Digital Speech Processing—Lecture 20

The Hidden Markov Model (HMM)

Lecture Outline

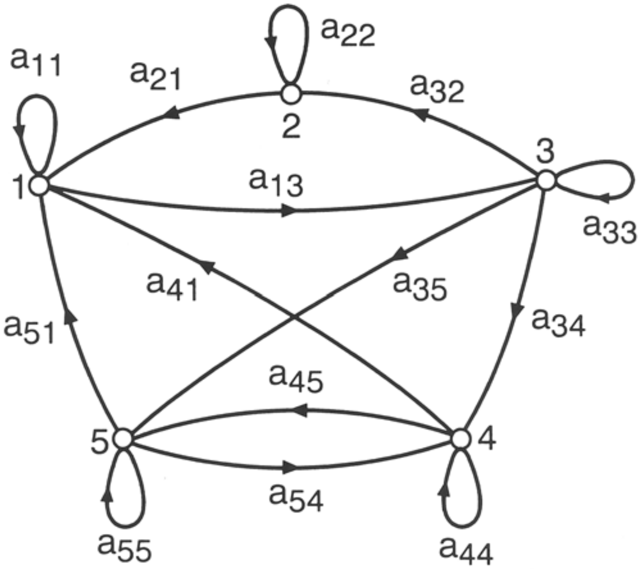
- Theory of Markov Models
  - discrete Markov processes
  - hidden Markov processes
- Solutions to the Three Basic Problems of HMM's
  - computation of observation probability
  - determination of optimal state sequence
  - optimal training of model
- Variations of elements of the HMM
  - model types
  - densities
- Implementation Issues
  - scaling
  - multiple observation sequences
  - initial parameter estimates
  - insufficient training data
- Implementation of Isolated Word Recognizer Using HMM's

Stochastic Signal Modeling

- Reasons for Interest:
  - basis for theoretical description of signal processing algorithms
  - can learn about signal source properties
  - models work well in practice in real world applications
- Types of Signal Models
  - deteministic, parametric models
  - stochastic models

Discrete Markov Processes

System of *N* distinct states, $\{S_1, S_2, \ldots, S_N\}$



Time(t)	1	2	3	4	5	...
State	q_1	q_2	q_3	q_4	q_5	...

Markov Property:

$$\boxed{P\left[q_t = S_i \mid q_{t-1} = S_j, q_{t-2} = S_k, \ldots\right] = P\left[q_t = S_i \mid q_{t-1} = S_j\right]}$$

Properties of State Transition Coefficients

Consider processes where state transitions are time independent, i.e.,

$$\boxed{a_{ji} = P\left[q_t = S_i \,|\, q_{t-1} = S_j \right], 1 \le i, j \le N}$$

$$\boxed{\begin{array}{l} a_{ji} \ge 0 \quad \forall j, i \\ \displaystyle\sum_{i=1}^{N} a_{ji} = 1 \ \ \forall j \end{array}}$$

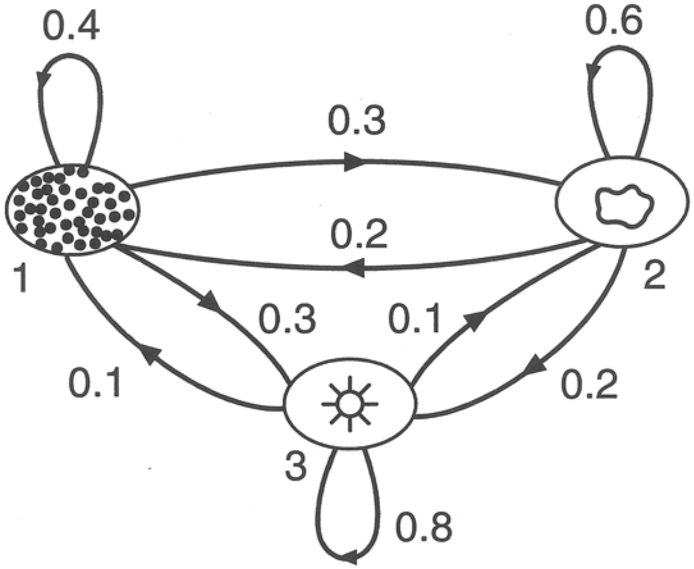
Example of Discrete Markov Process

Once each day (e.g., at noon), the weather is observed and classified as being one of the following:

- State 1—Rain (or Snow; e.g. precipitation)
- State 2—Cloudy
- State 3—Sunny

with state transition probabilities:

$$A = \{a_{ij}\} = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$



Discrete Markov Process

Problem: Given that the weather on day 1 is sunny, what is the probability (according to the model) that the weather for the next 7 days will be "sunny-sunny-rain-rain-sunny-cloudy-sunny"?

Solution: We define the observation sequence, O, as:

$$O = \left\{ S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \right\}$$

and we want to calculate P(O|Model). That is:

$$P(O \,|\, \text{Model}) = P\left[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \,|\, \text{Model} \right]$$

Discrete Markov Process

$$P(O \mid \text{Model}) = P\left[S_3, S_3, S_3, S_1, S_1, S_3, S_2, S_3 \mid \text{Model}\right]$$

$$= P\left[S_3\right] P\left[S_3 \mid S_3\right]^2 P\left[S_1 \mid S_3\right] P\left[S_1 \mid S_1\right]$$

$$\cdot P\left[S_3 \mid S_1\right] P\left[S_2 \mid S_3\right] P\left[S_3 \mid S_2\right]$$

$$= \pi_3 \left(a_{33}\right)^2 a_{31} a_{11} a_{13} a_{32} a_{23}$$

$$= 1 \left(0.8\right)^2 \left(0.1\right)\left(0.4\right)\left(0.3\right)\left(0.1\right)\left(0.2\right)$$

$$= 1.536 \cdot 10^{-04}$$

$$\boxed{\pi_i = P\left[q_1 = S_i\right], \quad 1 \le i \le N}$$

Discrete Markov Process

Problem: Given that the model is in a known state, what is the probability it stays in that state for exactly *d* days?

Solution:

$$O=\left\{\underset{1}{S_i},\underset{2}{S_i},\underset{3}{S_i},\ldots,\underset{d}{S_i},\underset{d+1}{S_j}\neq S_i\right\}$$

$$P\left(O\,|\,\text{Model},\ q_1=S_i\right)=\left(a_{ii}\right)^{d-1}(1-a_{ii})=p_i(d)$$

$$\overline{d}_i=\sum_{d=1}^{\infty}d\,p_i(d)=\frac{1}{1-a_{ii}}$$

Exercise

Given a single fair coin, i.e., *P* (H=Heads)= *P* (T=Tails) = 0.5, which you toss once and observe Tails:

a) what is the probability that the next 10 tosses will provide the sequence {H H T H T T H T T H}?

SOLUTION:

For a fair coin, with independent coin tosses, the probability of any specific observation sequence of length 10 (10 tosses) is $(1/2)^{10}$ since there are 2^{10} such sequences and all are equally probable. Thus:

$$P\ (\mathrm{H\ H\ T\ H\ T\ T\ H\ T\ T\ H}) = (1/2)^{10}$$

Exercise

b) what is the probability that the next 10 tosses will produce the sequence {H H H H H H H H H H}?

SOLUTION:

Similarly:

$$P\text{ (H H H H H H H H H H)} = (1/2)^{10}$$

Thus a specified run of length 10 is equally as likely as a specified run of interlaced H and T.

Exercise

c) what is the probability that 5 of the next 10 tosses will be tails? What is the expected number of tails over the next 10 tosses?

SOLUTION:

The probability of 5 tails in the next 10 tosses is just the number of observation sequences with 5 tails and 5 heads (in any sequence) and this is:

P (5H, 5T)=(10C5) $(1/2)^{10}$ = 252/1024≈0.25

since there are (10C5) combinations (ways of getting 5H and 5T) for 10 coin tosses, and each sequence has probability of $(1/2)^{10}$. The expected number of tails in 10 tosses is:

$$E(\text{Number of } T \text{ in 10 coin tosses}) = \sum_{d=0}^{10} d \binom{10}{d} \left(\frac{1}{2}\right)^{10} = 5$$

Thus, on average, there will be 5H and 5T in 10 tosses, but the probability of exactly 5H and 5T is only about 0.25.

Coin Toss Models

A series of coin tossing experiments is performed. The number of coins is unknown; only the results of each coin toss are revealed. Thus a typical observation sequence is:

$$O = O_1 O_2 O_3 \ldots O_T = HHTTTHTTH \ldots H$$

Problem: Build an HMM to explain the observation sequence.

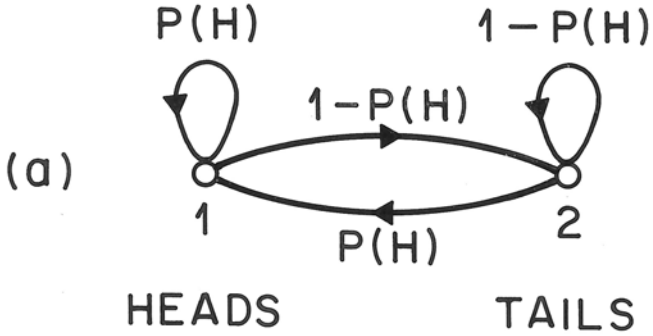
Issues:

1. What are the states in the model?

2. How many states should be used?

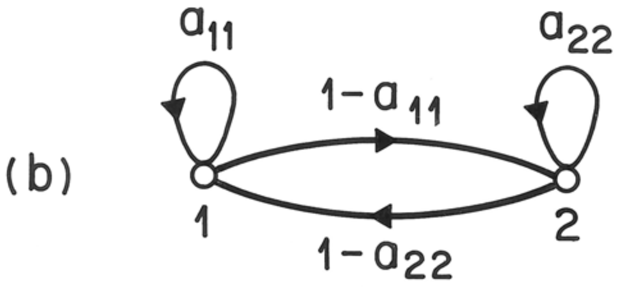
3. What are the state transition probabilities?

Coin Toss Models



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 . . .



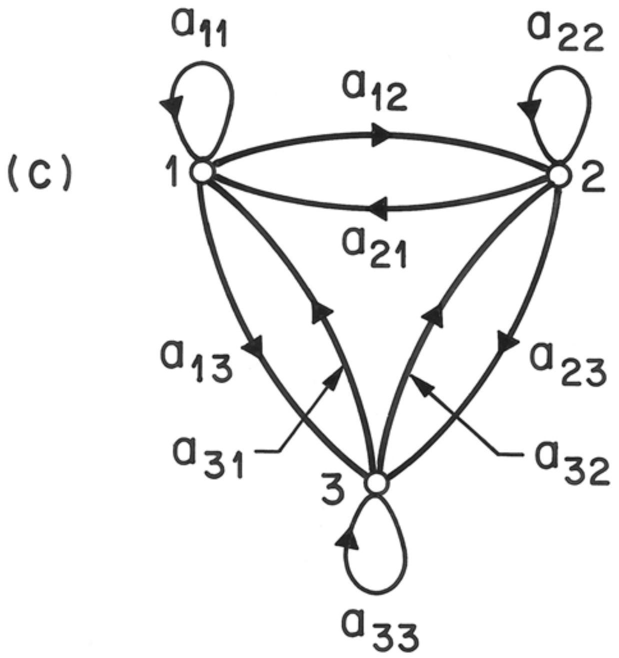
$P(H) = P_1$ $P(H) = P_2$

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

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 . . .

Coin Toss Models



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)	$1-P_1$	$1-P_2$	$1-P_3$

Coin Toss Models

Problem: Consider an HMM representation (model λ) of a coin tossing experiment. Assume a 3-state model (corresponding to 3 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. (Assume initial state probabilities of 1/3).

a) You observe the sequence: O=H H H H T H T T T T

What state sequence is most likely? What is the probability of the observation sequence and this most likely state sequence?

Coin Toss Problem Solution

SOLUTION:

Given O=HHHHTHTTTT, the most likely state sequence is the one for which the probability of each individual observation is maximum. Thus for each H, the most likely state is S_2 and for each T the most likely state is S_3. Thus the most likely state sequence is:

$$S = S_2\, S_2\, S_2\, S_2\, S_3\, S_2\, S_3\, S_3\, S_3\, S_3$$

The probability of O and S (given the model) is:

$$P(O,S \mid \lambda) = (0.75)^{10}\left(\frac{1}{3}\right)^{10}$$

Coin Toss Models

b) What is the probability that the observation sequence came entirely from state 1?

SOLUTION:

The probability of O given that S is of the form:

$$\hat{S} = S_1 S_1 S_1 S_1 S_1 S_1 S_1 S_1 S_1 S_1$$

is:

$$P(O, \hat{S} \mid \lambda) = (0.50)^{10} \left(\frac{1}{3} \right)^{10}$$

The ratio of $P(O, S \mid \lambda)$ to $P(O, \hat{S} \mid \lambda)$ is:

$$R = \frac{P(O, S \mid \lambda)}{P(O, \hat{S} \mid \lambda)} = \left(\frac{3}{2} \right)^{10} = 57.67$$

Coin Toss Models

c) Consider the observation sequence:

$$\tilde{O} = H\,T\,T\,HTHHTTH$$

How would your answers to parts a and b change?

SOLUTION:

Given $\tilde{O}$ which has the same number of H's and T's, the answers to parts a and b would remain the same as the most likely states occur the same number of times in both cases.

Coin Toss Models

d) If the state transition probabilities were of the form:

$$\begin{aligned} &a_{11} = 0.9, \quad && a_{21} = 0.45, \quad && a_{31} = 0.45 \\ &a_{12} = 0.05, \quad && a_{22} = 0.1, \quad && a_{32} = 0.45 \\ &a_{13} = 0.05, \quad && a_{23} = 0.45, \quad && a_{33} = 0.1 \end{aligned}$$

i.e., a new model λ', how would your answers to parts a-c change? What does this suggest about the type of sequences generated by the models?

Coin Toss Problem Solution

SOLUTION:

The new probability of *O* and *S* becomes:

$$P(O,S\,|\,\lambda') = (0.75)^{10}\left(\frac{1}{3}\right)(0.1)^{6}\,(0.45)^{3}$$

The new probability of *O* and $\hat{S}$ becomes:

$$P(O,\hat{S}\,|\,\lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^{9}$$

The ratio is:

$$R = \left(\frac{3}{2}\right)^{10}\left(\frac{1}{9}\right)^{6}\left(\frac{1}{2}\right)^{3} = 1.36\cdot 10^{-5}$$

Coin Toss Problem Solution

Now the probability of $\tilde{O}$ and S is not the same as the probability of O and S. We now have:

$$P(\tilde{O}, S \mid \lambda') = (0.75)^{10}\left(\frac{1}{3}\right)(0.45)^{6}(0.1)^{3}$$

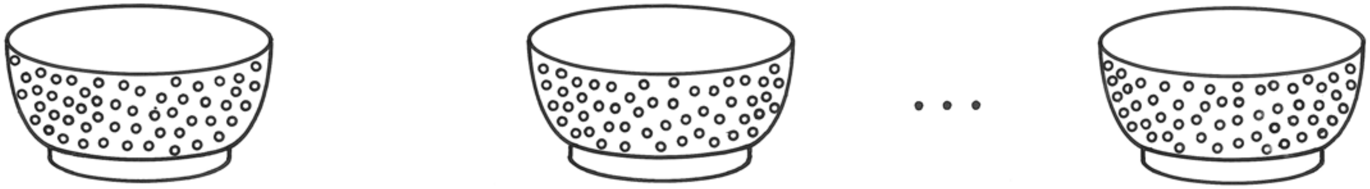
$$P(\tilde{O}, \hat{S} \mid \lambda') = (0.50)^{10}\left(\frac{1}{3}\right)(0.9)^{9}$$

with the ratio:

$$R = \left(\frac{3}{2}\right)^{10}\left(\frac{1}{2}\right)^{6}\left(\frac{1}{9}\right)^{3} = 1.24 \cdot 10^{-3}$$

Model λ, the initial model, clearly favors long runs of H's or T's, whereas model λ', the new model, clearly favors random sequences of H's and T's. Thus even a run of H's or T's is more likely to occur in state 1 for model λ', and a random sequence of H's and T's is more likely to occur in states 2 and 3 for model λ.

Balls in Urns Model



URN 1	URN 2	URN N
P(RED) = $b_1(1)$	P(RED) = $b_2(1)$	P(RED) = $b_N(1)$
P(BLUE) = $b_1(2)$	P(BLUE) = $b_2(2)$	P(BLUE) = $b_N(2)$
P(GREEN) = $b_1(3)$	P(GREEN) = $b_2(3)$	P(GREEN) = $b_N(3)$
P(YELLOW) = $b_1(4)$	P(YELLOW) = $b_2(4)$	P(YELLOW) = $b_N(4)$
⋮	⋮	⋮
P(ORANGE) = $b_1(M)$	P(ORANGE) = $b_2(M)$	P(ORANGE) = $b_N(M)$

O = {GREEN, GREEN, BLUE, RED, YELLOW, RED, ..., BLUE}

Elements of an HMM

1. N, number of states in the model

- states, $S = \{S_1, S_2, ..., S_N\}$
- state at time t, $q_t \in S$

2. M, number of distinct observation symbols per state

- observation symbols, $V = \{v_1, v_2, ..., v_M\}$
- observation at time t, $O_t \in V$

3. State transition probability distribution, $A = \{a_{ij}\}$,

$$a_{ij} = P(q_{t+1} = S_j \mid q_t = S_i), \quad 1 \le i, j \le N$$

4. Observation symbol probability distribution in state j

$$B = \{b_j(k)\}$$

$$b_j(k) = P\left[v_k \text{ at } t \mid q_t = S_j\right], \quad 1 \le j \le N, \ 1 \le k \le M$$

5. Initial state distribution, $\Pi = \{\pi_i\}$

$$\pi_i = P\left[q_1 = S_i\right], \ 1 \le i \le N$$

HMM Generator of Observations

1. Choose an initial state, $q_1 = S_i$, according to the initial state distribution, Π.

2. Set $t = 1$.

3. Choose $O_t = v_k$ according to the symbol probability distribution in state S_i, namely $b_i(k)$.

4. Transit to a new state, $q_{t+1} = S_j$ according to the state transition probability distribution for state S_i, namely a_{ij}.

5. Set $t = t + 1$; return to step 3 if $t \leq T$; otherwise terminate the procedure.

t	1	2	3	4	5	6	…	T
state	q_1	q_2	q_3	q_4	q_5	q_6	…	q_T
observation	O_1	O_2	O_3	O_4	O_5	O_6	…	O_T

Notation: $\lambda = (A, B, \Pi)$--HMM

Three Basic HMM Problems

Problem 1--Given the observation sequence, $O = O_1 O_2 \ldots O_T$, and a model $\lambda = (A, B, \Pi)$, how do we (efficiently) compute $P(O|\lambda)$, the probability of the observation sequence?

Problem 2--Given the observation sequence, $O = O_1 O_2 \ldots O_T$, how do we choose a state sequence $Q = q_1 q_2 \ldots q_T$ which is optimal in some meaningful sense?

Problem 3--How do we adjust the model parameters $\lambda = (A, B, \Pi)$ to maximize $P(O|\lambda)$?

Interpretation:

Problem 1--Evaluation or scoring problem.

Problem 2--Learn structure problem.

Problem 3--Training problem.

Solution to Problem 1—P(O|λ)

Consider the ***fixed*** state sequence (there are N^T such sequences):

$$Q = q_1 q_2 \ldots q_T$$

Then

$$P(O|Q,\lambda) = b_{q_1}(O_1) \cdot b_{q_2}(O_2) \ldots b_{q_T}(O_T)$$

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}$$

and

$$P(O,Q|\lambda) = P(O|Q,\lambda) \cdot P(Q|\lambda)$$

Finally

$$P(O|\lambda) = \sum_{\text{all } Q} P(O,Q|\lambda)$$

$$\boxed{P(O|\lambda) = \sum_{q_1, q_2, \ldots, q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T)}$$

Calculations required $\approx 2T \cdot N^T$; $N = 5, T = 100 \Rightarrow 2 \cdot 100 \cdot 5^{100}$

$\approx 10^{72}$ computations!

The “Forward” Procedure

Consider the forward variable, $\alpha_t(i)$, defined as the probability of the partial observation sequence (until time t) **and** state S_i at time t, given the model, i.e.,

$$\boxed{\alpha_t(i) = P(O_1 O_2 \ldots O_t, q_t = S_i \mid \lambda)}$$

Inductively solve for $\alpha_t(i)$ as:

1. **Initialization**

$$\alpha_1(i) = \pi_i \, b_i(O_1), \quad 1 \le i \le N$$

2. **Induction**

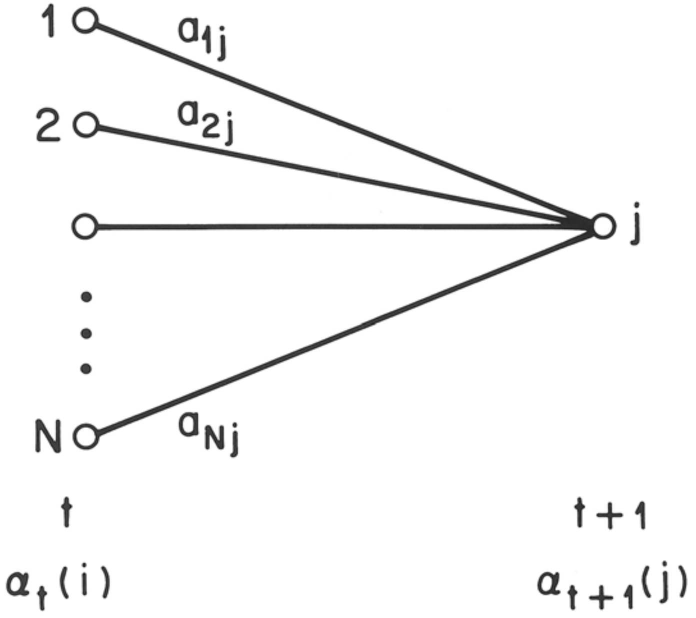
$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij} \right] b_j(O_{t+1}), \quad 1 \le t \le T-1, \, i \le j \le N$$

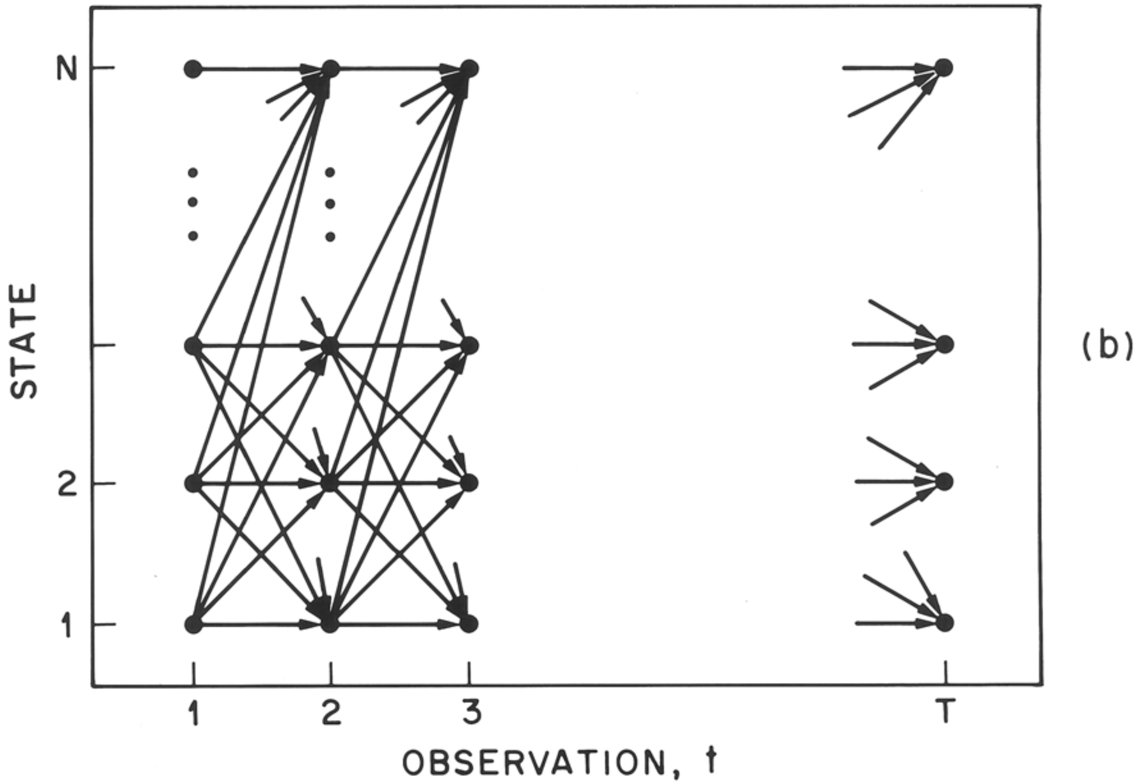
3. **Termination**

$$P(O|\lambda) = \sum_{i=1}^{N} P(O_1 O_2 \ldots O_T, q_T = S_i \mid \lambda) = \sum_{i=1}^{N} \alpha_T(i)$$

Computation: $N^2 T$ versus $2TN^T$; $N = 5, T = 100 \Rightarrow 2500$ versus 10^{72}

The “Forward” Procedure

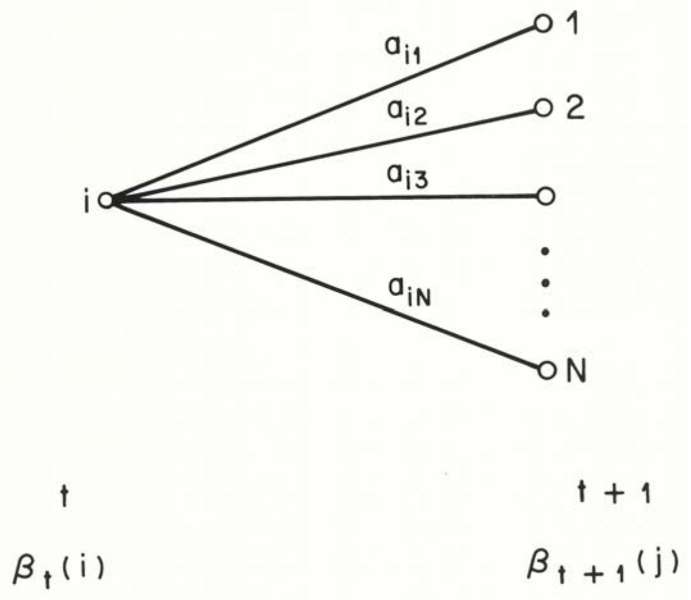




The “Backward” Algorithm

Consider the backward variable, $\beta_t(i)$, defined as the probability of the partial observation sequence from $t+1$ to the end, given state S_i at time t, and the model, i.e.,

$$\boxed{\beta_t(i) = P(O_{t+1}\, O_{t+2} \ldots O_T \,|\, q_t = S_i, \lambda)}$$



Inductive Solution :

1. **Initialization**

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

2. **Induction**

$$\beta_t(i) = \sum_{j=1}^{N} a_{ij}\, b_j(O_{t+1})\, \beta_{t+1}(j), \quad t = T-1, T-2, \ldots, 1, \; 1 \le i \le N$$

· N^2T calculations, same as in forward case

Solution to Problem 2—Optimal State Sequence

1. Choose states, q_t, which are *individually* most likely $\Rightarrow$ maximize expected number of correct individual states
2. Choose states, q_t, which are *pair - wise* most likely $\Rightarrow$ maximize expected number of correct state pairs
3. Choose states, q_t, which are *triple - wise* most likely $\Rightarrow$ maximize expected number of correct state triples
4. Choose states, q_t, which are *T - wise* most likely $\Rightarrow$ find the single best state sequence which maximizes $P(Q,O|\lambda)$

This solution is often called the Viterbi state sequence because it is found using the Viterbi algorithm.

Maximize Individual States

We define $\gamma_t(i)$ as the probability of being in state S_i at time t, given the observation sequence, and the model, i.e.,

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

then

$$\boxed{\gamma_t(i) = \frac{P(q_t = S_i, O | \lambda)}{\sum_{i=1}^{N} P(q_t = S_i, O | \lambda)} = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{i=1}^{N} \alpha_t(i)\beta_t(i)}}$$

with

$$\sum_{i=1}^{N} \gamma_t(i) = 1, \ \forall t$$

then

$$\boxed{q_t^* = \underset{1 \le i \le N}{\operatorname{argmax}} \left[\gamma_t(i)\right], \ \ 1 \le t \le T}$$

Problem: q_t^* need not obey state transition constraints.

Best State Sequence—The Viterbi Algorithm

Define $\delta_t(i)$ as the highest probability along a single path, at time *t*, which accounts for the first *t* observations, i.e.,

$$\boxed{\delta_t(i) = \max_{q_1,q_2,\ldots,q_{t-1}} P\left[q_1 q_2 \ldots q_{t-1}, q_t = i, O_1 O_2 \ldots O_t \mid \lambda\right]}$$

We must keep track of the state sequence which gave the best path, at time *t*, to state *i*. We do this in the array $\psi_t(i)$.

The Viterbi Algorithm

Step 1 - -Initialization

$$\delta_1(i) = \pi_i \, b_i(O_1), \quad 1 \le i \le N$$

$$\psi_1(i) = 0, \quad 1 \le i \le N$$

Step 2 - -Recursion

$$\delta_t(j) = \max_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right] b_j\left(O_t\right), \quad 2 \le t \le T,\ 1 \le j \le N$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N}\left[\delta_{t-1}(i) a_{ij}\right], \quad 2 \le t \le T,\ 1 \le j \le N$$

Step 3 - -Termination

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

$$q_T^* = \operatorname*{argmax}_{1 \le i \le N}\left[\delta_T(i)\right]$$

Step 4 - -Path (State Sequence) Backtracking

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

Calculation $\approx N^2 T$ operations $(*,+)$

Alternative Viterbi Implementation

$$\tilde{\pi}_i = \log\left(\pi_i\right) \qquad 1 \le i \le N$$

$$\tilde{b}_i\left(O_t\right) = \log\left[b_i\left(O_t\right)\right] \qquad 1 \le i \le N, 1 \le t \le T$$

$$\tilde{a}_{ij} = \log\left[a_{ij}\right] \qquad 1 \le i, j \le N$$

Step 1 - -Initialization

$$\tilde{\delta}_1(i) = \log(\delta_1(i)) = \tilde{\pi}_i + \tilde{b}_i\left(O_1\right), \quad 1 \le i \le N$$

$$\psi_1(i) = 0, \qquad 1 \le i \le N$$

Step 2 - -Recursion

$$\tilde{\delta}_t(j) = \log(\delta_t(\text{j}))\text{=}\max_{1 \le i \le N}\left[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}\right] + \tilde{b}_j\left(O_t\right), \quad 2 \le t \le T,\ 1 \le j \le N$$

$$\psi_t(j) = \operatorname*{argmax}_{1 \le i \le N}\left[\tilde{\delta}_{t-1}(i) + \tilde{a}_{ij}\right], \qquad 2 \le t \le T,\ 1 \le j \le N$$

Step 3 - -Termination

$$\tilde{P}^* = \max_{1 \le i \le N}\left[\tilde{\delta}_T(i)\right], \qquad 1 \le i \le N$$

$$q_T^* = \operatorname*{argmax}_{1 \le i \le N}\left[\tilde{\delta}_T(i)\right], \quad 1 \le i \le N$$

Step 4 - -Backtracking

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

Calculation $\approx N^2 T$ additions

Problem

Given the model of the coin toss experiment used earlier (i.e., 3 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

with all state transition probabilities equal to 1/3, and with initial state probabilities equal to 1/3. For the observation sequence O=H H H H T H T T T T, find the Viterbi path of maximum likelihood.

Problem Solution

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 $\left(2 \le t \le 10\right)$

$$\delta_2(1) = (0.75)(0.5), \quad \delta_2(2) = (0.75)^2, \quad \delta_2(3) = (0.75)(0.25)$$

$$\delta_3(1) = (0.75)^2(0.5), \quad \delta_3(2) = (0.75)^3, \quad \delta_3(3) = (0.75)^2(0.25)$$

$$\delta_4(1) = (0.75)^3(0.5), \quad \delta_4(2) = (0.75)^4, \quad \delta_4(3) = (0.75)^3(0.25)$$

$$\delta_5(1) = (0.75)^4(0.5), \quad \delta_5(2) = (0.75)^4(0.25), \quad \delta_5(3) = (0.75)^5$$

$$\delta_6(1) = (0.75)^5(0.5), \quad \delta_6(2) = (0.75)^6, \quad \delta_6(3) = (0.75)^5(0.25)$$

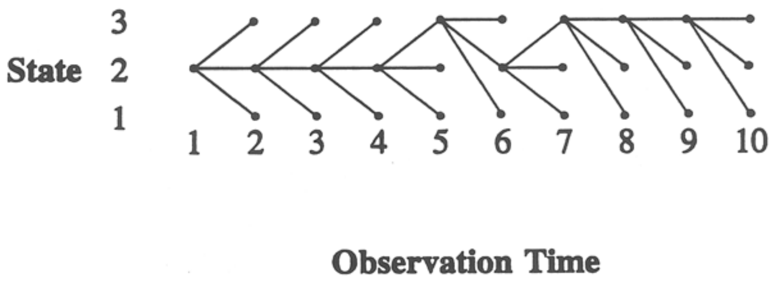
$$\delta_7(1) = (0.75)^6(0.5), \quad \delta_7(2) = (0.75)^6(0.25), \quad \delta_7(3) = (0.75)^7$$

$$\delta_8(1) = (0.75)^7(0.5), \quad \delta_8(2) = (0.75)^7(0.25), \quad \delta_8(3) = (0.75)^8$$

$$\delta_9(1) = (0.75)^8(0.5), \quad \delta_9(2) = (0.75)^8(0.25), \quad \delta_9(3) = (0.75)^9$$

$$\delta_{10}(1) = (0.75)^9(0.5), \quad \delta_{10}(2) = (0.75)^9(0.25), \quad \delta_{10}(3) = (0.75)^{10}$$

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



Solution to Problem 3—the Training Problem

- no globally optimum solution is known
- all solutions yield local optima
  - can get solution via gradient techniques
  - can use a re-estimation procedure such as the Baum-Welch or EM method
- consider re-estimation procedures
  - basic idea: given a current model estimate, λ, compute expected values of model events, then refine the model based on the computed values

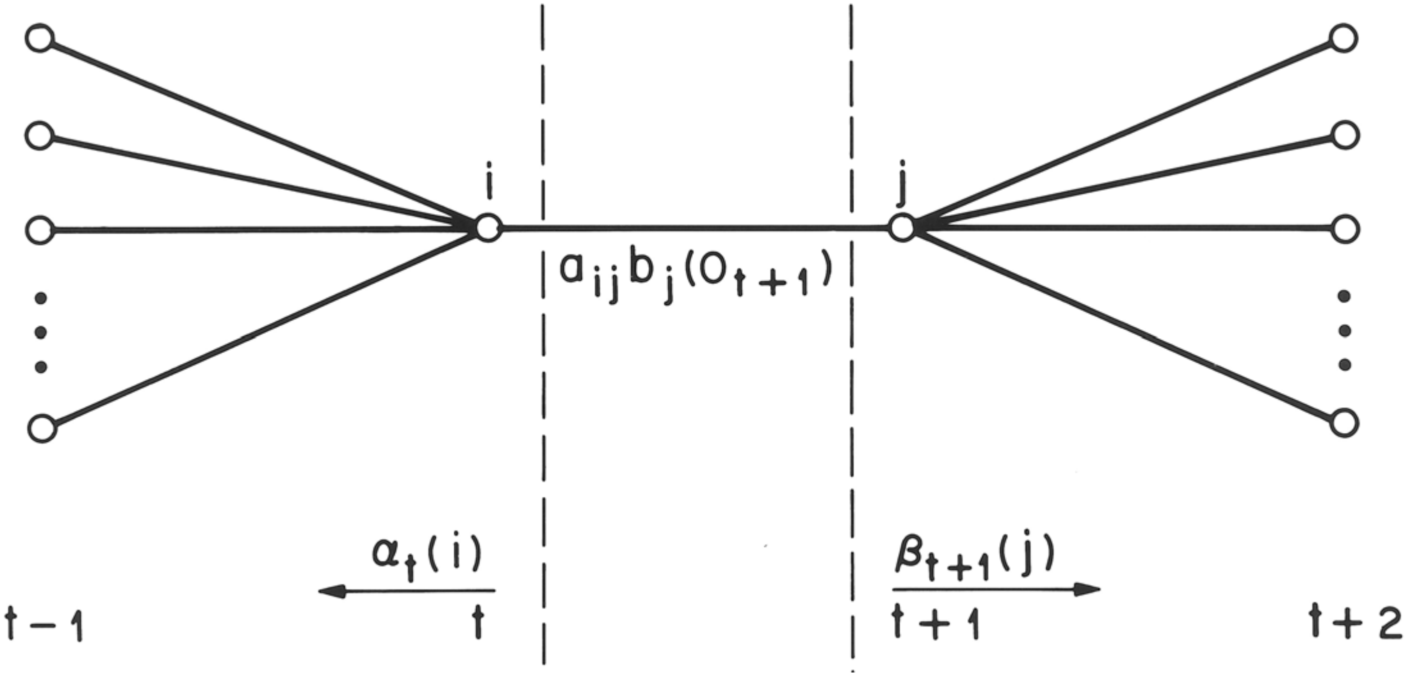
$$\lambda^{(0)} \xrightarrow{E[\text{Model Events}]} \lambda^{(1)} \xrightarrow{E[\text{Model Events}]} \lambda^{(2)} \cdots$$

Define $\xi_t(i,j)$, the probability of being in state S_i at time t, and state S_j at time $t+1$, given the model and the observation sequence, i.e.,

$$\boxed{\xi_t(i,j) = P\left[q_t = S_i, q_{t+1} = S_j | O, \lambda\right]}$$

The Training Problem

$$\boxed{\xi_t(i,j) = P\left[q_t = S_i,\, q_{t+1} = S_j \,|\, O, \lambda\right]}$$



The Training Problem

$$\boxed{\xi_t(i,j) = P\left[q_t = S_i,\, q_{t+1} = S_j \,|O,\lambda\right]}$$

$$\xi_t(i,j) = \frac{P\left[q_t = S_i,\, q_{t+1} = S_j\,, O \,|\, \lambda\right]}{P(O|\lambda)}$$

$$= \frac{\alpha_t(i) a_{ij}\, b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{ij}\, b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij}\, b_j(O_{t+1}) \beta_{t+1}(j)}$$

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

$$\sum_{t=1}^{T-1} \gamma_t(i) = \text{Expected number of transitions from } S_i$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = \text{Expected number of transitions from } S_i \text{ to } S_j$$

Re-estimation Formulas

$$\bar{\pi}_i = \text{Expected number of times in state } S_i \text{ at } t = 1$$

$$= \gamma_1(i)$$

$$\bar{a}_{ij} = \frac{\text{Expected number of transitions from state } S_i \text{ to state } S_j}{\text{Expected number of transitions from state } S_i}$$

$$= \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T} \gamma_t(i)}$$

$$\bar{b}_j(k) = \frac{\text{Expected number of times in state } j \text{ with symbol } v_k}{\text{Expected number of times in state } j}$$

$$= \frac{\sum_{\substack{t=1 \\ \ni O_t = v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)}$$

Re-estimation Formulas

If $\lambda = (A, B, \Pi)$ is the initial model, and $\bar{\lambda} = (\bar{A}, \bar{B}, \bar{\Pi})$ is the re-estimated model, then it can be proven that either:

1. the initial model, λ, defines a critical point of the likelihood function, in which case $\bar{\lambda} = \lambda$, or
2. model $\bar{\lambda}$ is more likely than model λ in the sense that $P(O|\bar{\lambda}) > P(O|\lambda)$, i.e., we have found a new model $\bar{\lambda}$ from which the observation sequence is more likely to have been produced.

Conclusion: Iteratively use $\bar{\lambda}$ in place of λ, and repeat the re-estimation until some limiting point is reached. The resulting model is called the maximum likelihood (ML) HMM.

Re-estimation Formulas

1. The re-estimation formulas can be derived by maximizing the auxiliary function $Q(\lambda,\bar{\lambda})$ over $\bar{\lambda}$, i.e.,

$$Q(\lambda,\bar{\lambda}) = \sum_{q} P(O,q|\lambda)\log\left[P(O,q|\bar{\lambda})\right]$$

It can be proved that:

$$\max_{\bar{\lambda}}\left[Q(\lambda,\bar{\lambda})\right] \Rightarrow P(O|\bar{\lambda}) \geq P(O|\lambda)$$

Eventually the likelihood function converges to a critical point

2. Relation to EM algorithm:
   - E (Expectation) step is the calculation of the auxiliary function, $Q(\lambda,\bar{\lambda})$
   - M (Modification) step is the maximization over $\bar{\lambda}$

Notes on Re-estimation

1. Stochastic constraints on $\pi_i, a_{ij}, b_j(k)$ are automatically met, i.e.,

$$\sum_{i=1}^{N} \overline{\pi}_i = 1, \quad \sum_{j=1}^{N} \overline{a}_{ij} = 1, \quad \sum_{k=1}^{M} \overline{b}_j(k) = 1$$

2. At the critical points of $P = P(O|\lambda)$, then

$$\pi_\mathrm{i} = \frac{\pi_\mathrm{i} \dfrac{\partial P}{\partial \pi_i}}{\displaystyle\sum_{k=1}^{N} \pi_k \frac{\partial P}{\partial \pi_k}} = \overline{\pi}_i$$

$$a_{ij} = \frac{a_{ij} \dfrac{\partial P}{\partial a_{ij}}}{\displaystyle\sum_{k=1}^{N} a_{ik} \frac{\partial P}{\partial a_{ik}}} = \overline{a}_{ij}$$

$$b_j(k) = \frac{b_j(k) \dfrac{\partial P}{\partial b_j(k)}}{\displaystyle\sum_{\ell=1}^{M} b_j(l) \frac{\partial P}{\partial b_j(\ell)}} = \overline{b}_j(k)$$

$\Rightarrow$ at critical points, the re-estimation formulas are *exactly* correct.

Variations on HMM's

1. Types of HMM—model structures
2. Continuous observation density models—mixtures
3. Autoregressive HMM's—LPC links
4. Null transitions and tied states
5. Inclusion of explicit state duration density in HMM's
6. Optimization criterion—ML, MMI, MDI

Types of HMM

1. Ergodic models--no transient states

2. Left-right models--all transient states (except the last state) with the constraints:

$$\pi_i = \begin{cases} 1, & i = 1 \\ 0, & i \neq 1 \end{cases}$$

$$a_{ij} = 0 \qquad j > i$$

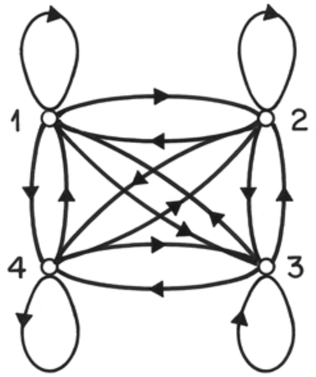
Controlled transitions implies:

$$a_{ij} = 0, \quad j > i + \Delta \ (\Delta = 1,2 \text{ typically})$$

3. Mixed forms of ergodic and left-right models (e.g., parallel branches)

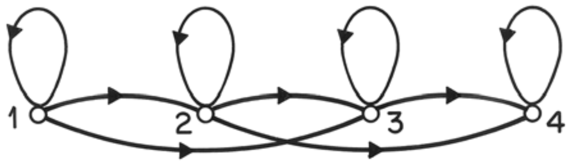
Note: Constraints of left-right models don't affect re-estimation formulas (i.e., a parameter initially set to 0 remains at 0 during re-estimation).

Types of HMM



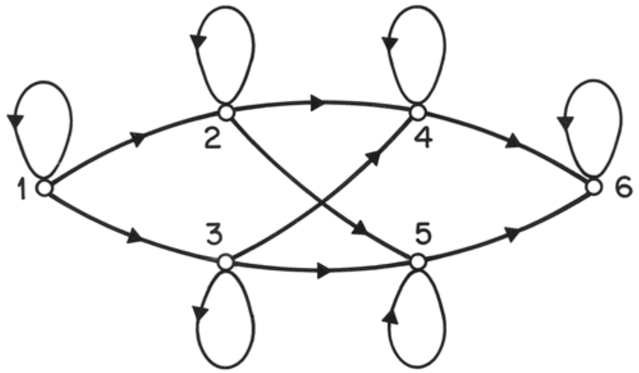
(a)

Ergodic Model



(b)

Left-Right Model



(c)

Mixed Model

Continuous Observation Density HMM's

Most general form of pdf with a valid re-estimation procedure is:

$$\boxed{b_j(x) = \sum_{m=1}^{M} c_{jm} \mathbb{N}\left[x, \mu_{jm}, U_{jm}\right], \quad 1 \le j \le N}$$

$x =$ observation vector$=\left\{x_1, x_2, ..., x_D\right\}$

$M =$ number of mixture densities

$c_{jm} =$ gain of *m*-th mixture in state *j*

$\mathbb{N} =$ any log-concave or elliptically symmetric density (e.g., a Gaussian)

$\mu_{jm} =$ mean vector for mixture *m*, state *j*

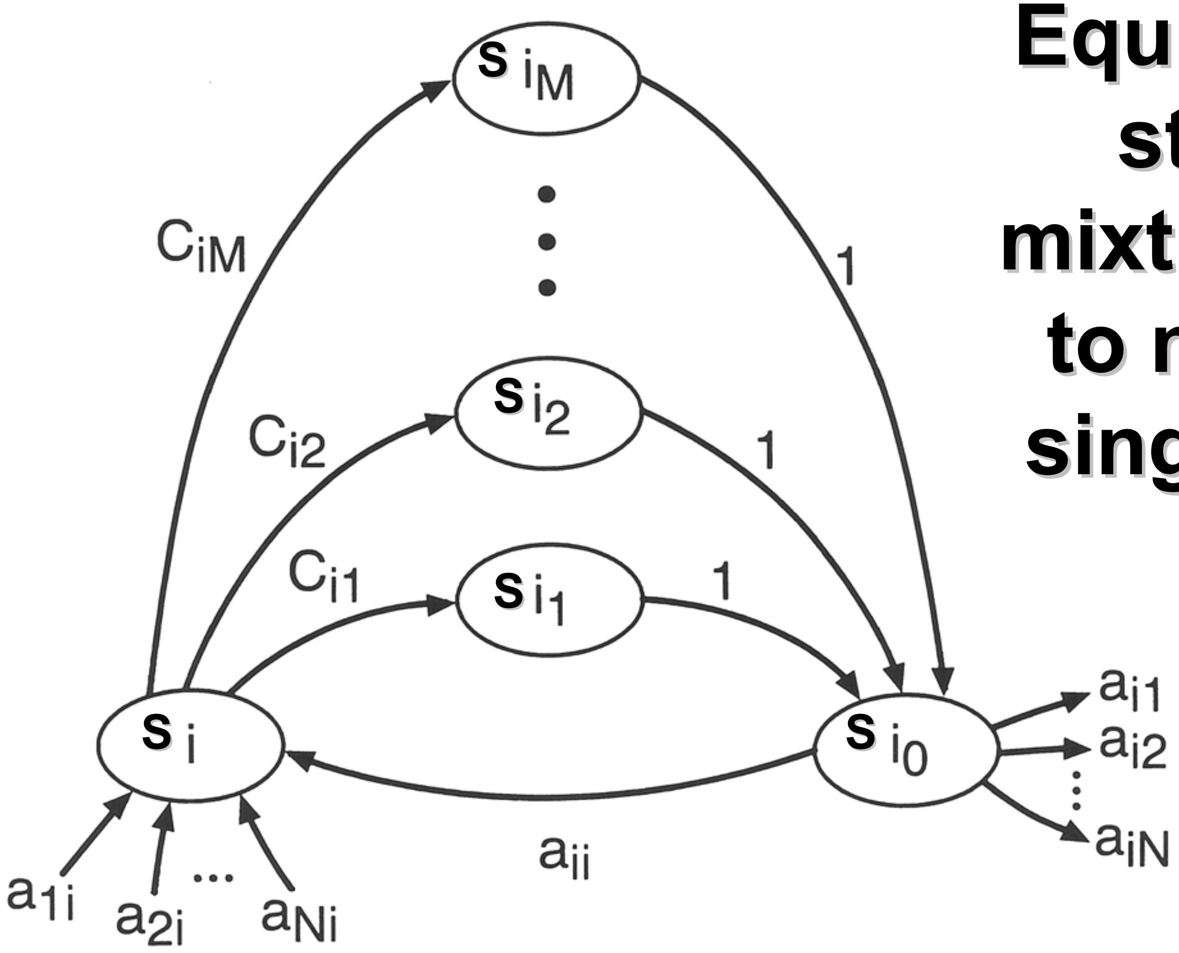
$U_{jm} =$ covariance matrix for mixture *m*, state *j*

$$c_{jm} \ge 0, \qquad 1 \le j \le N, \quad 1 \le m \le M$$

$$\sum_{m=1}^{M} c_{jm} = 1, \qquad 1 \le j \le N$$

$$\int_{-\infty}^{\infty} b_j(x) dx = 1, \quad 1 \le j \le N$$

State Equivalence Chart



Equivalence of state with mixture density to multi-state single mixture case

Re-estimation for Mixture Densities

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

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

$$\bar{U}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot \left(O_t - \mu_{jk}\right)\left(O_t - \mu_{jk}\right)'}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

- $\gamma_t(j,k)$ is the probability of being in state j at time t with the k-th mixture component accounting for O_t

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)}\right]\left[\frac{c_{jk}\mathbb{N}(O_t, \mu_{jk}, U_{jk})}{\sum_{m=1}^{M} c_{jm}\mathbb{N}(O_t, \mu_{jm}, U_{jm})}\right]$$

Autoregressive HMM

Consider an observation vector $O = (x_0, x_1, ..., x_{K-1})$ where each x_k is a waveform sample, and O represents a frame of the signal (e.g., $K = 256$ samples). We assume x_k is related to previous samples of O by a Gaussian autoregressive process of order p, i.e.,

$$O_k = -\sum_{i=1}^{p} a_i O_{k-i} + e_k, \quad 0 \le k \le K-1$$

where e_k are Gaussian, independent, identically distributed random variables with zero mean and variance σ^2, and $a_i, 1 \le i \le p$ are the autoregressive or predictor coefficients.

As $K \to \infty$, then

$$f(O) = (2\pi\sigma^2)^{-K/2} \exp\left\{ -\frac{1}{2\sigma^2} \delta(O,a) \right\}$$

where

$$\delta(O,a) = r_a(0) r(0) + 2\sum_{i=1}^{p} r_a(i) r(i)$$

Autoregressive HMM

$$r_a(i) = \sum_{n=0}^{p-i} a_n a_{n+i}, \ (a_0 = 1), \ 1 \le i \le p$$

$$r(i) = \sum_{n=0}^{K-i-1} x_n x_{n+i}, \ 0 \le i \le p$$

$$[a]' = [1, a_1, a_2, \ldots, a_p]$$

The prediction residual is:

$$\alpha = E\left[\sum_{i=1}^{K} (e_i)^2\right] = K\sigma^2$$

Consider the normalized observation vector

$$\hat{O} = \frac{O}{\sqrt{\alpha}} = \frac{O}{\sqrt{K\sigma^2}}$$

$$f(\hat{O}) = (2\pi)^{-K/2} \exp\left(-\frac{K}{2}\delta(\hat{O}, a)\right)$$

In practice, K is replaced by $\hat{K}$, the effective frame length, e.g.,

$\hat{K} = K/3$ for frame overlap of 3 to 1.

Application of Autoregressive HMM

$$b_j(0) = \sum_{m=1}^{M} c_{jm} b_{jm}(O)$$

$$b_{jm}(O) = (2\pi)^{-K/2} \exp\left\{ -\frac{K}{2} \delta(O, a_{jm}) \right\}$$

Each mixture characterized by predictor vector or by autocorrelation vector from which predictor vector can be derived. Re-estimation formulas for r_{jk} are:

$$\overline{r}_{jk} = \frac{\sum_{t=1}^{T} \gamma_t(j,k) \cdot r_t}{\sum_{t=1}^{T} \gamma_t(j,k)}$$

$$\gamma_t(j,k) = \left[\frac{\alpha_t(j)\beta_t(j)}{\sum_{j=1}^{N} \alpha_t(j)\beta_t(j)} \right] \left[\frac{c_{jk} b_{jk}(O_t)}{\sum_{k=1}^{M} c_{jk} b_{jk}(O_t)} \right]$$

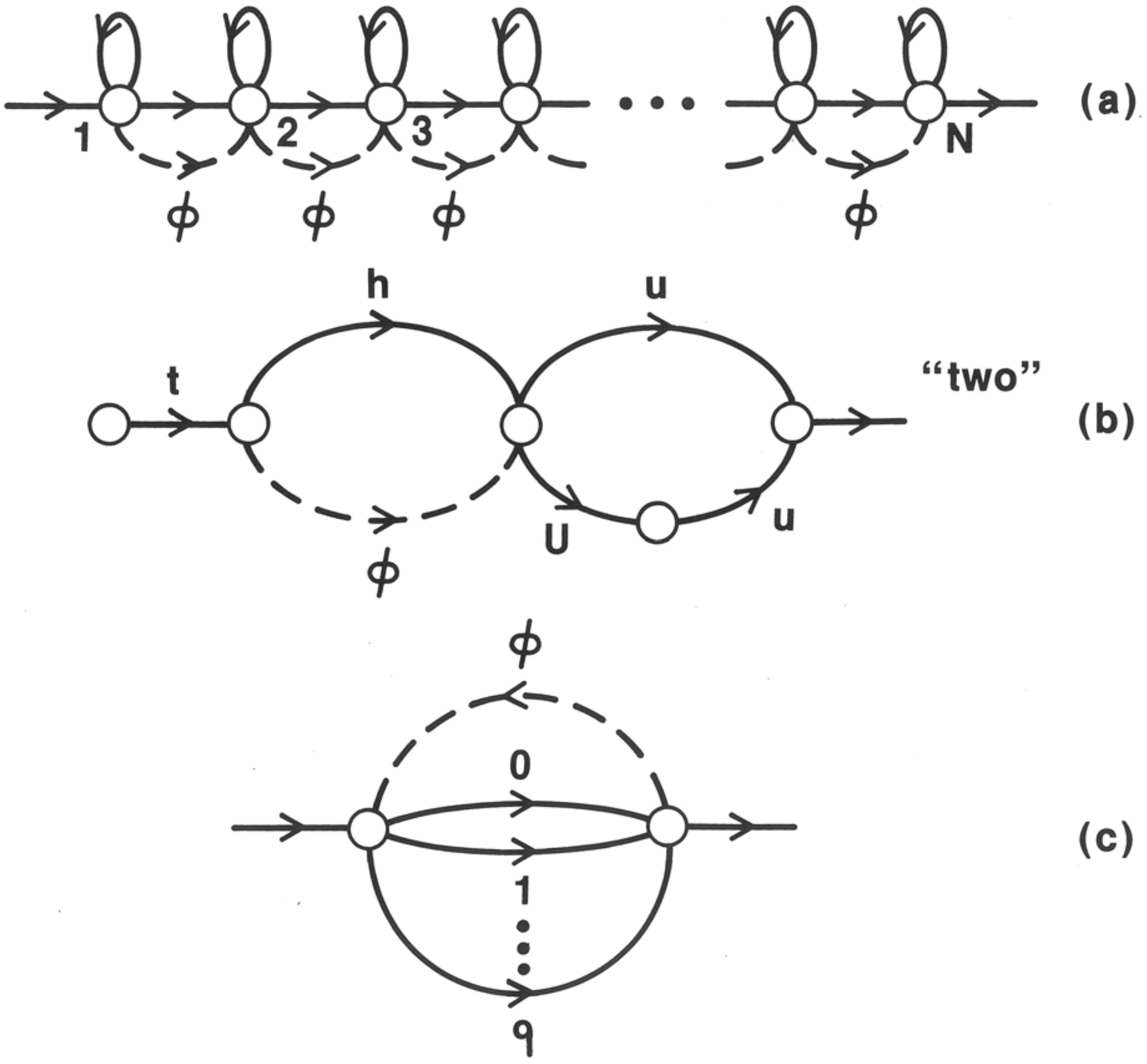
Null Transitions and Tied States

Null Transitions: transitions which produce no output, and take no time, denoted by φ

Tied States: sets up an equivalence relation between HMM parameters in different states

- number of independent parameters of the model reduced
- parameter estimation becomes simpler
- useful in cases where there is insufficient training data for reliable estimation of all model parameters

Null Transitions



Inclusion of Explicit State Duration Density

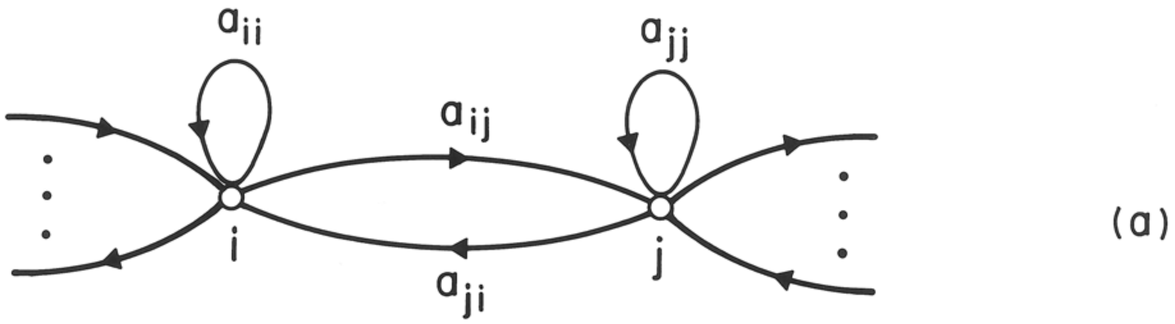
For standard HMM's, the duration density is:

$$p_i(d) = \text{probability of exactly } d \text{ observations in state } S_i$$
$$= (a_{ii})^{d-1}(1 - a_{ii})$$

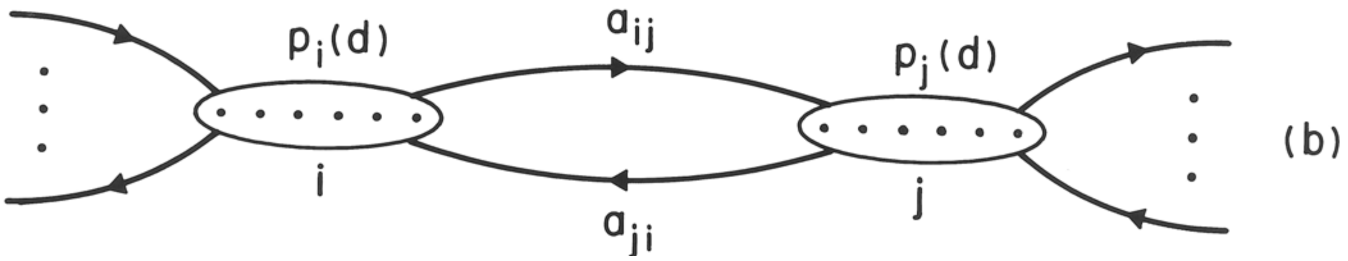
With arbitrary state duration density, $p_i(d)$, observations are generated as follows:

1. an initial state, $q_1 = S_i$, is chosen according to the initial state distribution, π_i
2. a duration d_1 is chosen according to the state duration density $p_{q_1}(d_1)$
3. observations $O_1 O_2 ... O_{d_1}$ are chosen according to the joint density $b_{q_1}(O_1 O_2 ... O_{d_1})$. Generally we assume independence, so
$$b_{q_1}(O_1 O_2 ... O_{d_1}) = \prod_{t=1}^{d_1} b_{q_1}(O_t)$$
4. the next state, $q_2 = S_j$, is chosen according to the state transition probabilities, $a_{q_1 q_2}$, with the constraint that $a_{q_1 q_1} = 0$, i.e., no transition back to the same state can occur.

Explicit State Duration Density



Standard HMM



HMM with explicit state duration density

Explicit State Duration Density

t	1	d_1+1	d_1+d_2+1
state	q_1	q_2	q_3
duration	d_1	d_2	d_3
observations	$O_1 \ldots O_{d_1}$	$O_{d_1+1} \ldots O_{d_1+d_2}$	$O_{d_1+d_2+1} \ldots O_{d_1+d_2+d_3}$

Assume:

1. first state, q_1, **begins** at $t=1$
2. last state, q_r, **ends** at $t=T$

$\Rightarrow$ entire duration intervals are included within the observation sequence $O_1 O_2 \ldots O_T$

Modified α:

$$\boxed{\alpha_t(i) = P(O_1 O_2 \ldots O_t, S_i \text{ ending at } t \mid \lambda)}$$

Assume r states in first t observations, i.e.,

$$Q = \{q_1 q_2 \ldots q_r\} \text{ with } q_r = S_i$$

$$D = \{d_1 d_2 \ldots d_r\} \text{ with } \sum_{s=1}^{r} d_s = t$$

Explicit State Duration Density

Then we have

$$\alpha_t(i)=\sum_q\sum_d \pi_{q_1}p_{q_1}(d_1)P(O_1O_2...O_{d_1}|q_1)$$
$$\cdot a_{q_1q_2}p_{q_2}(d_2)P(O_{d_1+1}...O_{d_1+d_2}|q_2)...$$
$$\cdot a_{q_{r-1}q_r}p_{q_r}(d_r)P(O_{d_1+d_2+...+d_{r-1}+1}...O_t|q_r)$$

By induction:

$$\boxed{\alpha_t(j)=\sum_{i=1}^{N}\sum_{d=1}^{D}\alpha_{t-d}(i)a_{ij}\,p_j(d)\prod_{s=t-d+1}^{t}b_j(O_s)}$$

Initialization of $\alpha_t(i)$:

$$\alpha_1(i)=\pi_i p_i(1)b_i(O_1)$$

$$\alpha_2(i)=\pi_i p_i(2)\prod_{s=1}^{2}b_i(O_s)+\sum_{j=1,j\neq i}^{N}\alpha_1(j)a_{ji}\,p_i(1)b_i(O_2)$$

$$\alpha_3(i)=\pi_i p_i(3)\prod_{s=1}^{3}b_i(O_s)+\sum_{d=1}^{2}\sum_{j=1,j\neq i}^{N}\alpha_{3-d}(j)a_{ji}\,p_i(d)\prod_{s=4-d}^{3}b_i(O_s)$$

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

Explicit State Duration Density

- re-estimation formulas for a_{ij}, $b_i(k)$, and $p_i(d)$ can be formulated and appropriately interpreted
- modifications to Viterbi scoring required, i.e.,

$$\delta_t(i) = P(O_1O_2...O_t, q_1q_2...q_r = S_i \text{ ending at } t|O)$$

Basic Recursion :

$$\boxed{\delta_t(i) = \max_{1\le j\le N,\, j\ne i}\ \max_{1\le d\le D}\left[\delta_{t-d}(j)a_{ji}\,p_i(d)\prod_{s=t-d+1}^{t} b_j(O_s)\right]}$$

- storage required for $\delta_{t-1}...\delta_{t-D} \Rightarrow N\cdot D$ locations
- maximization involves all terms--not just old δ's and a_{ji} as in previous case $\Rightarrow$ significantly larger computational load $\approx (D^2/2)N^2T$ computations involving $b_j(O)$

Example: $N = 5$, $D = 20$

	implicit duration	explicit duration
storage	5	100
computation	2500	500,000

Issues with Explicit State Duration Density

1. quality of signal modeling is often improved significantly
2. significant increase in the number of parameters per state (D duration estimates)
3. significant increase in the computation associated with probability calculation ($\approx D^2/2$)
4. insufficient data to give good $p_i(d)$ estimates

Alternatives :

1. use parametric state duration density

$$p_i(d) = \mathbb{N}(d, \mu_i, \sigma_i^2) \text{ -- Gaussian}$$

$$p_i(d) = \frac{\eta_i^{\nu_i} d^{\nu_i - 1} e^{-\eta_i d}}{\Gamma(\nu_i)} \text{ -- Gamma}$$

2. incorporate state duration information after probability calculation, e.g., in a post-processor

Alternatives to ML Estimation

Assume we wish to design V different HMM's, $\lambda_1,\lambda_2,...,\lambda_V$.

Normally we design each HMM, λ_V, based on a training set of observations, O^V, using a maximum likelihood (ML) criterion, i.e.,

$$\boxed{P_V^* = \max_{\lambda_V} P\left[O^V \,|\, \lambda_V\right]}$$

Consider the **mutual information**, I_V, between the observation sequence, O^V, and the **complete** set of models $\lambda = \left(\lambda_1,\lambda_2,...,\lambda_V\right)$,

$$I_V = \left[\log P(O^V \,|\, \lambda_V) - \log \sum_{w=1}^{V} P(O^V \,|\, \lambda_W)\right]$$

Consider maximizing I_V over λ, giving

$$I_V^* = \max_{\lambda}\left[\log P(O^V \,|\, \lambda_V) - \log \sum_{w=1}^{V} P(O^V \,|\, \lambda_W)\right]$$

- choose λ so as to separate the correct model, λ_V, from all other models, as much as possible, for the training set, O^V.

Alternatives to ML Estimation

Sum over all such training sets to give models according to an MMI criterion, i.e.,

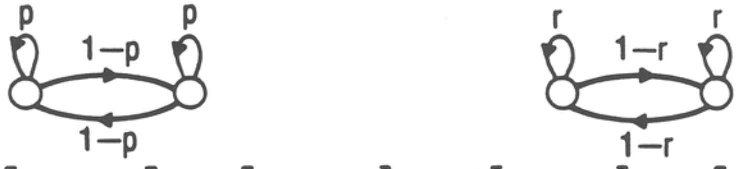
$$I^* = \max_{\lambda}\left\{\sum_{v=1}^{V}\left[\log\left(P(O^v \mid \lambda_v\right) - \log\sum_{w=1}^{V} P(O^v \mid \lambda_w)\right]\right\}$$

- solution via steepest descent methods.

Comparison of HMM's

Problem: given two HMM's, λ_1 and λ_2, is it possible to give a measure of how similar the two models are

Example :



$$A_1 = \begin{bmatrix} p & 1-p \\ 1-p & p \end{bmatrix}, B_1 = \begin{bmatrix} q & 1-q \\ 1-q & q \end{bmatrix} \quad A_2 = \begin{bmatrix} r & 1-r \\ 1-r & r \end{bmatrix}, B_2 = \begin{bmatrix} s & 1-s \\ 1-s & s \end{bmatrix}$$

For $(A_1, B_1) \overset{equivalent}{\Leftrightarrow} (A_2, B_2)$ we require $P(O_t = \nu_k)$ to be the same for both models and for all symbols ν_k. Thus we require

$$pq + (1-p)(1-q) = rs + (1-r)(1-s)$$

$$2pq - p - q = 2rs = r = s$$

$$s = \frac{p + 1 - 2pq - r}{1 - 2r}$$

Let $\quad p = 0.6,\ q = 0.7,\ r = 0.2$, then

$$s = 13/30 \simeq 0.433$$

Comparison of HMM's

Thus the two models have very different *A* and *B* matrices, but are equivalent in the sense that all symbol probabilities (averaged over time) are the same.

We generalize the concept of model distance (dis-similarity) by defining a distance measure, $D(\lambda_1, \lambda_2)$ between two Markov sources, λ_1 and λ_2, as

$$\boxed{D(\lambda_1, \lambda_2) = \frac{1}{T}\left[\log P(O_T^{(2)} | \lambda_1) - \log P(O_T^{(2)} | \lambda_2)\right]}$$

where $O_T^{(2)}$ is a sequence of observations generated by model λ_2, and scored by **both** models.

We symmetrize *D* by using the relation:

$$D_S(\lambda_1, \lambda_2) = \frac{1}{2}\left[D(\lambda_1, \lambda_2) + D(\lambda_2, \lambda_1)\right]$$

Implementation Issues for HMM's

1. Scaling—to prevent underflow and/or overflow.
2. Multiple Observation Sequences—to train left-right models.
3. Initial Estimates of HMM Parameters—to provide robust models.
4. Effects of Insufficient Training Data

Scaling

- $\alpha_t(i)$ is a sum of a large number of terms, each of the form:

$$\left[\prod_{s=1}^{t\text{-}1} a_{q_s q_{s+1}} \prod_{s=1}^{t} b_{q_s}(O_s)\right]$$

- since each *a* and *b* term is less than 1, as *t* gets larger, $\alpha_t(i)$ exponentially heads to 0. Thus scaling is required to prevent underflow.
- consider scaling $\alpha_t(i)$ by the factor

$$c_t = \frac{1}{\sum_{i=1}^{N} \alpha_t(i)}, \quad \text{independent of } t$$

- we denote the scaled α's as:

$$\hat{\alpha}_t(i) = c_t \alpha_t(i) = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)}$$

$$\sum_{i=1}^{N} \hat{\alpha}_t(i) = 1$$

Scaling

- for fixed *t*, we compute

$$\alpha_t(i) = \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji}\, b_i(O_t)$$

- scaling gives

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji}\, b_i(O_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \hat{\alpha}_{t-1}(j) a_{ji}\, b_i(O_t)}$$

- by induction we get

$$\hat{\alpha}_{t-1}(j) = \left[\prod_{\tau=1}^{t-1} c_\tau \right] \alpha_{t-1}(j)$$

- giving

$$\hat{\alpha}_t(i) = \frac{\sum_{j=1}^{N} \alpha_{t-1}(j) \left[\prod_{\tau=1}^{t-1} c_\tau \right] a_{ji}\, b_i(O_t)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_{t-1}(j) \left[\prod_{\tau=1}^{t-1} c_\tau \right] a_{ji}\, b_i(O_t)} = \frac{\alpha_t(i)}{\sum_{i=1}^{N} \alpha_t(i)}$$

Scaling

- for scaling the $\beta_t(i)$ terms we use the **same** scale factors as for the $\alpha_t(i)$ terms, i.e.,

$$\hat{\beta}_t(i) = c_t \beta_t(i)$$

  since the magnitudes of the α and β terms are comparable.

- the re-estimation formula for a_{ij} in terms of the scaled α's and β's is:

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T-1} \hat{\alpha}_t(i) a_{ij} b_j(O_{t+1}) \hat{\beta}_{t+1}(j)}$$

- we have

$$\hat{\alpha}_t(i) = \left[\prod_{\tau=1}^{t} c_\tau \right] \alpha_t(i) = C_t \alpha_t(i)$$

$$\hat{\beta}_{t+1}(j) = \left[\prod_{\tau=t+1}^{T} c_\tau \right] \beta_{t+1}(j) = D_{t+1} \beta_{t+1}(j)$$

Scaling

- giving

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)}{\sum_{j=1}^{N} \sum_{t=1}^{T-1} C_t \alpha_t(i) a_{ij} b_j(O_{t+1}) D_{t+1} \beta_{t+1}(j)}$$

$$C_t D_{t+1} = \prod_{\tau=1}^{t} c_\tau \prod_{\tau=t+1}^{T} c_\tau = \prod_{\tau=1}^{T} c_\tau = C$$

- independent of t.

Notes on Scaling :

1. scaling procedure works equally well on π or B coefficients

2. scaling need not be performed each iteration; set $c_t = 1$ whenever scaling is skipped

c. can solve for $P(O|\lambda)$ from scaled coefficients as:

$$\prod_{t=1}^{T} c_t \sum_{i=1}^{N} \alpha_T(i) = C \sum_{i=1}^{N} \alpha_T(i) = 1$$

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) = 1 / \prod_{t=1}^{T} c_t$$

$$\log P(O|\lambda) = -\sum_{t=1}^{T} \log(c_t)$$

Multiple Observation Sequences

For left-right models, we need to use multiple sequences of observations for training.

Assume a set of K observation sequences (i.e., training utterances):

$$O = \left[O^{(1)}, O^{(2)}, \ldots, O^{(K)}\right]$$

where

$$O^{(k)} = \left[O_1^{(k)} O_2^{(k)} \ldots O_{T_k}^{(k)}\right]$$

We wish to maximize the probability

$$P(O|\lambda) = \prod_{k=1}^{K} P(O^{(k)}|\lambda) = \prod_{k=1}^{K} P_k$$

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \beta_{t+1}^k(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k - 1} \alpha_t^k(i) \beta_t^k(i)}$$

Scaling requires:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \hat{\alpha}_t^k(i) a_{ij} b_j(O_{t+1}^{(k)}) \hat{\beta}_{t+1}^k(j)}{\sum_{k=1}^{K} \frac{1}{P_k} \sum_{t=1}^{T_k - 1} \hat{\alpha}_t^k(i) \hat{\beta}_t^k(i)}$$

- all scaling factors cancel out

Initial Estimates of HMM Parameters

N -- choose based on physical considerations

M -- choose based on model fits

π_i -- random or uniform ($\pi_i \neq 0$)

a_{ij} -- random or uniform ($a_{ij} \neq 0$)

$b_j(k)$ -- random or uniform ($b_j(k) \geq \varepsilon$)

$b_j(O)$ -- need good initial estimates of mean vectors;

need reasonable estimates of covariance matrices

Effects of Insufficient Training Data

Insufficient training data leads to poor estimates of model parameters.

Possible Solutions:

1. use more training data--often this is impractical
2. reduce the size of the model--often there are physical reasons for keeping a chosen model size
3. add extra constraints to model parameters

$$b_j(k) \geq \varepsilon$$

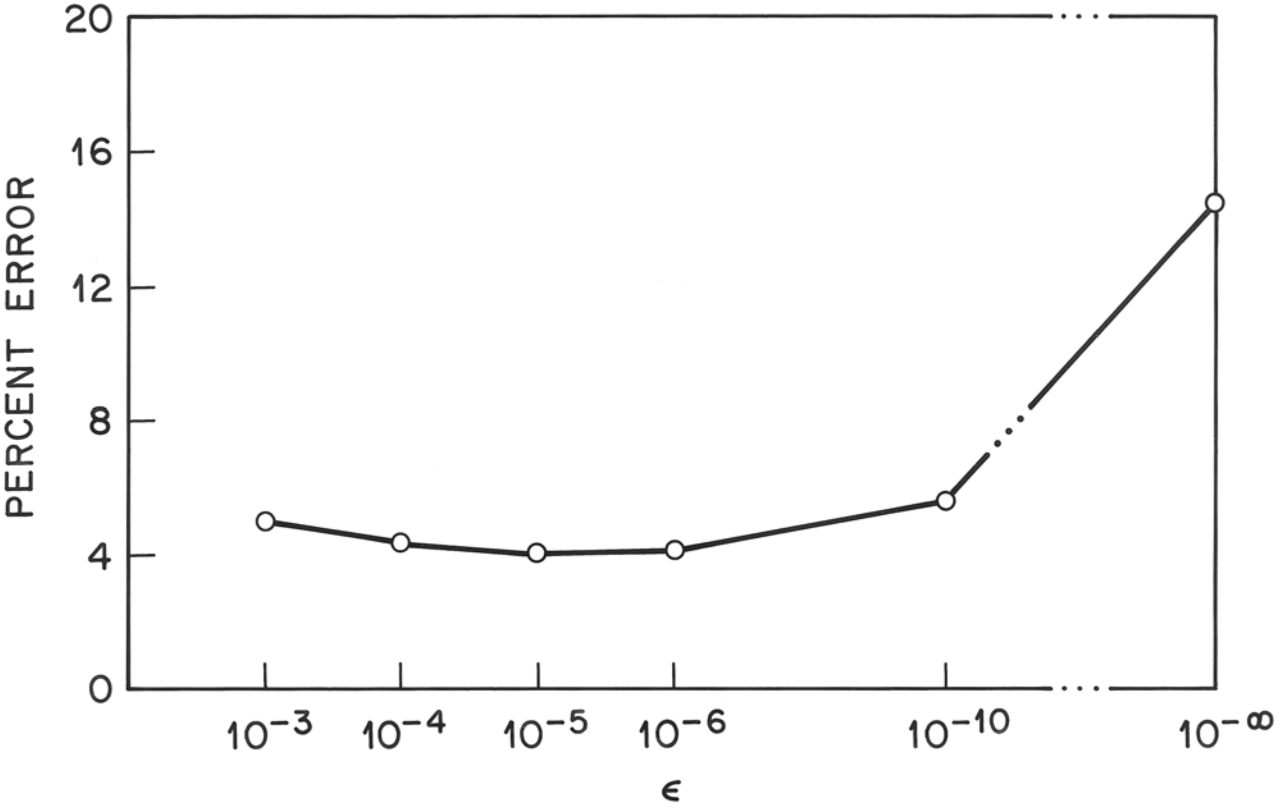
$$U_{jk}(r,r) \geq \delta$$

- often the model performance is relatively insensitive to exact choice of ε, δ

4. method of deleted interpolation

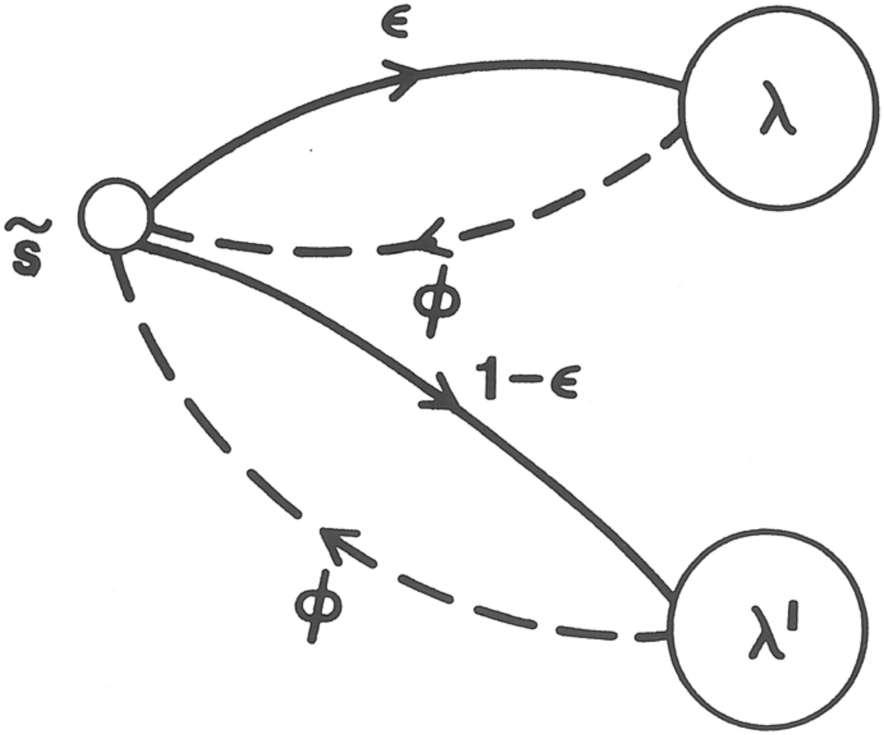
$$\overline{\lambda} = \varepsilon\lambda + (1-\varepsilon)\lambda'$$

Methods for Insufficient Data



Performance insensitivity to ε

Deleted Interpolation



Isolated Word Recognition Using HMM's

Assume a vocabulary of V words, with K occurrences of each spoken word in a training set. Observation vectors are spectral characterizations of the word. For isolated word recognition, we do the following:

1. for each word, v, in the vocabulary, we must build an HMM, λ^v, i.e., we must re-estimate model parameters $\left(A,B,\Pi\right)$ that optimize the likelihood of the training set observation vectors for the v-th word. (TRAINING)
2. for each unknown word which is to be recognized, we do the following:
   a. measure the observation sequence $O=\left[O_1 O_2 \ldots O_T\right]$

   b. calculate model likelihoods, $P(O|\lambda^v), 1 \leq v \leq V$

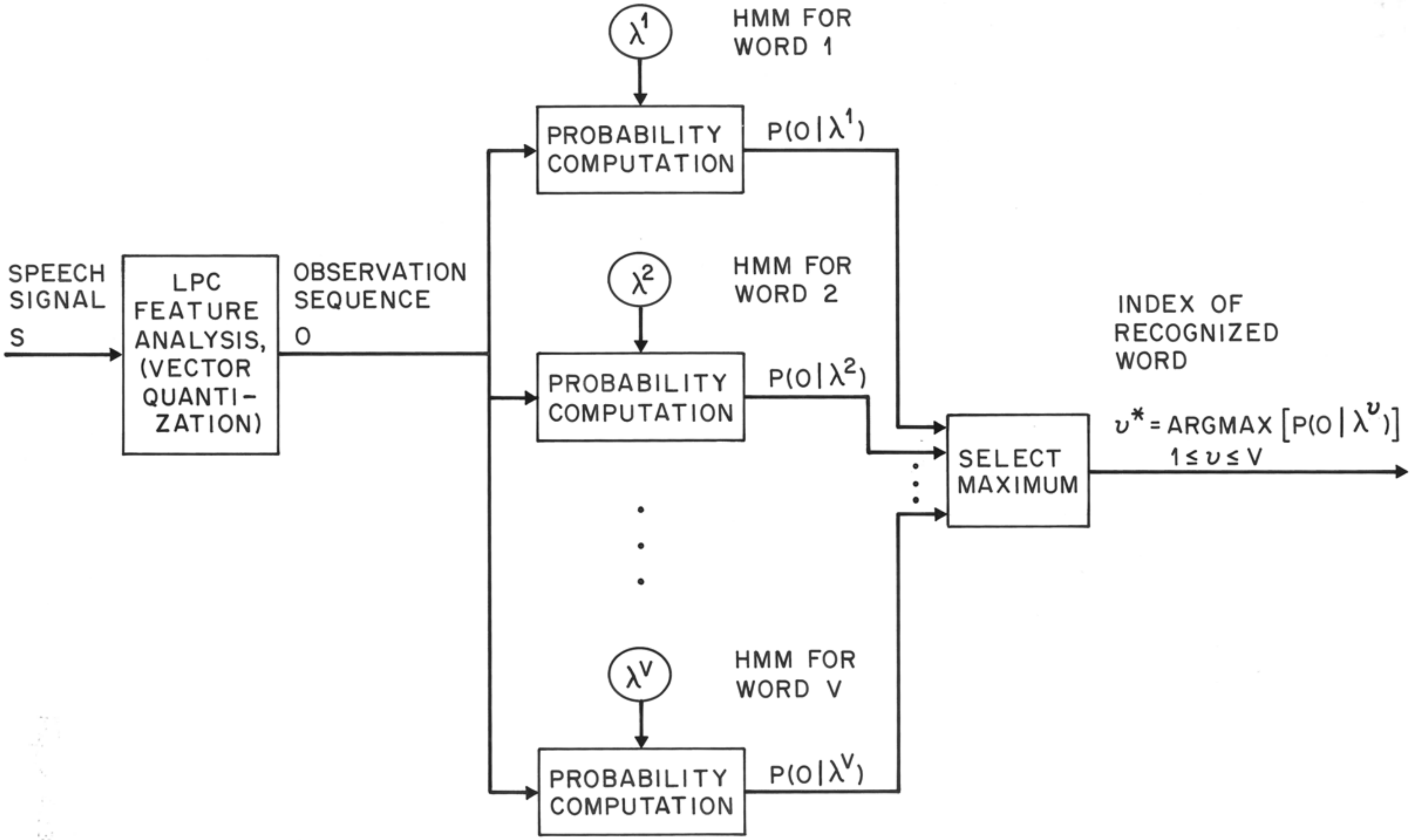
   c. select the word whose model likelihood score is highest

$$v^* = \underset{1\leq v\leq V}{\operatorname{argmax}}\left[P(O|\lambda^v)\right]$$

Computation is on the order of $V \cdot N^2 T$ required; $V=100, N=5, T=40$

$\Rightarrow 10^5$ computations

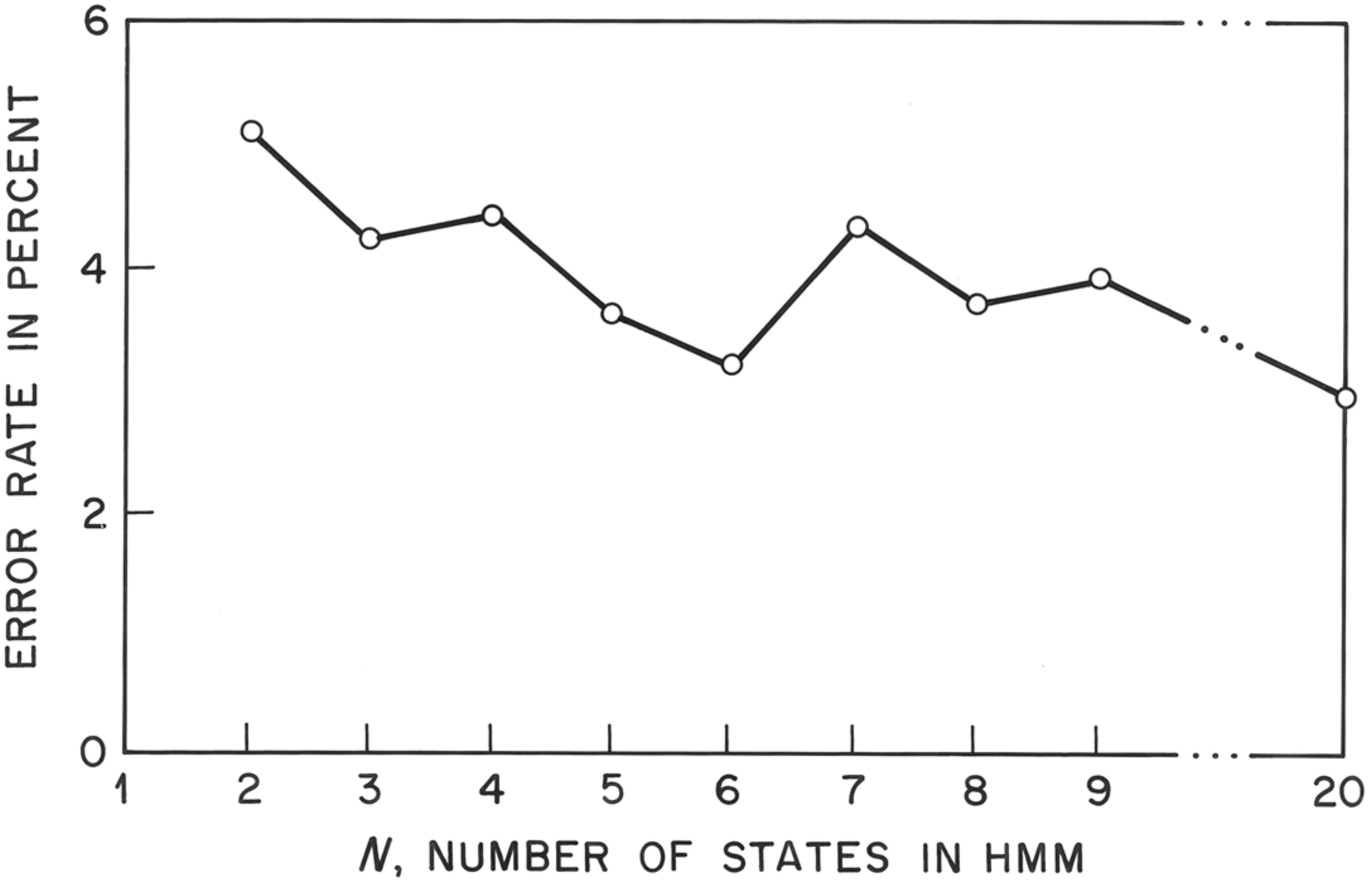
Isolated Word HMM Recognizer



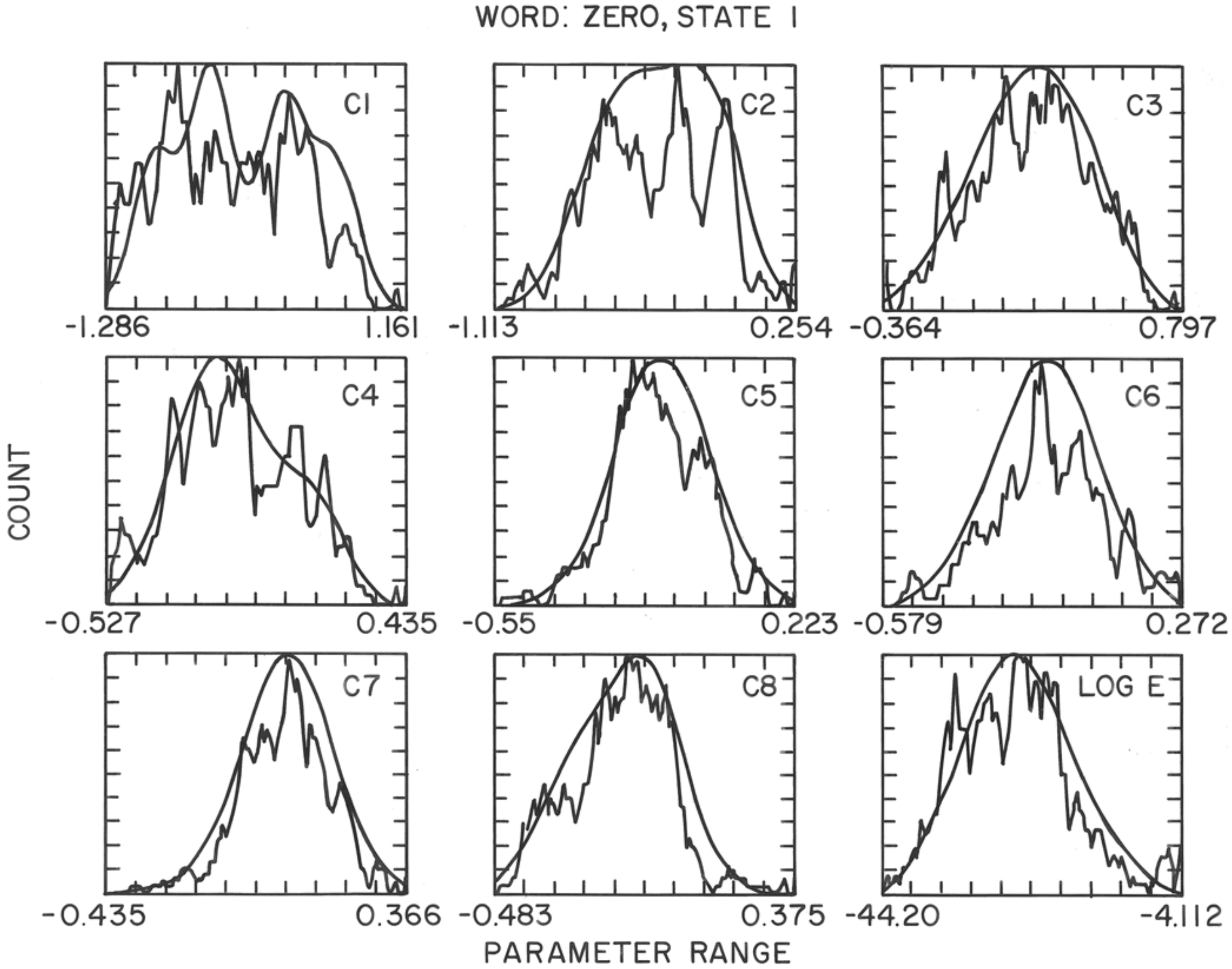
Choice of Model Parameters

1. Left-right model preferable to ergodic model (speech is a left-right process)
2. Number of states in range 2-40 (from sounds to frames)
    - Order of number of distinct sounds in the word
    - Order of average number of observations in word
3. Observation vectors
    - Cepstral coefficients (and their second and third order derivatives) derived from LPC (1-9 mixtures), diagonal covariance matrices
    - Vector quantized discrete symbols (16-256 codebook sizes)
4. Constraints on $b_j(O)$ densities
    - $bj(k)>\varepsilon$ for discrete densities
    - $C_{jm}>\delta$, $U_{jm}(r,r)>\delta$ for continuous densities

Performance Vs Number of States in Model



HMM Feature Vector Densities



Segmental K-Means Segmentation into States

Motivation:

derive good estimates of the $b_j(O)$ densities as required for rapid convergence of re-estimation procedure.

Initially:

training set of multiple sequences of observations, initial model estimate.

Procedure:

segment each observation sequence into states using a Viterbi procedure. For discrete observation densities, code all observations in state *j* using the *M*-codeword codebook, giving

$b_j(k)$ = number of vectors with codebook index *k*, in state *j*, divided by the number of vectors in state *j.*

for continuous observation densities, cluster the observations in state *j* into a set of *M* clusters, giving

Segmental K-Means Segmentation into States

c_{jm} = number of vectors assigned to cluster *m* of state *j* divided by the number of vectors in state *j*.

μ_{jm} = sample mean of the vectors assigned to cluster *m* of state *j*

U_{jm} = sample covariance of the vectors assigned to cluster *m* of state *j*

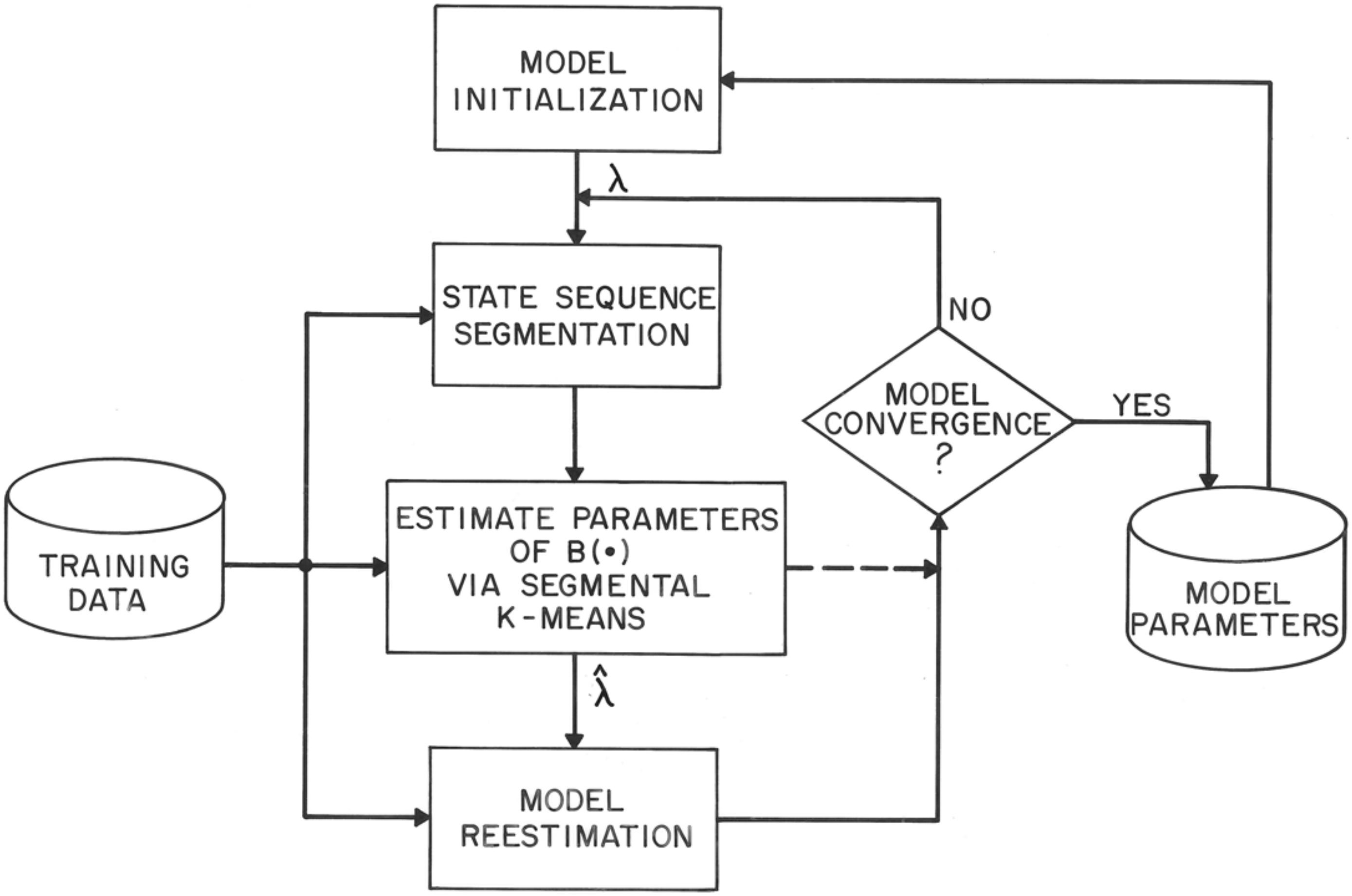
use as the estimate of the state transition probabilities

a_{ii} = number of vectors in state *i* minus the number of observation sequences for the training word divided by the number of vectors in state *i.*

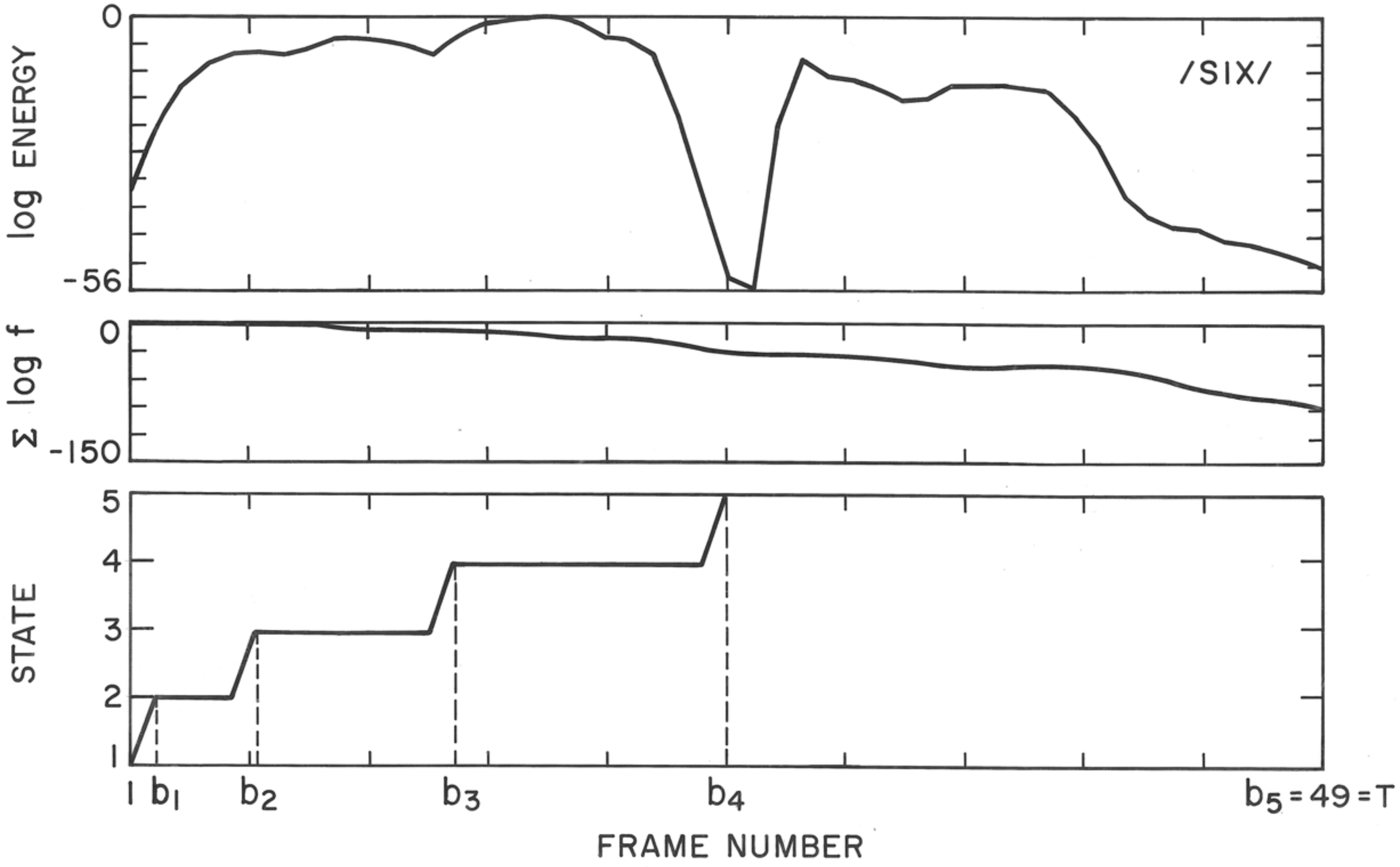
$a_{i,i+1} = 1 – a_{ii}$

the segmenting HMM is updated and the procedure is iterated until a converged model is obtained.

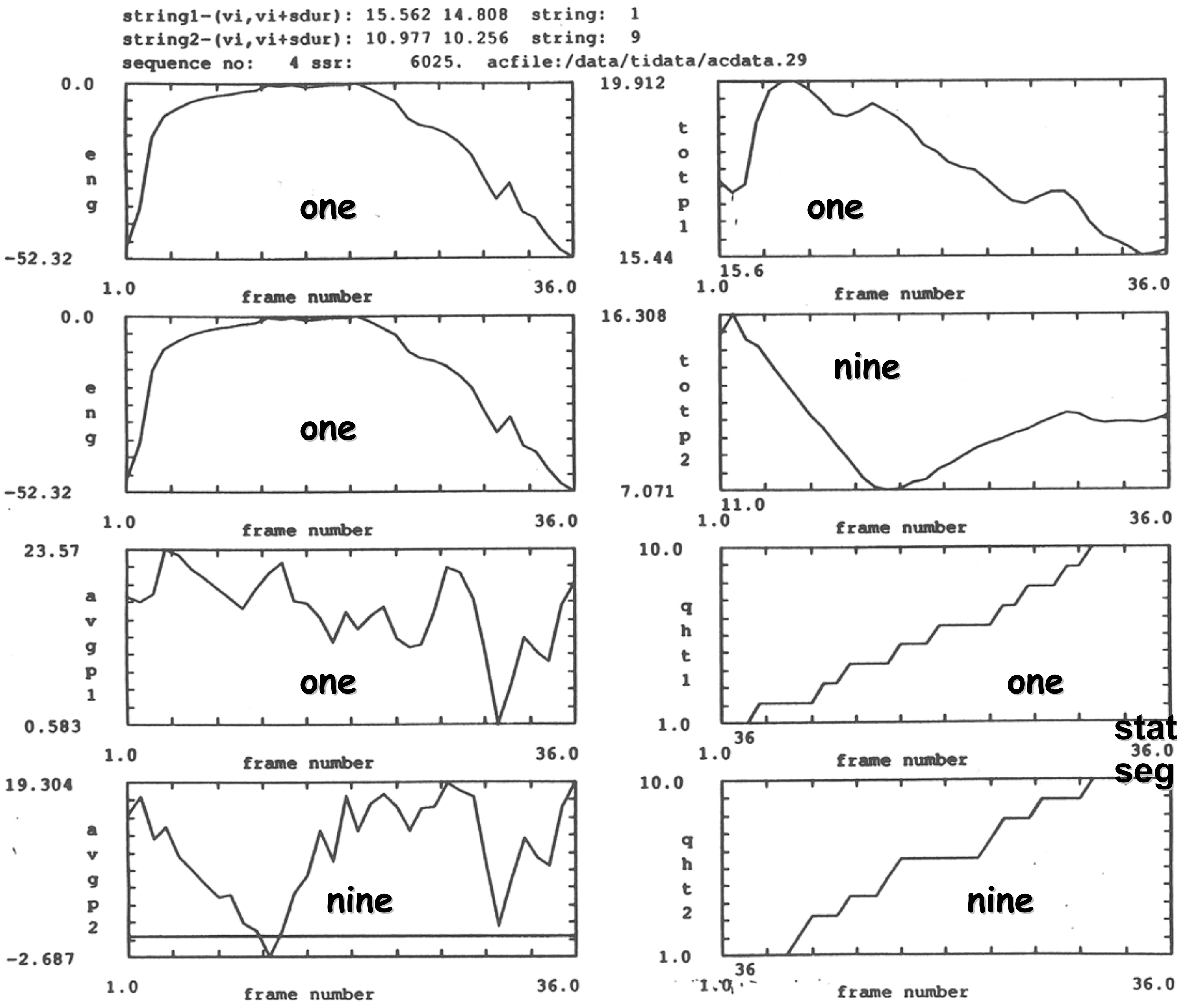
Segmental K-Means Training



HMM Segmentation for /SIX/



Digit Recognition Using HMM’s



```
string1-(vi,vi+sdur): 15.562 14.808   string:  1
string2-(vi,vi+sdur): 10.977 10.256   string:  9
sequence no:   4 ssr:      6025.  acfile:/data/tidata/acdata.29
```

unknown
log
energy

frame
likelihood
scores

frame
cumulative
scores

state
segmentation

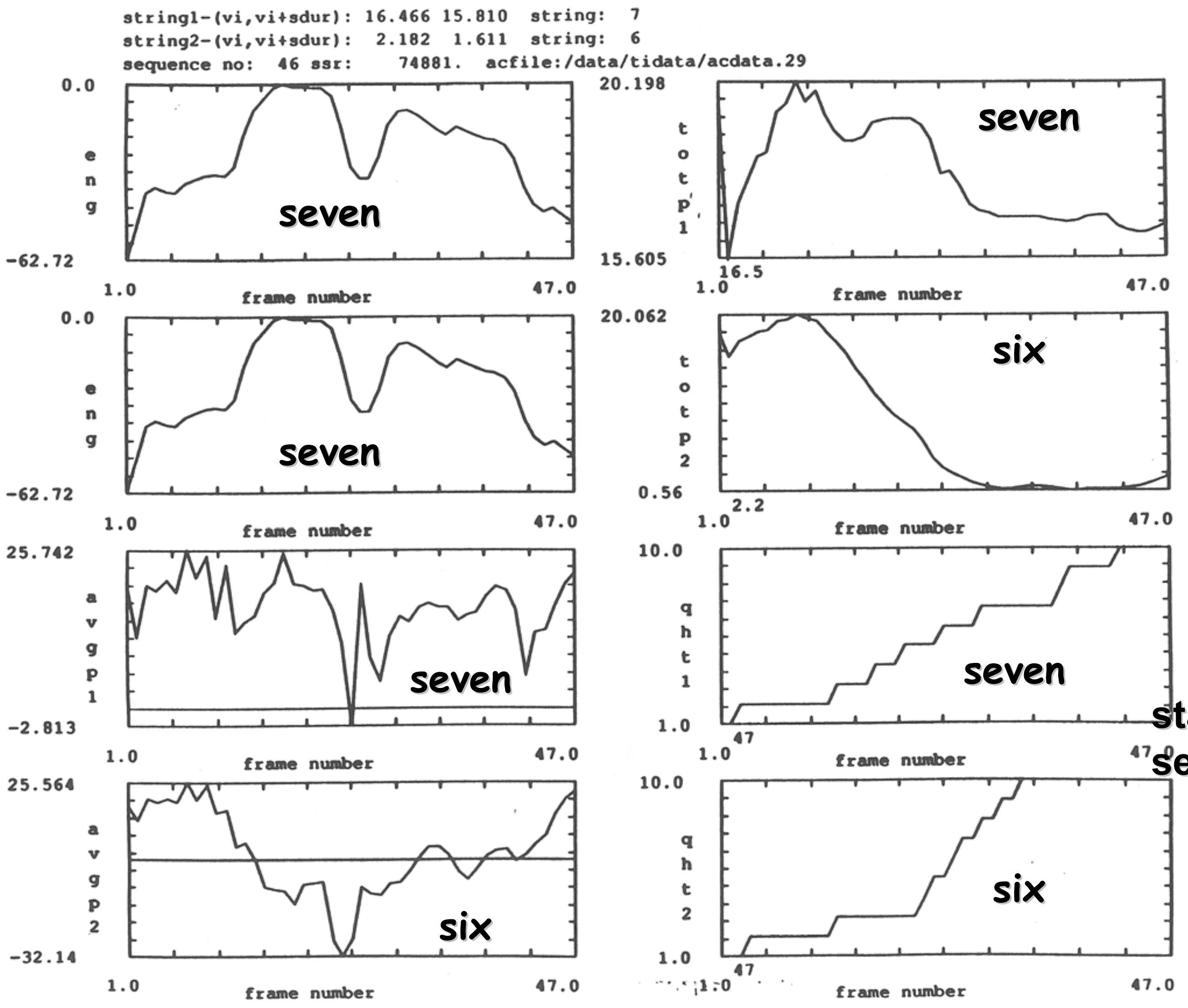
Digit Recognition Using HMM’s

unknown
log energy

frame
cumulative
scores

frame
likelihood
scores

state
segmentation



```
string1-(vi,vi+sdur): 16.466 15.810  string:  7
string2-(vi,vi+sdur):  2.182  1.611  string:  6
sequence no:  46 ssr:     74881.  acfile:/data/tidata/acdata.29
```

HMM PERFORMANCE ON SPEAKER INDEPENDENT, ISOLATED DIGITS

Recognizer Type	Original Training Set	Test Set 2	Test Set 3	Test Set 4
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

AVERAGE DIGIT ERROR RATES (%)

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

LPC/DTW/VQ – Conventional recognizer with vector quantization ($M = 64$ codebook).

HMM/VQ – HMM Recognizer with $M = 64$ codebook.

HMM/CD – HMM recognizer using continuous density model with 5 mixtures per state.

MHH/AR – HMM recognizer using mixture autoregressive observation density.