Αναγνώριση Προτύπων & Αναγνώριση Φωνής

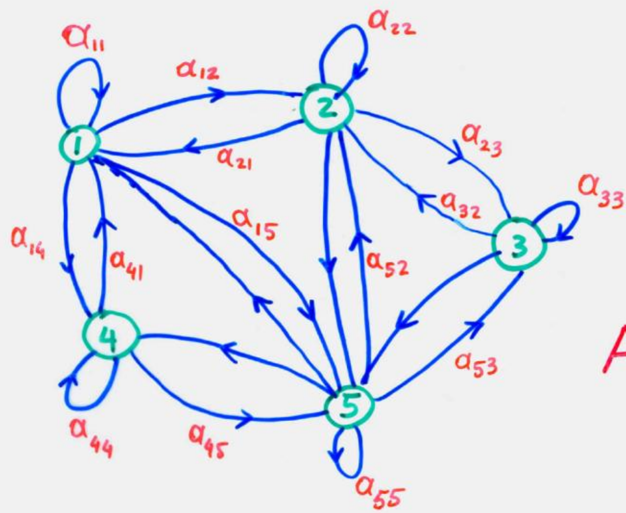
Πετρος Μαραγκος

HMM

DTW

http://cvsp.cs.ntua.gr/courses/patrec

HMM



$$a_{ij} \geq 0$$

$$\boxed{\sum_{j=1}^{N} a_{ij} = 1 \quad \forall i}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1N} \\ a_{21} & a_{22} & \cdots & a_{2N} \\ \vdots & & \ddots & \vdots \\ a_{N1} & a_{N2} & \cdots & a_{NN} \end{bmatrix}$$

$t = 1, 2, \dots, T$ discrete time

$O = O_1 O_2 \cdots O_T$ observation sequence

$T =$ length of " "

$N =$ # of states

$M =$ # of observation symbols

$Q = \{q_1, q_2, \dots, q_N\}$ states

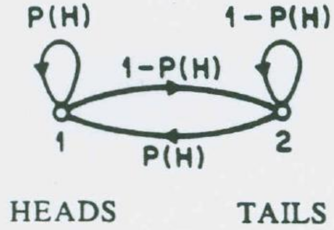
$V = \{v_1, v_2, \dots, v_M\}$ symbol observations

$A = \{a_{ij}\} = P_r(q_j \text{ at } t+1 / q_i \text{ at } t)$ state trans. probab.

$B = \{b_j(k)\}$, $b_j(k) = P_r(v_k \text{ at } t / q_j \text{ at } t)$ obs. symbol probab.

$\pi = \{\pi_i\}$, $\pi_i = P_r(q_i \text{ at } t=1)$ initial state probab.

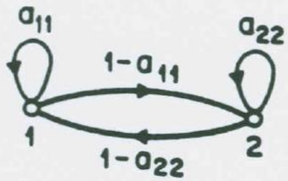
(a)



1-COIN MODEL
(OBSERVABLE MARKOV MODEL

O = $H\ H\ T\ T\ H\ T\ H\ H\ T\ T\ H$ ...
S = 1 1 2 2 1 2 1 1 2 2 1 ...

(b)



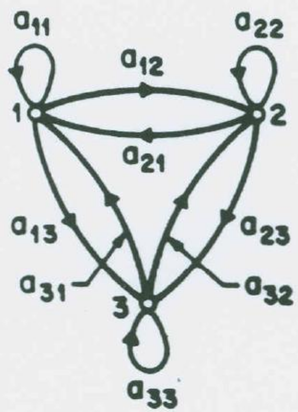
$P(H) = P_1$ $P(H) = P_2$

$P(T) = 1-P_1$ $P(T) = 1-P_2$

2-COINS MODEL
(HIDDEN MARKOV MODEL)

O = $H\ H\ T\ T\ H\ T\ H\ H\ T\ T\ H$ ...
S = 2 1 1 2 2 2 1 2 2 1 2 ...

(c)

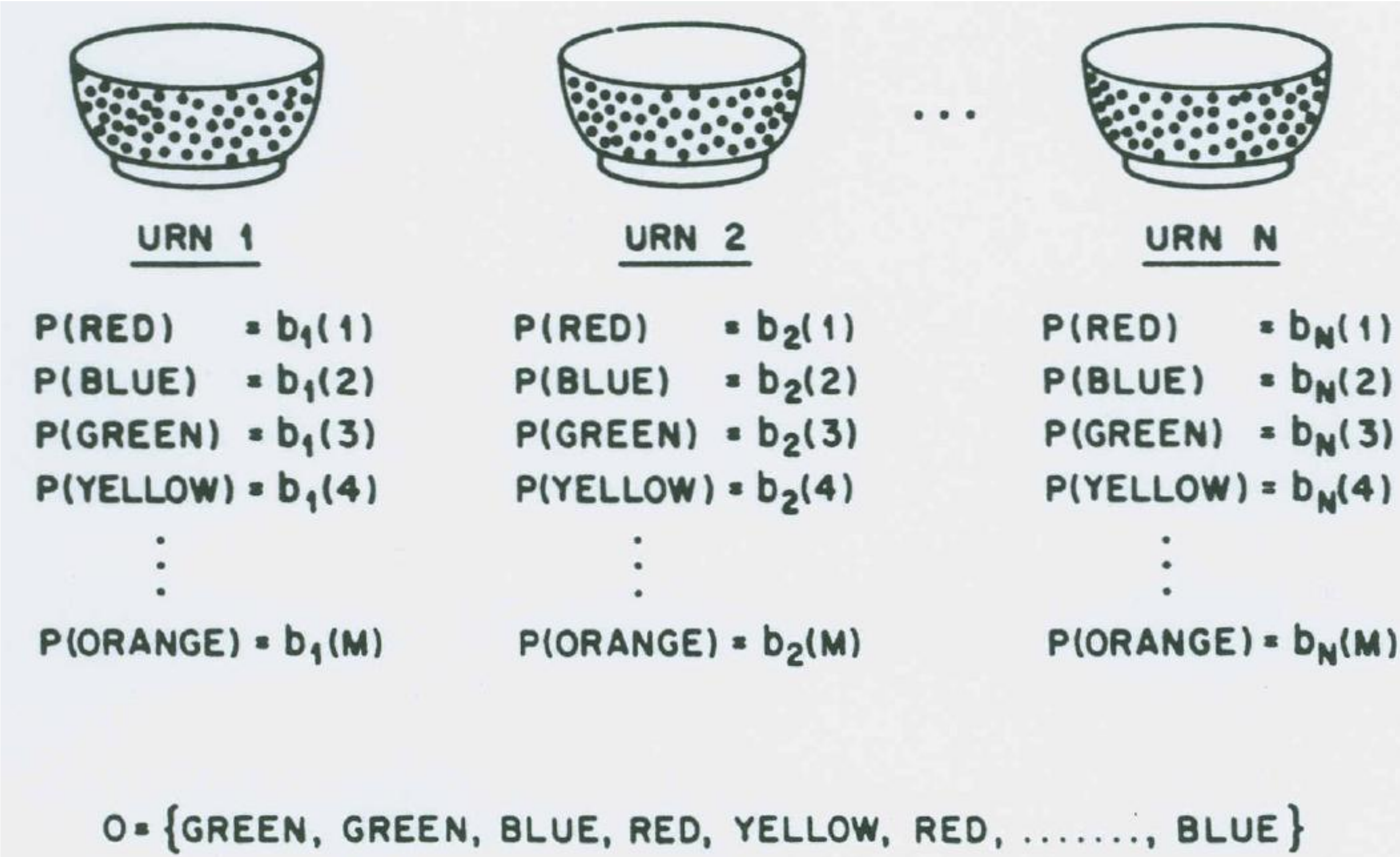


3-COINS MODEL
(HIDDEN MARKOV MODEL)

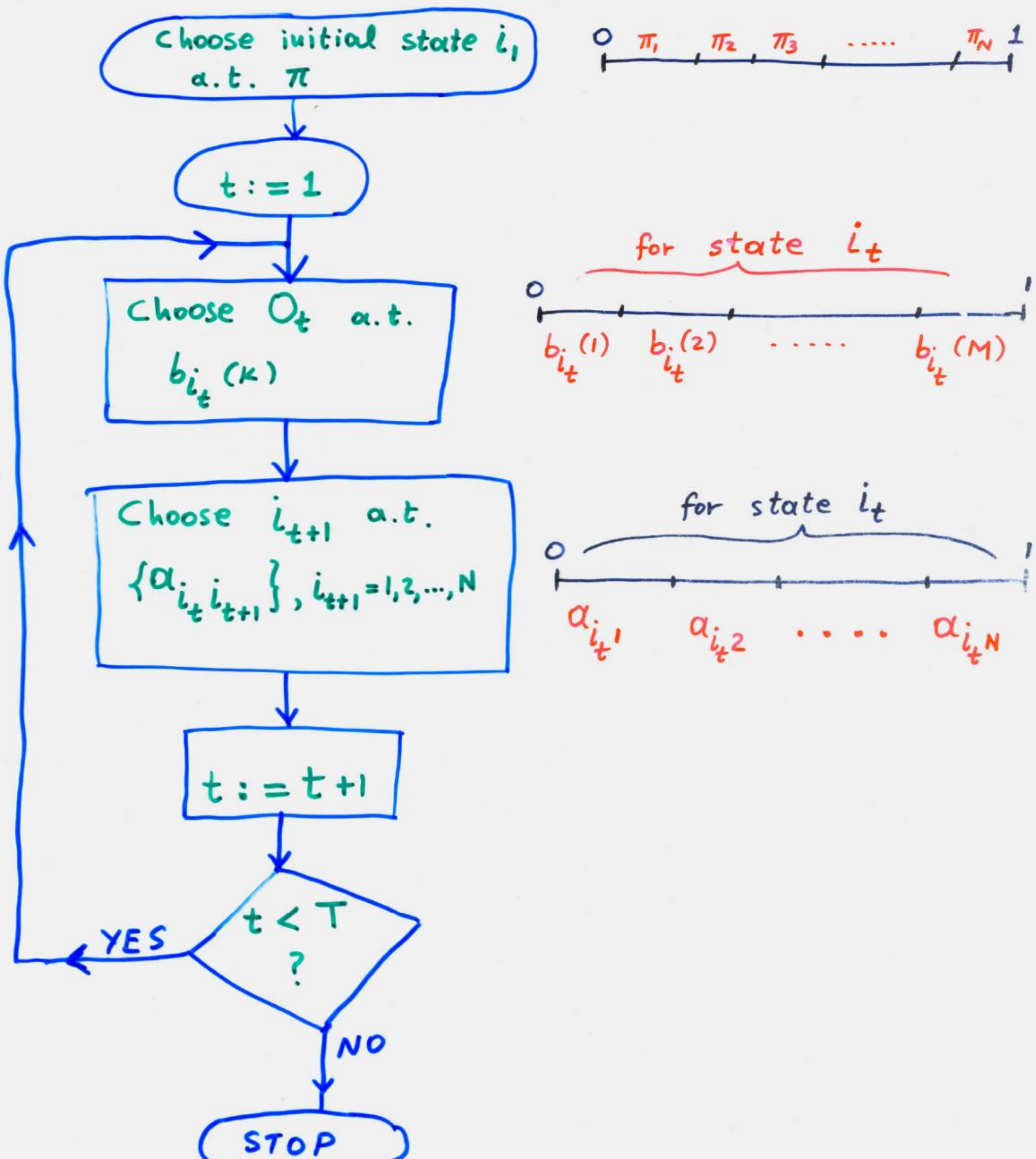
O = $H\ H\ T\ T\ H\ T\ H\ H\ T\ T\ H$ ...
S = 3 1 2 3 3 1 1 2 3 1 3 ...

STATE

	1	2	3
$P(H)$	P_1	P_2	P_3
$P(T)$	$1-P_1$	$1-P_2$	$1-P_3$


URN 1
URN 2
URN N
P(RED) = b1(1)
P(BLUE) = b1(2)
P(GREEN) = b1(3)
P(YELLOW) = b1(4)
P(ORANGE) = b1(M)
P(RED) = b2(1)
P(BLUE) = b2(2)
P(GREEN) = b2(3)
P(YELLOW) = b2(4)
P(ORANGE) = b2(M)
P(RED) = bN(1)
P(BLUE) = bN(2)
P(GREEN) = bN(3)
P(YELLOW) = bN(4)
P(ORANGE) = bN(M)
O = {GREEN, GREEN, BLUE, RED, YELLOW, RED, ..., BLUE}

GENERATE OBSERV. SEQ. ($O_1, O_2, \ldots, O_T$) FROM HMM



PROBLEMS TO BE SOLVED IN HMM

Problem 1 : CLASSIFICATION (SCORING)

Given an observ. seq. $O = O_1 O_2 \cdots O_T$
and a model $\lambda = (\pi, A, B)$,
compute $P_r(O/\lambda)$.

Problem 2 : ESTIMATION

Given an observ. seq. $O = O_1 O_2 \cdots O_T$,
choose an optimum state seq. $q_1 q_2 \cdots q_T$

Problem 3 : TRAINING

Adjust model parameters $\lambda = (\pi, A, B)$
to maximize $P_r(O/\lambda)$.

Forward Algorithm

$$\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2\ldots\mathbf{o}_t, q_t = i|\lambda)$$

1. Initialization

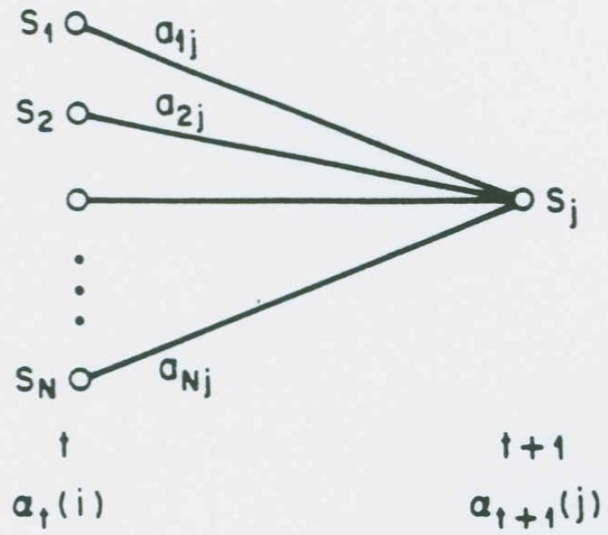
$$\alpha_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N$$

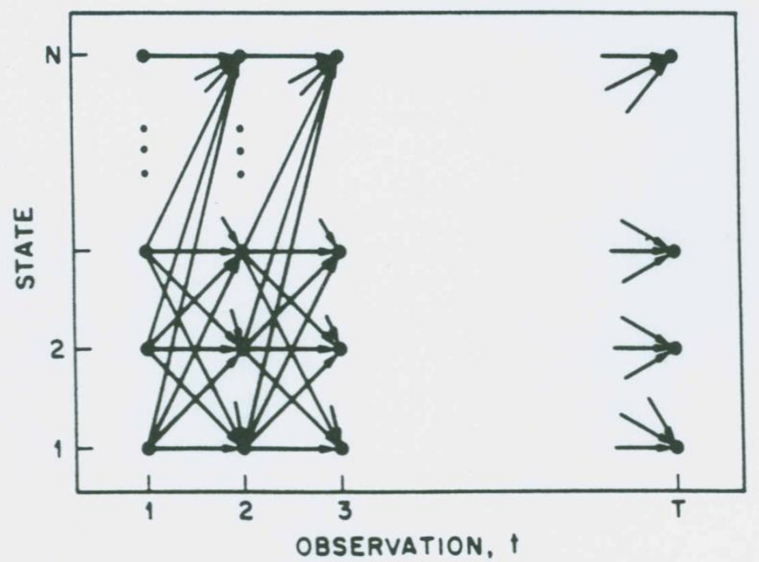
2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i)a_{ij}\right] b_j(\mathbf{o}_{t+1}), \qquad \begin{array}{l} 1 \le t \le T-1 \\ 1 \le j \le N \end{array}$$

3. Termination

$$P(\mathbf{O}|\lambda) = \sum_{i=1}^{N} \alpha_T(i)$$





Backward Algorithm

$$\beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\ldots\mathbf{o}_T | q_t = i, \lambda)$$

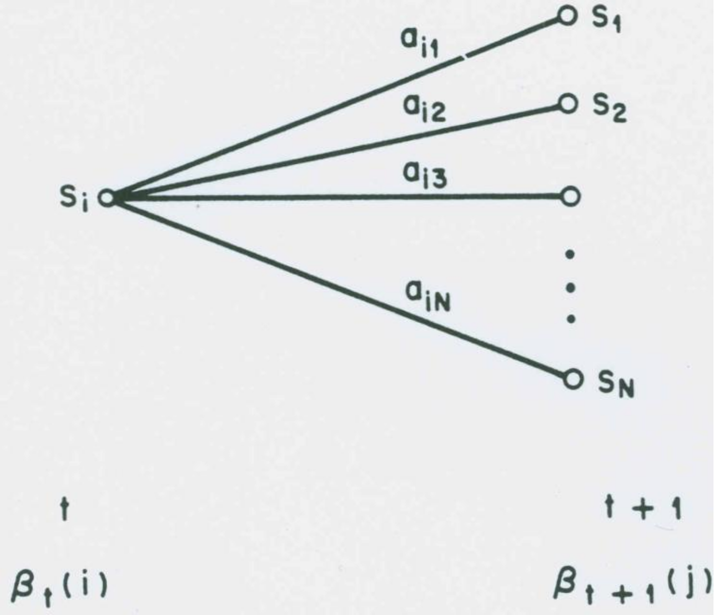
1. Initialization

$$\beta_T(i) = 1, \qquad 1 \le i \le N$$

2. Induction

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j),$$

$$t = T-1, T-2, \ldots, 1, \qquad 1 \le i \le N$$



Probability Functions for Local State Estimation

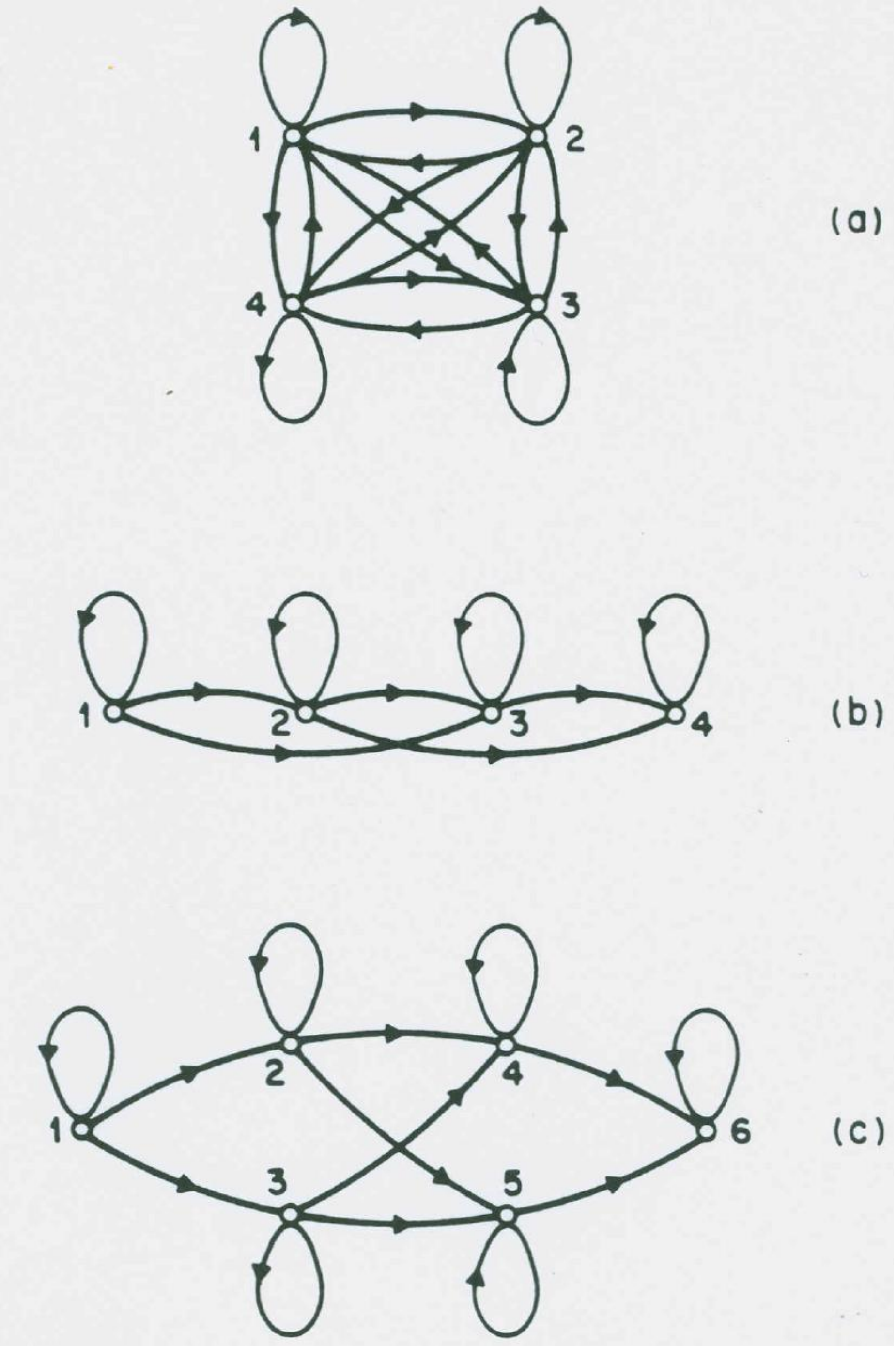
$$\alpha_t(i) = P(\mathbf{o}_1\mathbf{o}_2\ldots\mathbf{o}_t, q_t = i|\lambda) \qquad \beta_t(i) = P(\mathbf{o}_{t+1}\mathbf{o}_{t+2}\ldots\mathbf{o}_T|q_t = i, \lambda)$$

$$\gamma_t(i) = P(q_t = i|\mathbf{O}, \lambda)$$

$$\gamma_t(i) = P(q_t = i \mid \mathbf{O}, \lambda) = \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{P(\mathbf{O} \mid \lambda)} = \frac{P(\mathbf{O}, q_t = i \mid \lambda)}{\sum_{i=1}^{N} P(\mathbf{O}, q_t = i \mid \lambda)}$$

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}$$

$$q_t = \underset{1 \le i \le N}{\operatorname{argmax}} \; \gamma_t(i)$$



Εργοδικο

Left-Right Serial

Left-Right Parallel

HMM: State Estimation, Viterbi Algorithm

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \ldots q_{t-1},\ q_t = i,\ \mathbf{o}_1 \mathbf{o}_2 \ldots \mathbf{o}_t | \lambda]$$

1. Initialization

$$\delta_1(i) = \pi_i b_i(\mathbf{o}_1), \qquad 1 \le i \le N$$

$$\psi_1(i) = 0.$$

2. Recursion

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}] b_j(\mathbf{o}_t), \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N \end{matrix}$$

$$\psi_t(j) = \arg \max_{1 \le i \le N} [\delta_{t-1}(i)\, a_{ij}], \qquad \begin{matrix} 2 \le t \le T \\ 1 \le j \le N. \end{matrix}$$

3. Termination

$$P^* = \max_{1 \le i \le N} [\delta_T(i)]$$

$$q_T^* = \arg \max_{1 \le i \le N} [\delta_T(i)]$$

Viterbi Score

$$P^* = \Pr(O \mid Q^*, \lambda)$$

4. Path (state sequence) backtracking

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \ldots, 1$$

Example of State Estimation via Viterbi Algorithm

Given the model of the coin-toss experiment used in Exercise 6.2 (i.e., three different coins) with probabilities

	State 1	State 2	State 3
$P(H)$	0.5	0.75	0.25
$P(T)$	0.5	0.25	0.75

and with all state transition probabilities equal to 1/3, and with initial probabilities equal to 1/3, for the observation sequence

$$\mathbf{O} = (H H H H T H T T T T)$$

find the most likely path with the Viterbi algorithm.

Solution 6.3

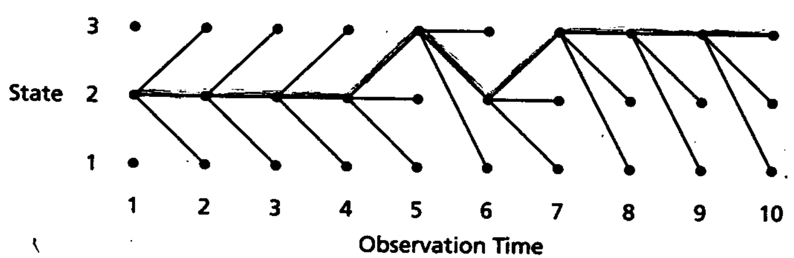
Since all a_{ij} terms are equal to 1/3, we can omit these terms (as well as the initial state probability term), giving

$$\delta_1(1) = 0.5, \quad \delta_1(2) = 0.75, \quad \delta_1(3) = 0.25.$$

The recursion for $\delta_t(j)$ gives $(2 \le t \le 10)$

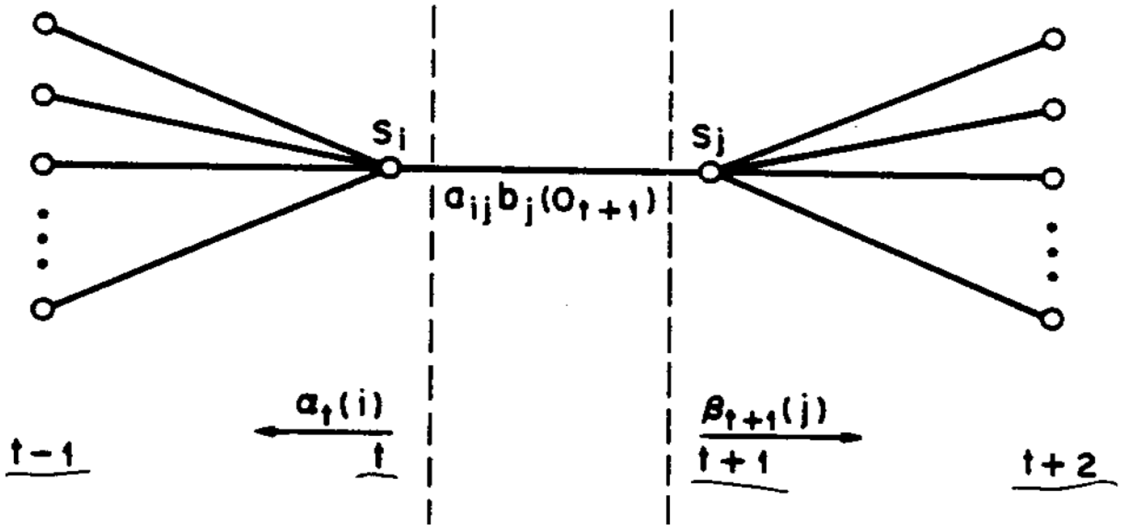
$\delta_2(1) = (0.75)(0.5)$,	$\delta_2(2) = (0.75)^2$,	$\delta_2(3) = (0.75)(0.25)$
$\delta_3(1) = (0.75)^2(0.5)$,	$\delta_3(2) = (0.75)^3$,	$\delta_3(3) = (0.75)^2(0.25)$
$\delta_4(1) = (0.75)^3(0.5)$,	$\delta_4(2) = (0.75)^4$,	$\delta_4(3) = (0.75)^3(0.25)$
$\delta_5(1) = (0.75)^4(0.5)$,	$\delta_5(2) = (0.75)^4(0.25)$,	$\delta_5(3) = (0.75)^5$
$\delta_6(1) = (0.75)^5(0.5)$,	$\delta_6(2) = (0.75)^6$,	$\delta_6(3) = (0.75)^5(0.25)$
$\delta_7(1) = (0.75)^6(0.5)$,	$\delta_7(2) = (0.75)^6(0.25)$,	$\delta_7(3) = (0.75)^7$
$\delta_8(1) = (0.75)^7(0.5)$,	$\delta_8(2) = (0.75)^7(0.25)$,	$\delta_8(3) = (0.75)^8$
$\delta_9(1) = (0.75)^8(0.5)$,	$\delta_9(2) = (0.75)^8(0.25)$,	$\delta_9(3) = (0.75)^9$
$\delta_{10}(1) = (0.75)^9(0.5)$,	$\delta_{10}(2) = (0.75)^9(0.25)$,	$\delta_{10}(3) = (0.75)^{10}$

This leads to a diagram (trellis) of the form:



Hence, the most likely state sequence is {2, 2, 2, 2, 3, 2, 3, 3, 3, 3}.

Probability Functions for HMM Parameter Estimation - I



$$\xi_t(i,j) = P(q_t = i,\ q_{t+1} = j \mid \mathbf{O}, \lambda)$$

$$\begin{aligned}
\xi_t(i,j) &= \frac{P(q_t = i,\ q_{t+1} = j,\ \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)} \\
&= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{P(\mathbf{O} \mid \lambda)} \\
&= \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}
\end{aligned}$$

Probability Functions for HMM Parameter Estimation - II

$$\xi_t(i,j) = P(q_t = i,\ q_{t+1} = j | \mathbf{O}, \lambda).$$

$$\xi_t(i,j) = \frac{P(q_t = i,\ q_{t+1} = j,\ \mathbf{O} \mid \lambda)}{P(\mathbf{O} \mid \lambda)} = \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{P(\mathbf{O}|\lambda)} = \frac{\alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i)\, a_{ij} b_j(\mathbf{o}_{t+1}) \beta_{t+1}(j)}.$$

$$\gamma_t(i) = \sum_{j=1}^{N} \xi_t(i,j).$$

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{expected number of transitions from state } i \text{ in } \mathbf{O}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{expected number of transitions from state } i \text{ to state } j \text{ in } \mathbf{O}.$$

Reestimation of HMM Parameters

$$\bar{\pi}_i = \text{expected frequency (number of times) in state } i \text{ at time } (t=1) = \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{expected number of transitions from state } i \text{ to state } j}{\text{expected number of transitions from state } i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{expected number of times in state } j \text{ and observing symbol } \mathbf{v}_k}{\text{expected number of times in state } j} = \frac{\sum_{\substack{t=1 \\ \text{s.t. } \mathbf{o}_t = \mathbf{v}_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}.$$

$$Q(\lambda', \lambda) = \sum_{\mathbf{q}} P(\mathbf{O}, \mathbf{q} | \lambda') \log P(\mathbf{O}, \mathbf{q} | \lambda)$$

$$Q(\lambda', \lambda) \geq Q(\lambda', \lambda') \Rightarrow P(\mathbf{O}|\lambda) \geq P(\mathbf{O}|\lambda')$$

HMM Continuous Densities

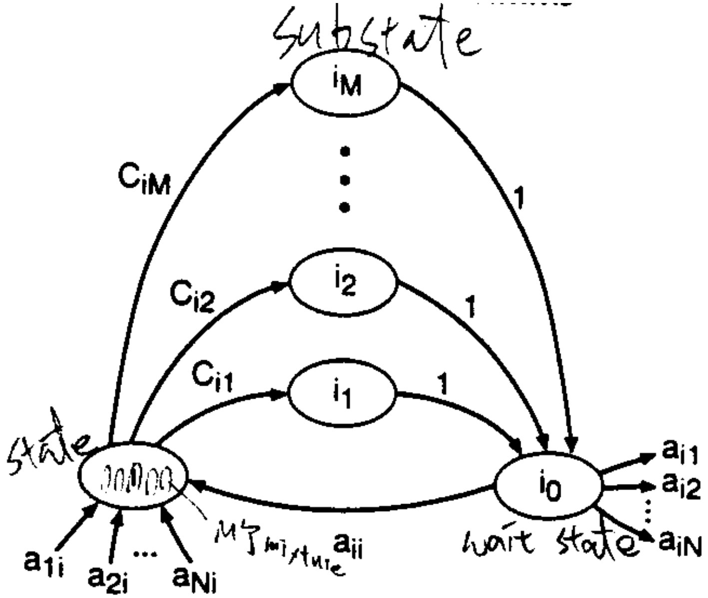


Figure 6.9 Equivalence of a state with a mixture density to a multistate single-density distribution (after Juang et al. [21]).

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}), \qquad 1 \le j \le N$$

$$\sum_{k=1}^{M} c_{jk} = 1, \qquad 1 \le j \le N$$

$$c_{jk} \ge 0,\ 1 \le j \le N,\ 1 \le k \le M$$

$$\int_{-\infty}^{\infty} b_j(\mathbf{o}) d\mathbf{o} = 1, \qquad 1 \le j \le N.$$

HMM Parameter Estimation for Continuous Densities

mixture

$$b_j(\mathbf{o}) = \sum_{k=1}^{M} c_{jk} \mathcal{N}(\mathbf{o}, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk}), \qquad 1 \le j \le N$$

$$\sum_{k=1}^{M} c_{jk} = 1, \qquad 1 \le j \le N$$

$$\int_{-\infty}^{\infty} b_j(\mathbf{o}) d\mathbf{o} = 1, \qquad 1 \le j \le N$$

$$c_{jk} \ge 0, \; 1 \le j \le N, \; 1 \le k \le M$$

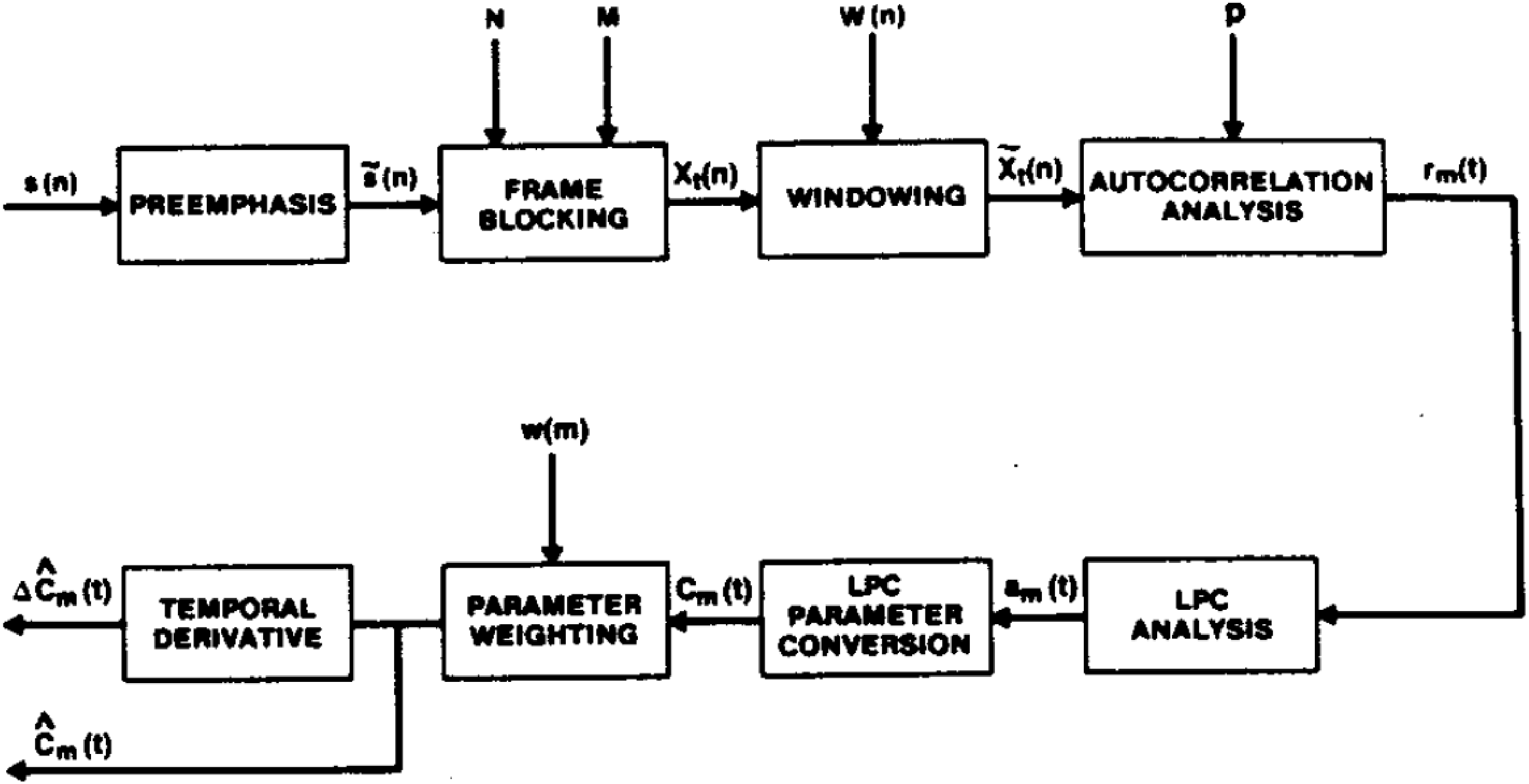
$$\bar{c}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k)}{\sum_{t=1}^{T} \sum_{k=1}^{M} \gamma_t(j,k)}$$

$$\bar{\boldsymbol{\mu}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \mathbf{o}_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\bar{\mathbf{U}}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot (\mathbf{o}_t - \boldsymbol{\mu}_{jk})(\mathbf{o}_t - \boldsymbol{\mu}_{jk})'}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} \mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}_{jk}, \mathbf{U}_{jk})}{\sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o}_t, \boldsymbol{\mu}_{jm}, \mathbf{U}_{jm})} \right]$$

LPC Processor for Speech Recognition



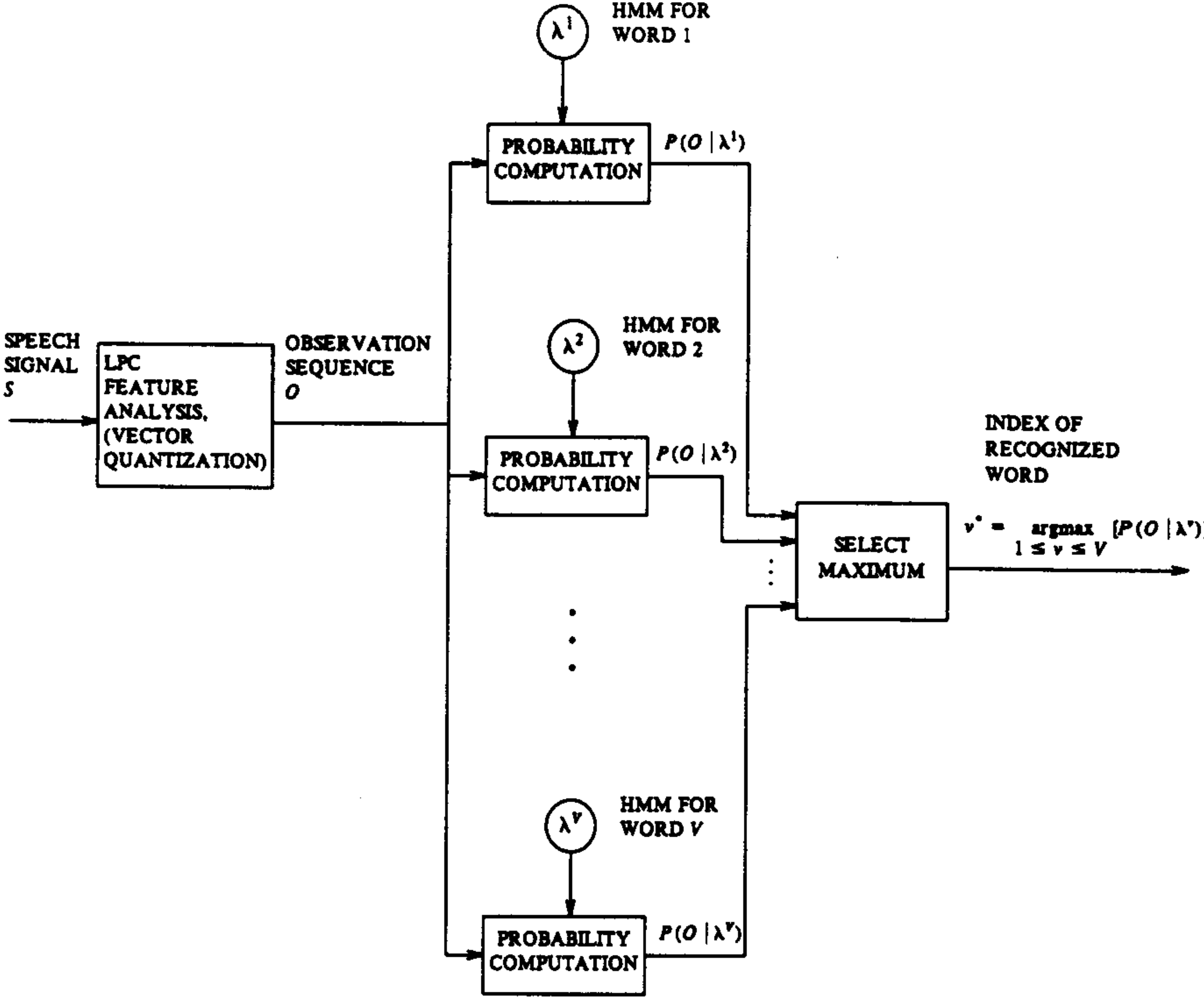


Figure 6.13 Block diagram of an isolated word HMM recognizer (after Rabiner [38]).

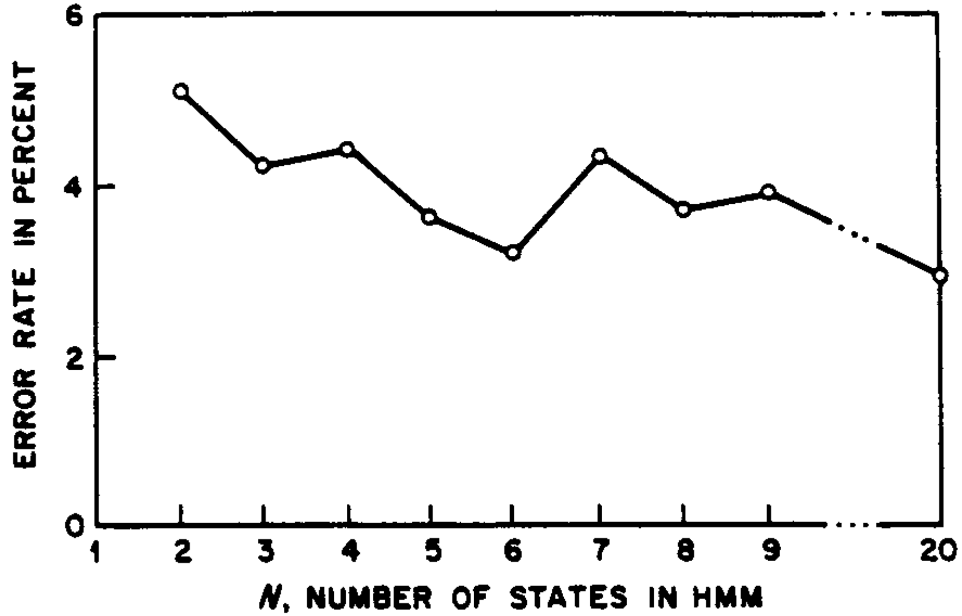


Figure 6.14 Average word error rate (for a digits vocabulary) versus the number of states N in the HMM (after Rabiner et al. [18]).

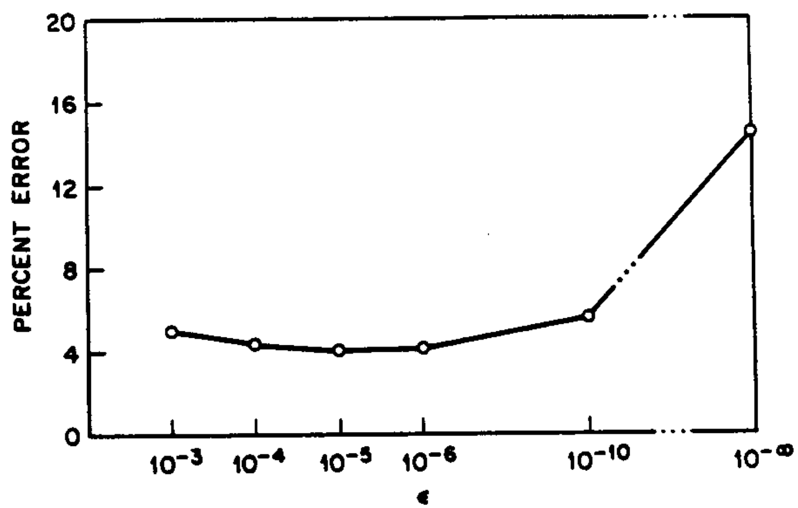


Figure 6.16 Average word error rate as a function of the minimum discrete density value ϵ (after Rabiner et al. [18]).

Probability Distributions of Cepstral Coefs of /zero/

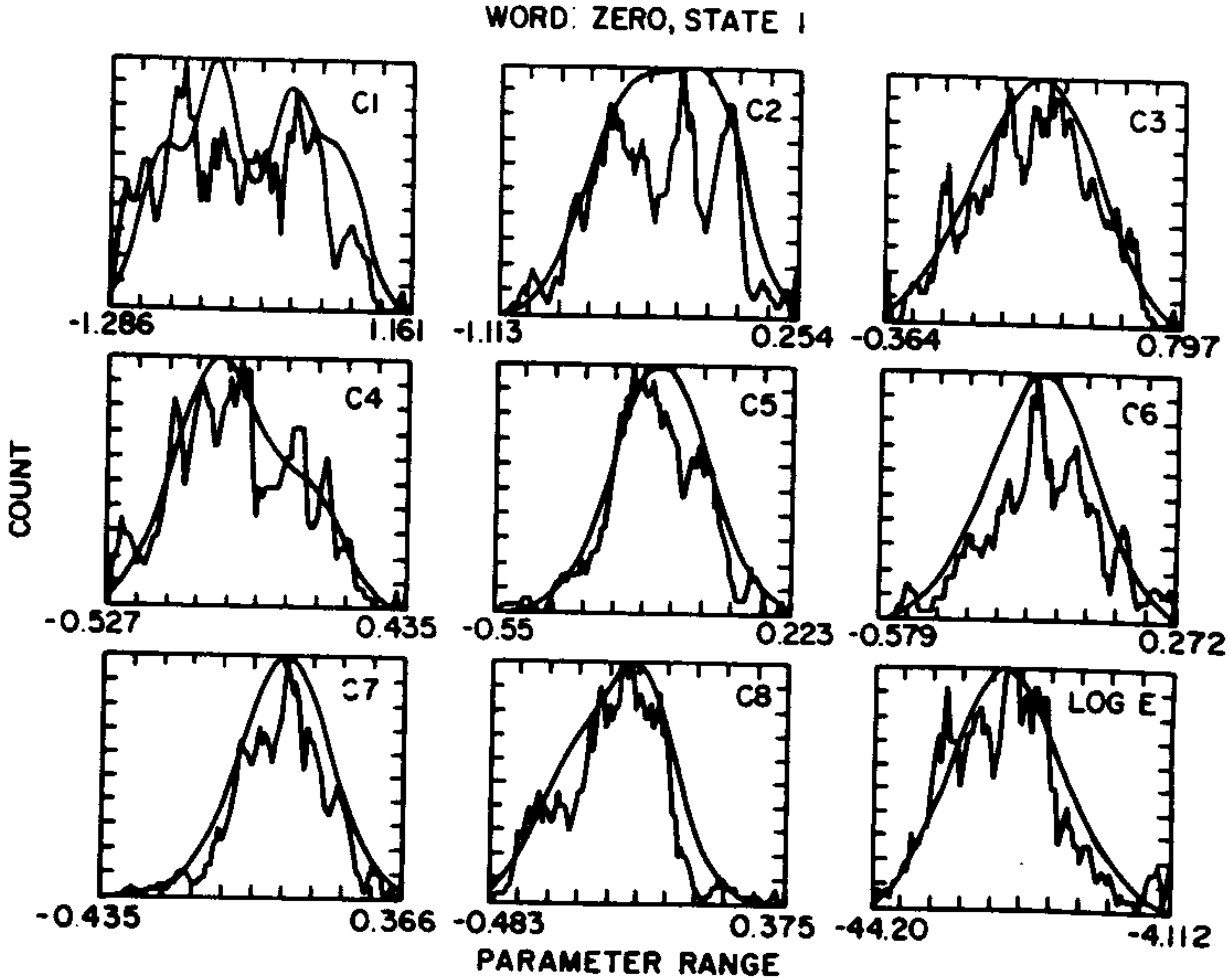
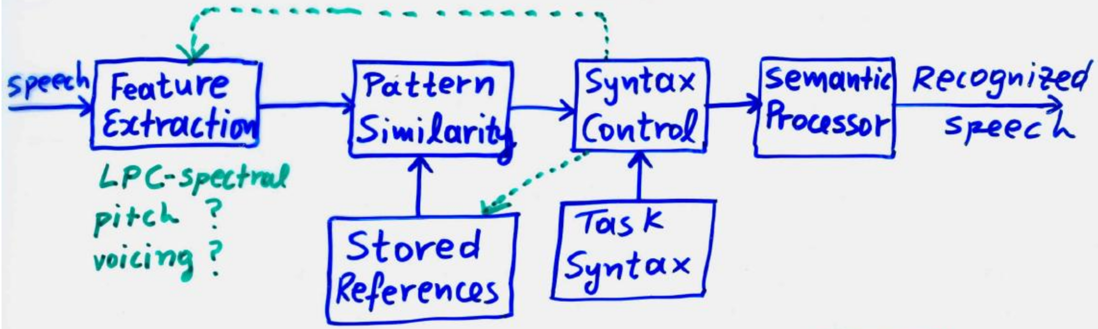


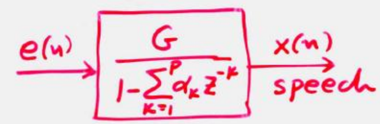
Figure 6.15 Comparison of estimated density (jagged contour) and model density (smooth contour) for each of the nine components of the observation vector (eight cepstral components, one log energy component) for state 1 of the digit zero (after Rabiner et al. [38]).

Dynamic Time Warping (DTW)

ASR by DTW (dynamic time-warp): Matching Time-Aligned Templates



* **Features** : LPC model parameters



* **Distance (dissimilarity) measure** : Itakura-Saito LPC distance

* Given N speech samples $\{x(1), \cdots, x(N)\} = X$, the LIKELIHOOD that X comes from the model $\{G, \alpha\} = P$, is

$$L(X/P) = -\frac{N}{2}\left[\log 2\pi G^2 + \frac{\alpha R \alpha^T}{G^2}\right],$$

$\vec{\alpha} = (1, -\alpha_1, -\alpha_2, \cdots, -\alpha_p)$: LPC vector

$R = [R(|i-j|)], \ (i,j = 0,1,\cdots,p)$: Correlation Matrix

$$R(i) = \frac{1}{N}\sum_{n=1}^{N-i} x(n)x(n+i).$$

* $\alpha R \alpha^T$ = energy of pred. error signal when $x(n)$ is predicted with LPC $\{\alpha_k\}$.

* $\partial L(X/P)/\partial G = 0 \implies G^2 = \alpha R \alpha^T$, $L'(X/\alpha) = \max_G L(X/P)$

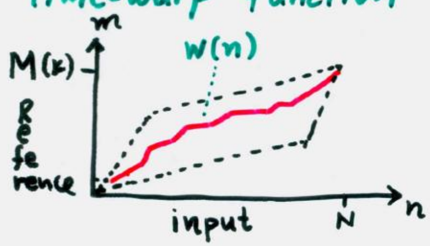
* $\frac{\partial L'(X/\alpha)}{\partial \alpha} = 0 \implies \boxed{\sum_{j=0}^{p} \hat{\alpha}_j R(i-j) = 0 \ , \ i = 1, \cdots, p}$

ISOLATED WORD RECOGNITION by DTW-template matching. (Itakura, 1975).

* Distance between speech sample vector X and LPC vector α:

$$d(X/\alpha) = \log\left(\frac{\alpha R \alpha^T}{\hat{\alpha} R \hat{\alpha}^T}\right)$$

* Partition a word into subunits: frames $m = 1, 2, \ldots, M(k)$.
* Stored Reference words $k = 1, 2, \ldots, K$.
  - each word has $M(k)$ frames
  - for each frame store its optimum LPC vector $\alpha(m,k)$
* For each input Test word to be recognized,
  - segment it into frames $n = 1, 2, \ldots, N$.
  - obtain correlations $R(n)$
  - find LPC $\hat{\alpha}(n)$
  - distance between n-th frame of input and m-th frame of k-th reference is $d(n,m;k) = \log \frac{\alpha(m,k) R(n) \alpha^T}{\hat{\alpha}(n) R(n) \hat{\alpha}^T}$
  - Time-warp function $m = w(n)$



$$D(k) = \min_{\{w(n)\}} \sum_{n=1}^{N} d(n, w(n); k).$$

distance between input word and k-th stored reference.

* Use Dynamic Programming to find $w(n)$ for each reference word.
* Recognize input word as the k^*-th ref. word,

$$k^* = \arg \min_{k=1}^{K} D(k).$$

Boundary conds. : $w(1) = 1$, $w(N) = M$

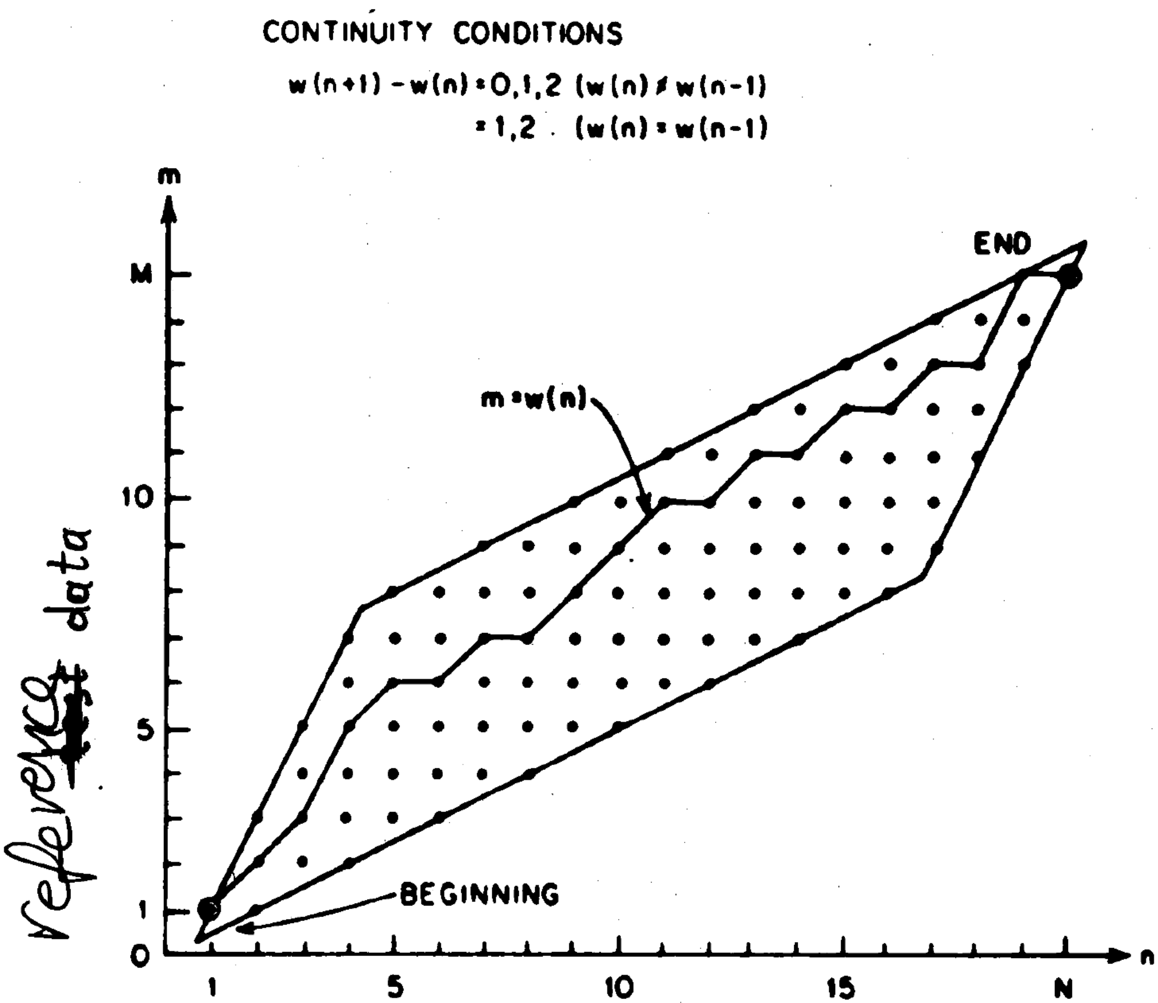


Fig. 9.17 An example of a typical warping function. (After Itakura [17].)

test ~~reference~~ data

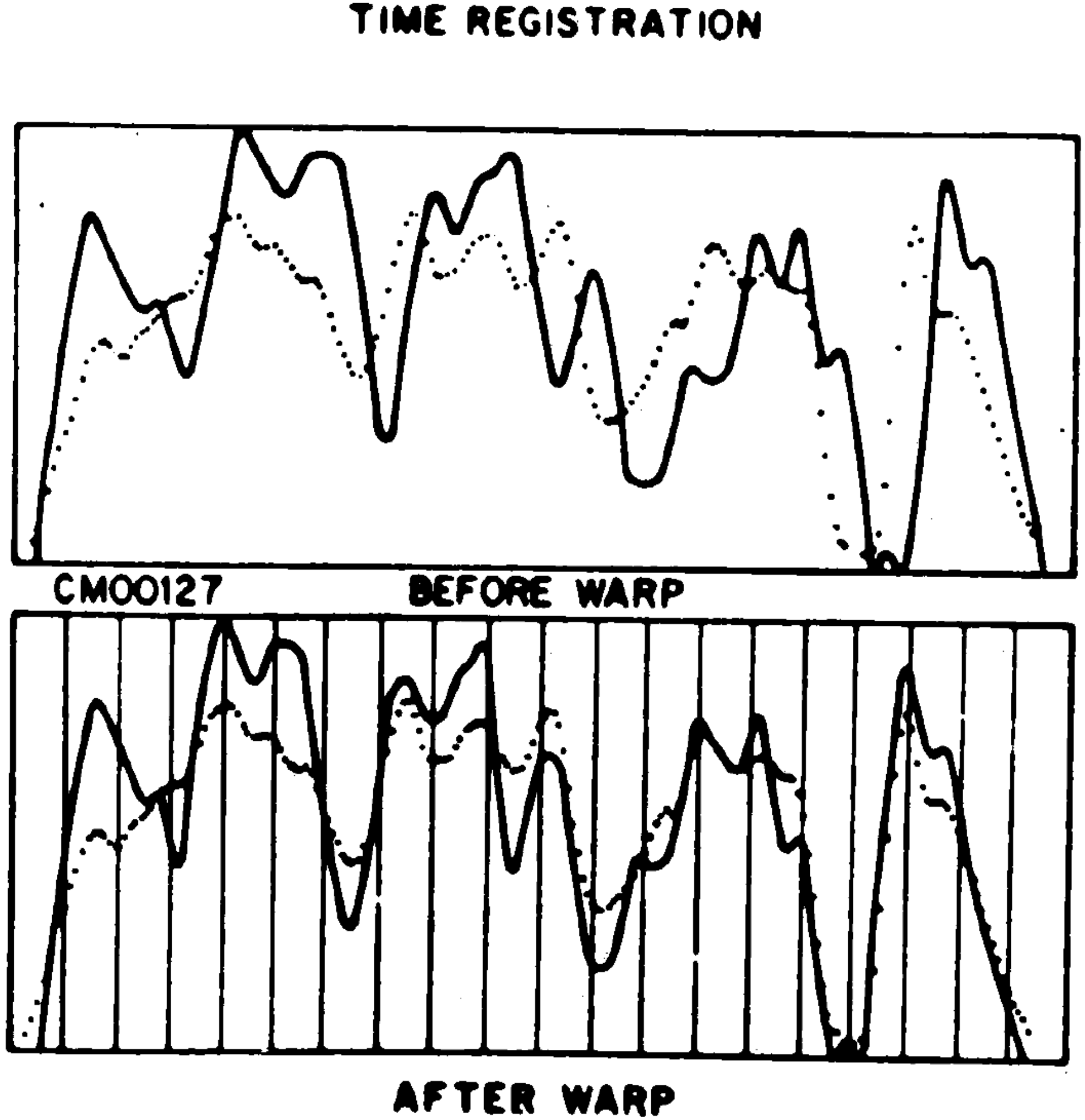


Fig. 9.18 An example of the effects of time warping on a speech intensity contour. (After Rosenberg [13].)

TABLE 6.1. Average Digit Error Rates for Several Recognizers and Evaluation Sets

Recognizer Type	Evaluation Set: Original Training	TS2	TS3	TS4
LPC/DTW	0.1	0.2	2.0	1.1
LPC/DTW/VQ	–	3.5	–	–
HMM/VQ	–	3.7	–	–
HMM/CD	0	0.2	1.3	1.8
HMM/AR	0.3	1.8	3.4	4.1

TS2 The same 100 talkers as were used in the training; 100 occurrences of each digit

TS3 A new set of 100 talkers (50 male, 50 female); 100 occurrences of each digit

TS4 Another new set of 100 talkers (50 male, 50 female); 100 occurrences of each digit

LPC/DTW Conventional template-based recognizer using dynamic time warping (DTW) alignment

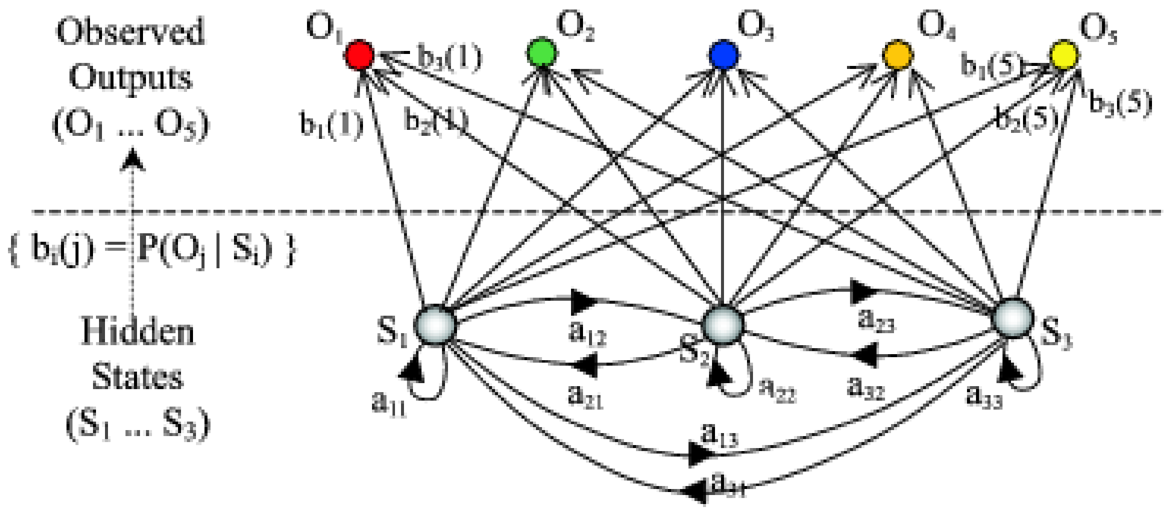
LPC/DTW/VQ Conventional recognizer with vector quantization of the feature vectors ($M = 64$)

HMM/VQ HMM recognizer with $M = 64$ codebook

HMM/CD HMM recognizer using continuous density model with $M = 5$ mixtures per state

HMM/AR HMM recognizer using autoregressive observation density

HMM (Hidden Markov Models)



- ***t*** **=** 1, 2, 3, …: **Discrete Time**
- ***O*** **=** $(O_1, O_2, ..., O_T)$ **: Observation Sequence**
- ***T*** **= Length of Observation Sequence**
- ***N*** **= Number of States**
- ***M*** **= # of Observation Symbols / Mixtures**
- **States** $S_1, S_2, \ldots, S_N$

HMM: $\lambda = (A, B, \pi)$

- ***A*** = $[a_{ij}], \quad a_{ij} = \Pr\ S_j \text{ at } t{+}1 \mid S_i \text{ at } t$

  State Transition Probability Matrix

- ***B*** = $b_j(k) \ , \quad b_j(k) = \Pr\ v_k \text{ at } t \mid S_j \text{ at } t$

  Observations Probability Distributions

- **π =** $\pi_i \ , \ \pi_i = \Pr\ q_i \text{ at } t{=}1$

  Initial State Probability

Problems to Be Solved in HMM

- **<u>Problem 1</u>: Classification – Scoring (*Forward-Backward Algorithm*)**

  Given an observed sequence $O=(O_1,O_2,\ldots,O_T)$ **and a model** ***λ=(π, A, B)*,**
  compute likelihood $\Pr(O\mid\lambda)$

- **<u>Problem 2</u>: State Estimation (*Viterbi Algorithm*)**

  Given an observed sequence $O=(O_1,O_2,\ldots,O_T)$ **estimate an optimum state sequence** $Q^*=(q_1,q_2,\ldots,q_T)$ **and compute the score** $\Pr(O,Q^*\mid\lambda)$

- **<u>Problem 3</u>: Training (*EM Algorithm*)**

  Given an observed sequence $O=(O_1,O_2,\ldots,O_T)$ **adjust model parameters** ***λ=(π, A, B)*** **to maximize likelihood** $\Pr(O\mid\lambda)$