))$, $b_i(v_m) = P(v_m \mid s_i)$ and a vector of initial probabilities $\pi=(\pi_i)$, $\pi_i = P(s_i)$. Model is represented by $M=(A, B, \pi)$.

Example of An HMM



Example of An HMM

- Two states : 'Low' and 'High' atmospheric pressure.
- Two observations : 'Rain' and 'Dry'.

- Transition probabilities:
  P('Low'|'Low')=0.3 , P('High'|'Low')=0.7 ,
  P('Low'|'High')=0.2, P('High'|'High')=0.8

- Observation probabilities :
  P('Rain'|'Low')=0.6 , P('Dry'|'Low')=0.4 ,
  P('Rain'|'High')=0.4 , P('Dry'|'High')=0.3 .

- Initial probabilities: P('Low')=0.4 , P('High')=0.6 .

Calculation of Observation Sequence Probability

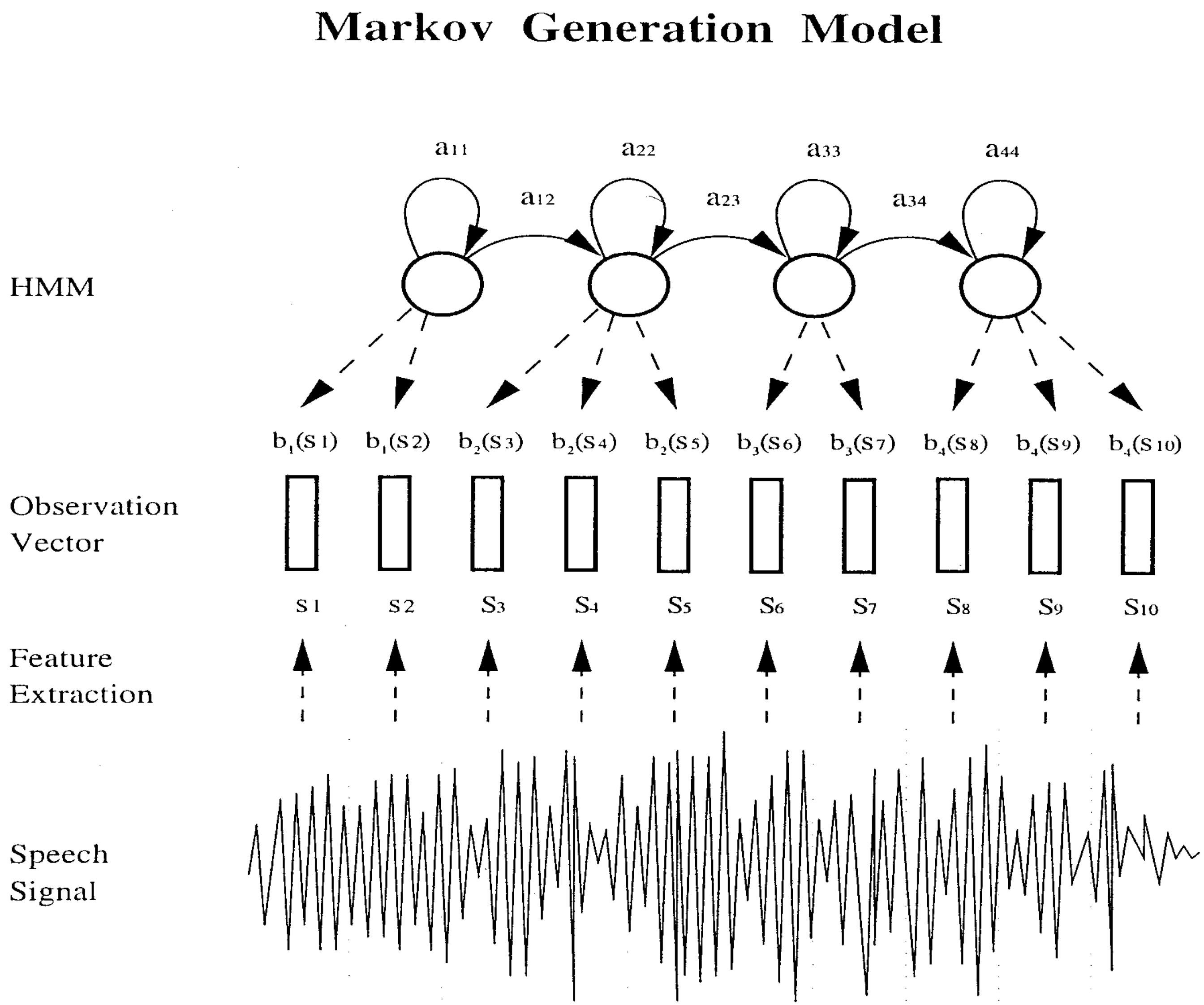
- Suppose we want to calculate a probability of a sequence of observations in our example, {'Dry','Rain'}.
- Consider all possible hidden state sequences:

  P({'Dry','Rain'}) = P({'Dry','Rain'} , {'Low','Low'}) + P({'Dry','Rain'} , {'Low','High'}) + P({'Dry','Rain'} , {'High','Low'}) + P({'Dry','Rain'} , {'High','High'})

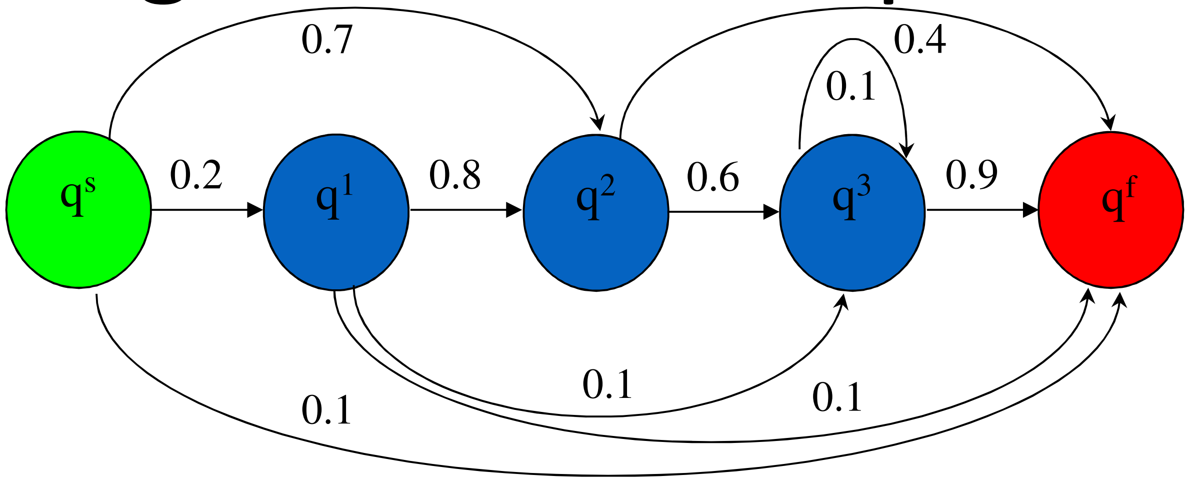
  where first term is :

  P({'Dry','Rain'} , {'Low','Low'})=
  P({'Dry','Rain'} | {'Low','Low'}) P({'Low','Low'}) =
  P('Low') P('Dry'|'Low') P('Low'|'Low) P('Rain'|'Low')
  = 0.4*0.4*0.3*0.6

Speech Generation by HMM



A Left-Right HMM Example



Three states + one starting state q^s + one final state q^f
q^s and q^f are non-emitting states (silence is modeled here).

Assume there are 2 symbols to observe $X = \{x^1=a, x^2=b\}$

$$\Pi = \begin{bmatrix} 0.2 \\ 0.7 \\ 0 \\ 0.1 \end{bmatrix} \qquad A = \begin{bmatrix} 0 & 0.8 & 0.1 & 0.1 \\ 0 & 0 & 0.6 & 0.4 \\ 0 & 0 & 0.1 & 0.9 \\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad B = \begin{bmatrix} 0.8 & 0.2 \\ 0.4 & 0.6 \\ 0.1 & 0.9 \end{bmatrix}$$

$P(a|q^1)$
$P(b|q^3)$

Initial state probabilities

Transition state probabilities

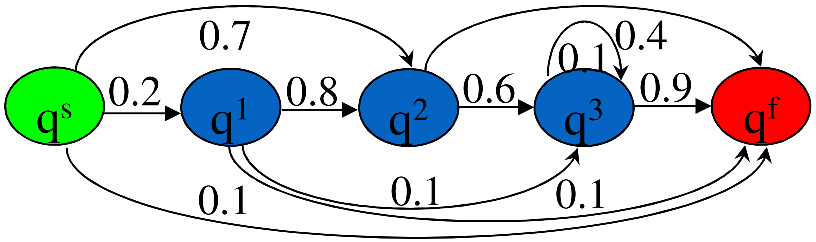
Observation symbol probabilities

Likelihood of An Observation Sequence

Given X = "aaa", compute the likelihood for this model : P(aaa | λ)

The likelihood P(X | λ) is given by the sum over all possible ways to generate X.

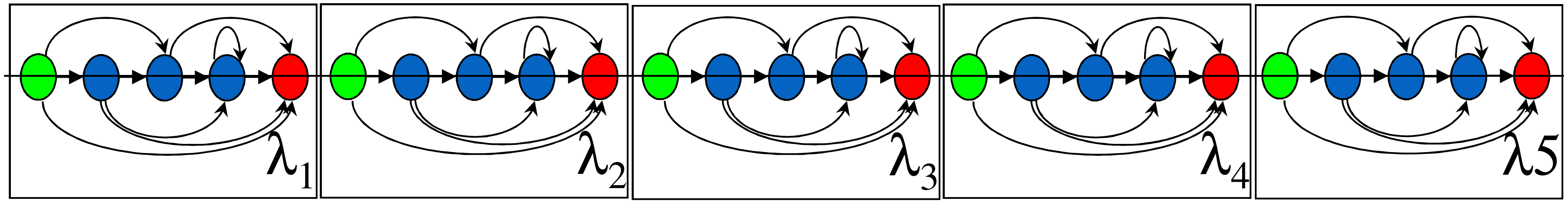
State ***sequence***	***Init***	***Obs a***	***Trans***	***Obs a***	***Trans***	***Obs a***	***Trans***	***Joint probability***
$q^1q^2q^3$	0.2	0.8	0.8	0.4	0.6	0.1	0.9	0.0027648
$q^1q^3q^3$	0.2	0.8	0.1	0.1	0.1	0.1	0.9	0.0000144
$q^2q^3q^3$	0.7	0.4	0.6	0.1	0.1	0.1	0.9	0.0001512
							P(aaa|λ) =	0.0029304



HMMs for Speech (Pattern) Recognition

Using HMM for speech recognition consists in computing the model li among a set of K models which maximizes the likelihood for an observation to have been generated by this model :

$$\lambda_{max} = \arg\max_{\lambda_i} P(X|\lambda_i) \qquad \text{for } i = 1, \ldots K$$



Three Problems for HMMs

- Problem 1 : Recognition

  Given X = (x1,x2, … xT) and the various models {li}

  How to efficiently compute P(X|li) ?

Forward-Backward algorithm

- Problem 2 : Analysis

  Given X = (x1,x2, … xT) and a model l, find the optimal state sequence G. How can we undiscovered the sequence of states corresponding to a given observation ?

Viterbi algorithm

- Problem 3 : Learning

  Given X = (x1,x2, … xT), estimate model parameters l = (P, A, B) that maximize P(X| l), i.e., How do we determine the model parameters l = (P, A, B) ?

Baum-Welch algorithm

References

- L. Rabiner, B.H. Juang, ***Fundamentals of Speech Recognition,*** Prentice Hall, 1993.
- L. Rabiner, B.H. Juang, "An introduction to hidden Markov models," IEEE Signal Processing Magazine, 3(1):4-16, Jan 1986.