Hidden Markov Models for Automatic Speech Recognition Part II

R. Hegde
EE627 : Speech Signal Processing
Dept. of Electrical Engg.

Problem (2) – Learning Problem

How do we adjust model parameters λ to maximize $P(\mathcal{O}|\lambda)$?

Solution : Reestimation Procedure

- initial state distribution :

$$\bar{\pi}_i = \text{expected frequency in } s_i \text{ at time } 1$$

- state transition probability distribution :

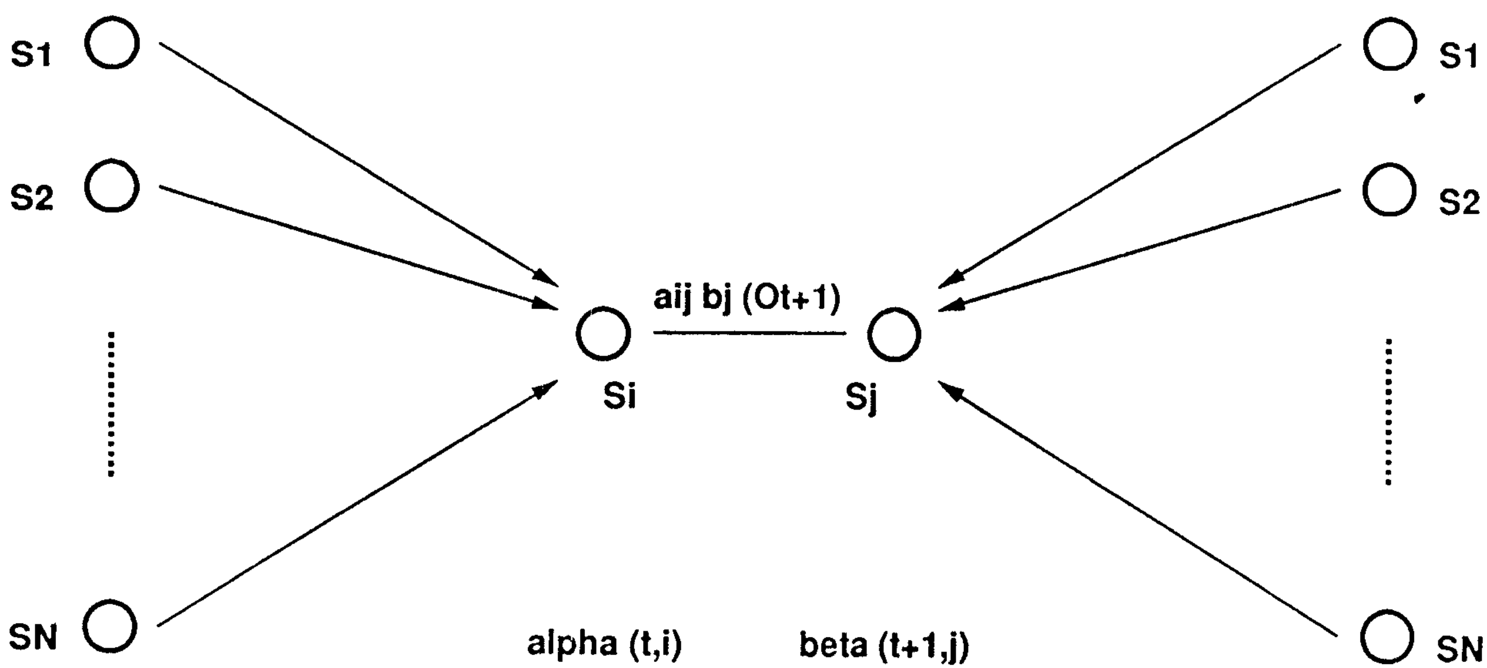
$$\bar{a}_{ij} = \frac{\text{expected \# of transitions from } s_i \text{ to } s_j}{\text{expected \# of transitions from } s_i}$$

- observation symbol probability distribution in s_j :

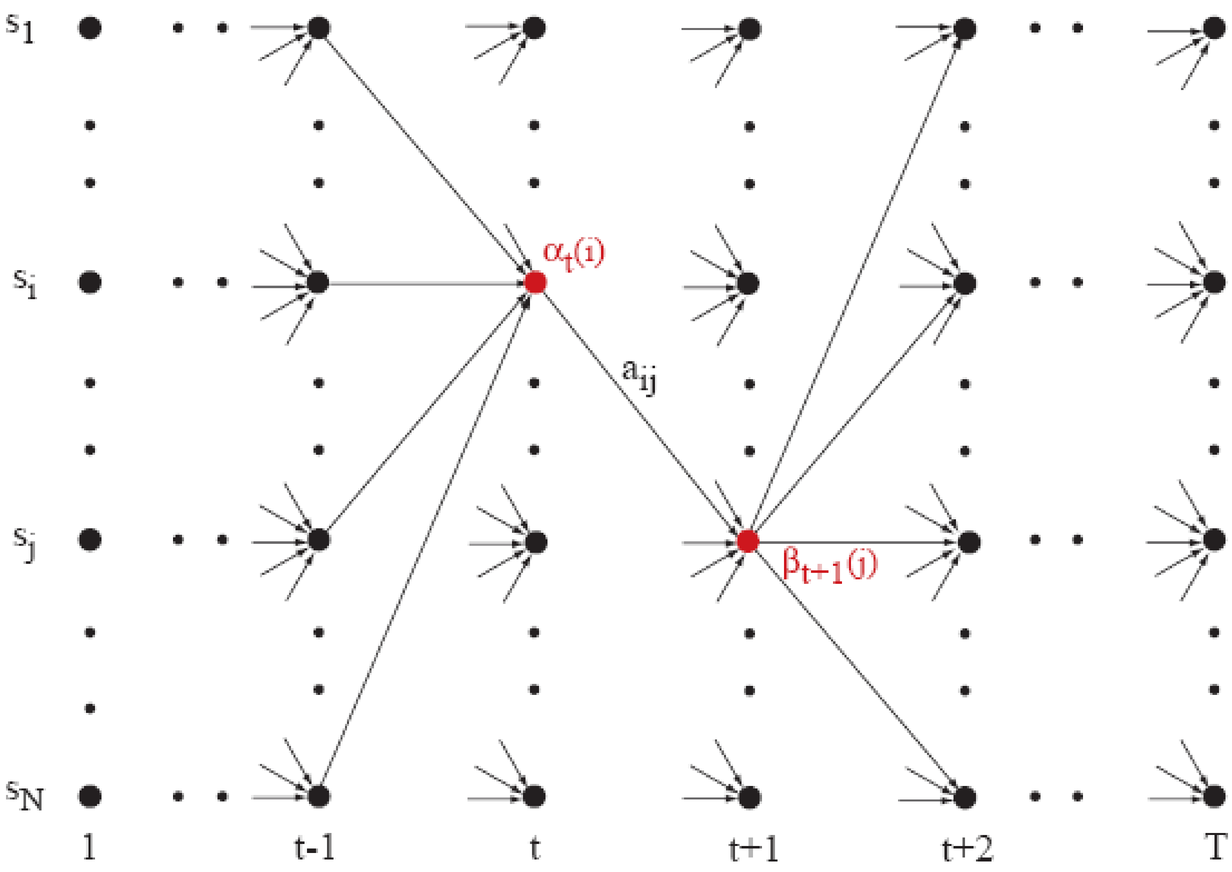
$$\bar{b}_j(k) = \frac{\text{expected frequency in } s_j \text{ and observing}}{\text{expected frequency in } s_j}$$

ξ terms :

$$
\begin{aligned}
\xi(t,i,j) &= P(s_i @\text{time } t,\ s_j @\text{time } t+1 | \mathcal{O}, \lambda) \\
&= \frac{P(s_i @\text{time } t,\ s_j @\text{time } t+1, \mathcal{O} | \lambda)}{P(\mathcal{O}|\lambda)} \\
&= \frac{\alpha(t,i) a_{ij} b_j(o_{t+1}) \beta(t+1,j)}{P(\mathcal{O}|\lambda)}
\end{aligned}
$$



Forward-Backward Illustration



γ **terms :**

$$\begin{aligned}\gamma(t,i) &= P(s_i\text{@time } t|\mathcal{O},\lambda)\\ &= \frac{P(s_i\text{@time } t,\mathcal{O}|\lambda)}{P(\mathcal{O}|\lambda)}\\ &= \frac{\alpha(t,i)\beta(t,i)}{P(\mathcal{O}|\lambda)}\end{aligned}$$

Relation between γ terms and ξ terms :

$$\gamma(t,i) = \sum_{j=1}^{N} \xi(t,i,j)$$

$$\sum_{t=1}^{T-1} \xi(t,i,j) = \text{expected \# of transitions from } s_i \text{ to } s_j$$

$$\sum_{t=1}^{T-1} \gamma(t,i) = \text{expected \# of transitions from } s_i$$

$$\sum_{t=1}^{T} \gamma(t,i) = \text{expected frequency in } s_j$$

Reestimation Equations :

- initial state distribution :

$$\bar{\pi}_i = \gamma(1, i), \qquad i = 1, 2, \cdots, N$$

- state transition probability distribution :

$$\bar{a}_{ij} = \frac{\sum_{t=1}^{T-1} \xi(t, i, j)}{\sum_{t=1}^{T-1} \gamma(t, i)}, \qquad i,\ j = 1, 2, \cdots, N$$

- observation symbol probability distribution in s_j :

$$\bar{b}_j(k) = \frac{\sum_{t=1}^{T} \{\gamma(t, i) \text{ s.t. } o_t = vk\}}{\sum_{t=1}^{T} \gamma(t, i)}$$

$$j = 1, 2, \cdots, N \qquad m = 1, 2, \cdots, M$$

Note :

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

γ **terms for Continuous Observation Density :**

$$\begin{aligned}\gamma(t,i,m) &= P(s_i\text{@time } t, \text{mixture } m|\mathcal{O},\lambda) \\ &= \frac{\alpha(t,i)\beta(t,i)}{P(\mathcal{O}|\lambda)} \cdot \frac{c_{jm}\mathcal{N}(\mathbf{x},\mathbf{m}_{jm},\mathbf{\Sigma}_{jm})}{\sum_{n=1}^{M} c_{jn}\mathcal{N}(\mathbf{x},\mathbf{m}_{jn},\mathbf{\Sigma}_{jn})}\end{aligned}$$

Reestimation Equations :

- mixture coefficient :

$$\bar{c}_{jm} = \frac{\sum_{t=1}^{T} \gamma(t,i,m)}{\sum_{t=1}^{T}\sum_{n=1}^{M} \gamma(t,i,n)}$$

$$j = 1,2,\cdots,N \qquad m = 1,2,\cdots,M$$

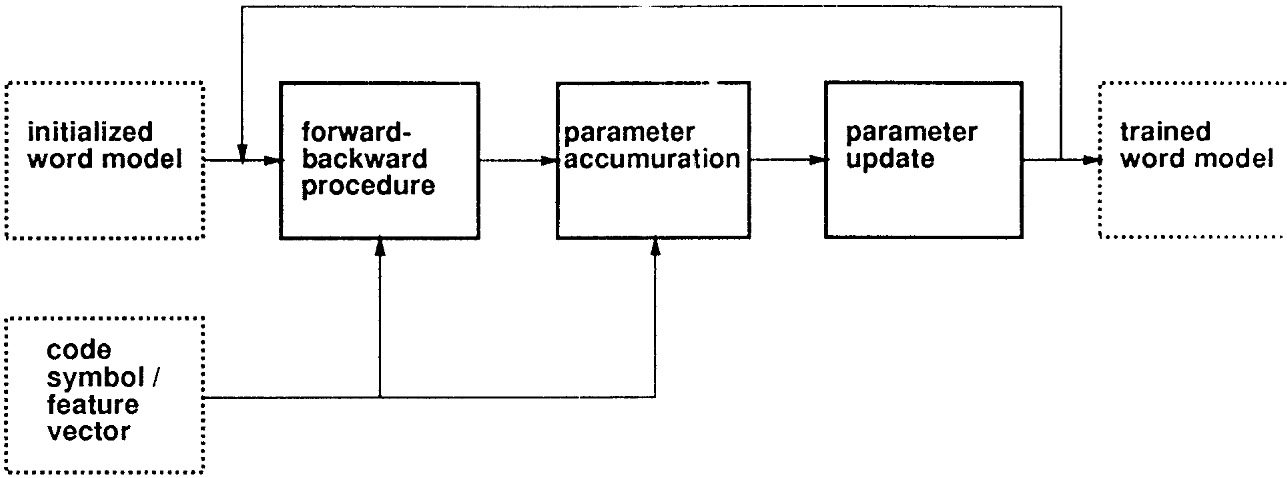
- mean :

$$\bar{\mathbf{m}}_{jm} = \frac{\sum_{t=1}^{T} \gamma(t,i,m)\cdot o_t}{\sum_{t=1}^{T} \gamma(t,i,m)}$$

- covariance :

$$\bar{\mathbf{\Sigma}}_{jm} = \frac{\sum_{t=1}^{T} \gamma(t,i,m)\cdot(\mathbf{o}_t - \mathbf{m}_{jm})^T(\mathbf{o}_t - \mathbf{m}_{jm})}{\sum_{t=1}^{T} \gamma(t,i,m)}$$

Training Loop



Given :

- sequence of code symbol (discrete case) :

$$\mathcal{O} = o_1, o_2, \cdots, o_T$$

- sequence of feature vector (continuous case) :

$$\mathcal{O} = \mathbf{o}_1, \mathbf{o}_2, \cdots, \mathbf{o}_T$$

Want to Generate :

- word model λ (discrete or continuous) :

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

Restimation : Revisited

- If $\lambda = (\boldsymbol{A}, \boldsymbol{B}, \pi)$ is the initial model, and $\bar{\lambda} = (\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}, \bar{\pi})$ is the re-estimated model. Then it can be proved 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 λ in the sense that $P(\boldsymbol{O}|\bar{\lambda}) > P(\boldsymbol{O}|\lambda)$, i.e., we have found a new model $\bar{\lambda}$ from which the observation sequence is more likely to have been produced.

- Thus we can improve the probability of $\boldsymbol{O}$ being observed from the model if we 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 HMM.

How to find optimal state sequence

- One criterion chooses states, q_t, which are *individually* most likely
  - This maximizes the expected number of correct states
- Let us 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.

$$\gamma_t(i) = P(q_t = s_i | \boldsymbol{O}, \lambda) \qquad \sum_{i=1}^{N} \gamma_t(i) = 1, \qquad \forall t$$

- Then the individually most likely state, q_t, at time t is:

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

- Note that it can be shown that:

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(\boldsymbol{O}|\lambda)}$$

Problem (3) – Decoding Problem

Given the observation sequence $\mathcal{O}$ and the model λ, how do we choose a state sequence $\mathcal{Q}$, which is optimal in some meaningful sense (that is, best "explains" the observations)?

Solution :

$$q_t = \operatorname*{argmax}_{1 \le i \le N} \gamma(t, i), \qquad t = 1, 2, \cdots, T$$

Viterbi Algorithm :

δ terms :

$$\delta_t(i) = \max_{q_1, \cdots, q_{t-1}} P(q_1, \cdots, q_t,\ o_1, \cdots, o_t,\ s_i\text{@time } t|\lambda)$$

(best score along a single state path $q_1, \cdots, q_t$ which accounts for a observation sequence $o_1, \cdots, o_t$ and ends in state s_i)

Induction :

$$\delta_{t+1}(j) = \left[\max_i \delta_t(i) a_{ij}\right] b_j(o_{t+1})$$

Viterbi Algorithm :

Initialization :

$$\delta_1(i) = \pi_i b_i(o_1)$$

$$\psi_1(i) = 0, \qquad i = 1, 2, \cdots, N$$

Recursion :

$$\delta_t(j) = \max_{1 \leq i \leq N} \left[\delta_{t-1}(i) a_{ij}\right] b_j(o_{t+1})$$

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

$$j = 1, 2, \cdots, N \qquad t = 2, 3, \cdots, T$$

Termination :

$$p^* = \max_{1 \leq i \leq N} \delta_T(i)$$

$$q_T^* = \operatorname*{argmax}_{1 \leq i \leq N} \delta_T(i)$$

Path (state sequence) backtracking :

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

Word Set :

alphabet	a,b, ... ,z
digit	0,1, ... ,9
misc.	period, space, silence

Result :

```
 utterance -> " 6 1 3 7 6 8 _ 3 4 4 6 7 6 "
recognized -> " 6 1 3 7 6 8 _ 3 4 4 6 7 6 "

 utterance -> " l a c q u e r _ j a m a i c a "
recognized -> " l a c q u e i _ j a n a i d k "
```

Confusion Matrix :

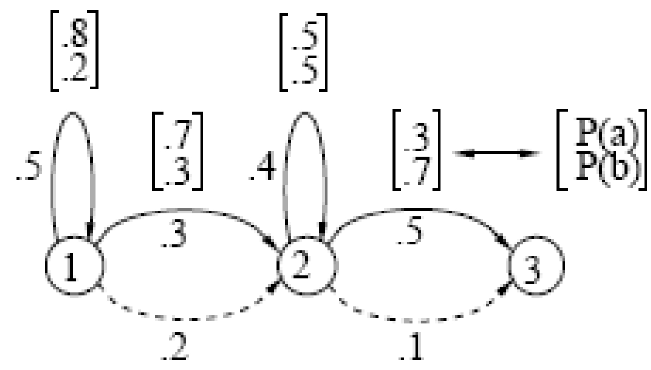
A-set :

	a	h	j	k	8
a	73	3	1	3	23
h	0	91	.	.	1
j	1	.	81	4	.
k	3	.	7	83	1
8	10	3	.	1	63
~	12	3	11	10	13

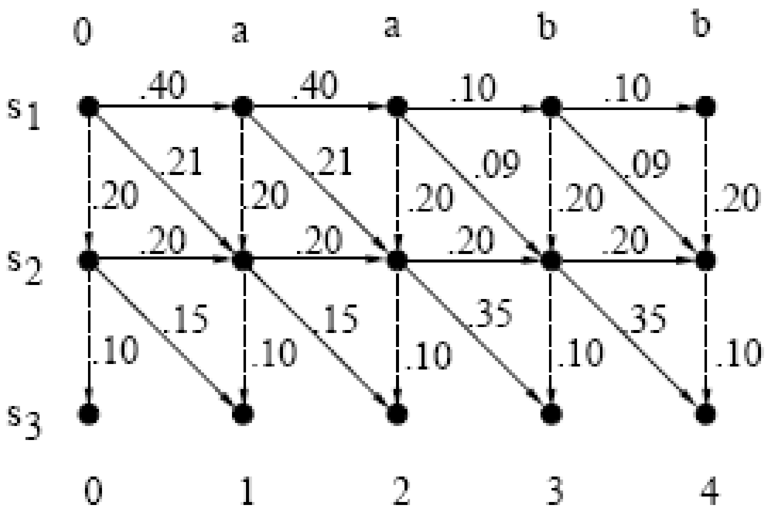
Nasal/Glide :

	l	m	n	7	9
l	71	1	1	.	.
m	5	64	12	1	1
n	1	25	75	1	4
7	.	.	.	97	.
9	0	.	1	.	87
~	23	11	12	2	9

Viterbi Algorithm : An Example

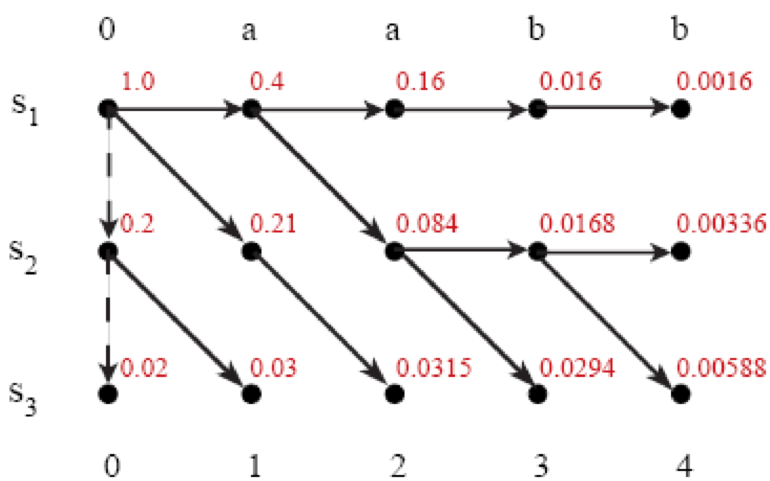


O={a a b b}



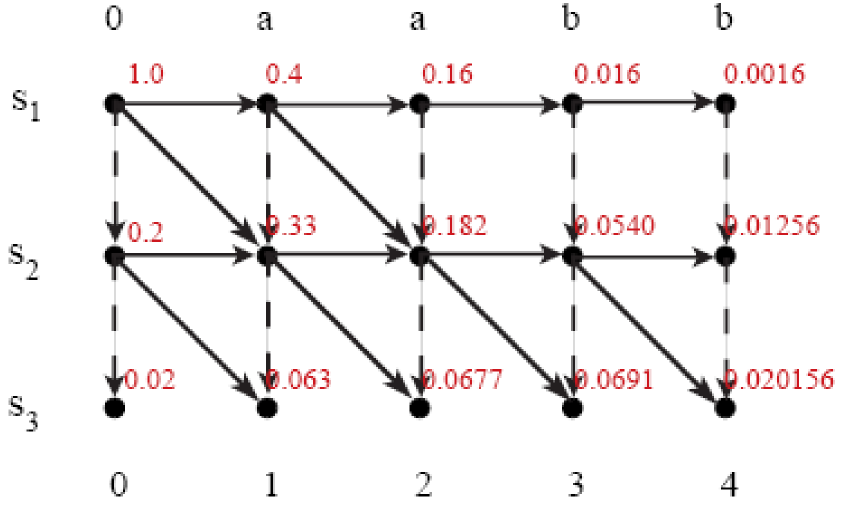
Viterbi Algorithm : Contd.

	0	a	aa	aab	$aabb$
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .042	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0168	$s_1, 0$ .00032 s_1, b .00144 s_2, b .00336
s_3	$s_2, 0$ .02	$s_2, 0$ .021 s_2, a .03	$s_2, 0$ .0084 s_2, a .0315	$s_2, 0$ .00168 s_2, b .0294	$s_2, 0$ .000336 s_2, b .00588



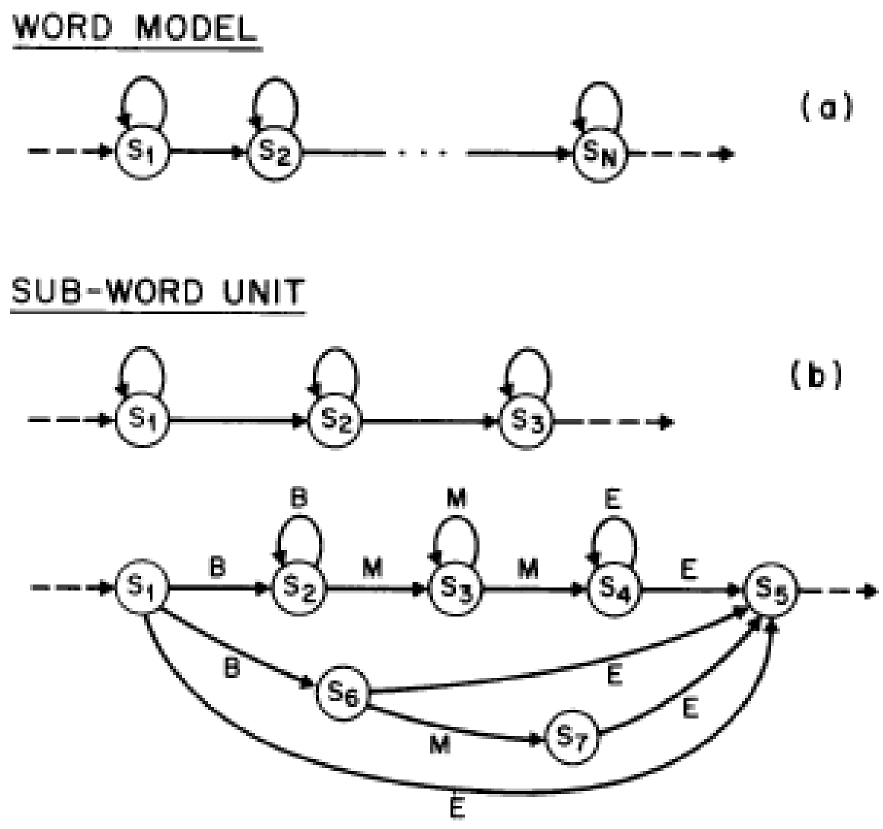
Decoding using Forward-Backward Algorithm

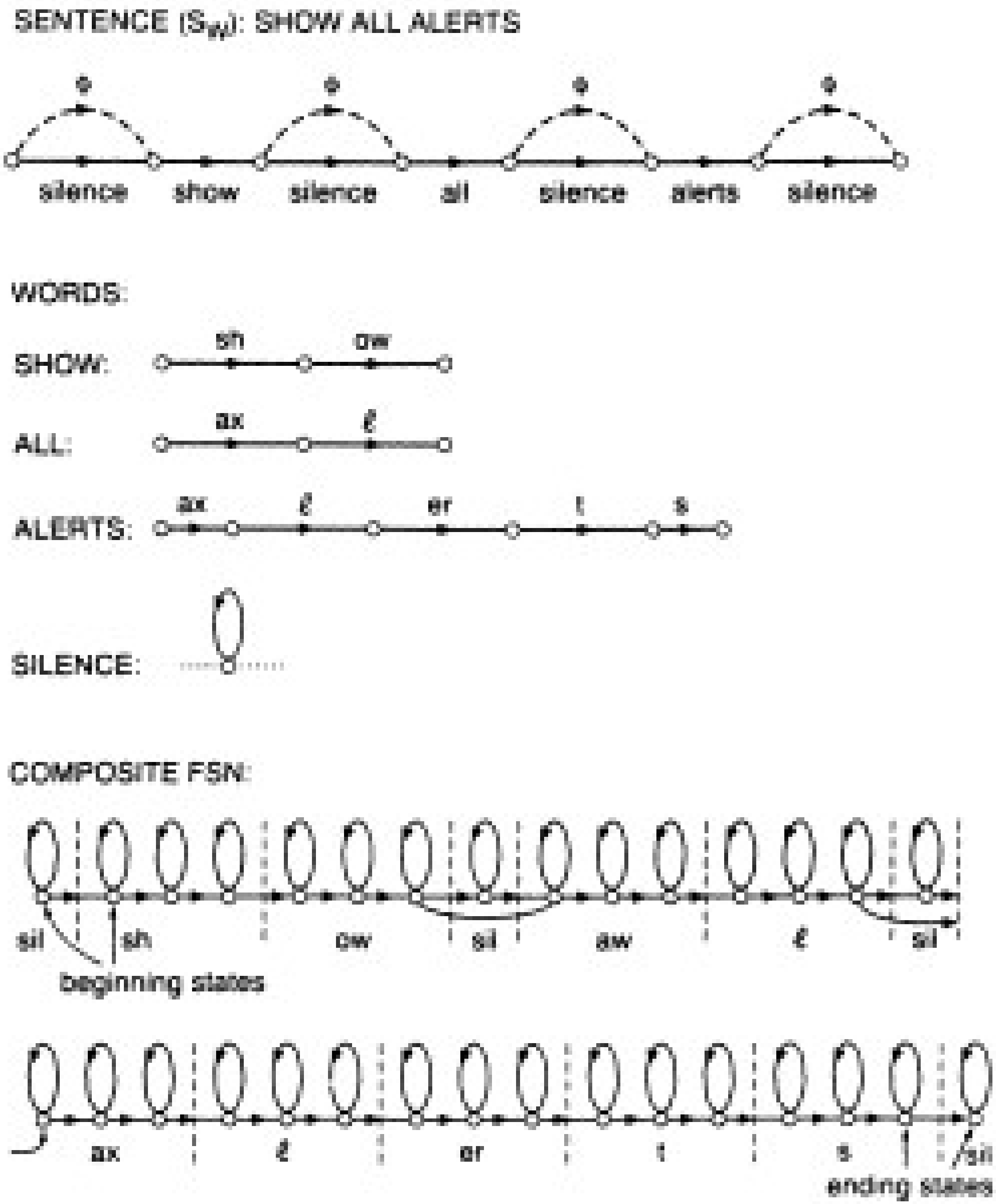
	0	a	aa	aab	$aabb$
s_1	1.0	s_1, a .4	s_1, a .16	s_1, b .016	s_1, b .0016
s_2	$s_1, 0$ .2	$s_1, 0$ .08 s_1, a .21 s_2, a .04	$s_1, 0$ .032 s_1, a .084 s_2, a .066	$s_1, 0$ .0032 s_1, b .0144 s_2, b .0364	$s_1, 0$ .00032 s_1, b .00144 s_2, b .0108
s_3	$s_2, 0$ .02	$s_2, 0$ .033 s_2, a .03	$s_2, 0$ .0182 s_2, a .0495	$s_2, 0$ .0054 s_2, b .0637	$s_2, 0$ .001256 s_2, b .0189



Word and Phone Based Models

- Small Voc : Word based models
- Large Vocabulary : Phone based models





SENTENCE (S_W): SHOW ALL ALERTS
silence
show
silence
all
silence
alerts
silence
WORDS:
SHOW:
sh
ow
ALL:
ax
ℓ
ALERTS:
ax
ℓ
er
t
s
SILENCE:
COMPOSITE FSN:
sil
sh
ow
sil
aw
ℓ
sil
beginning states
ax
ℓ
er
t
s
sil
ending states

Continuous (Density) HMM

- A *continuous density* HMM replaces the discrete observation probabilities, $b_j(k)$, by a continuous PDF $b_j(\boldsymbol{x})$
- A common practice is to represent $b_j(\boldsymbol{x})$ as a mixture of Gaussians:

$$b_j(\boldsymbol{x}) = \sum_{k=1}^{M} c_{jk} N[\boldsymbol{x}, \mu_{jk}, \boldsymbol{\Sigma}_{jk}] \qquad 1 \le j \le N$$

where c_{jk} is the mixture weight

$c_{jk} \ge 0 \quad (1 \le j \le N, 1 \le k \le M$, and $\sum_{k=1}^{M} c_{jk} = 1, 1 \le j \le N)$,

N is the normal density, and

μ_{jk} and $\boldsymbol{\Sigma}_{jk}$ are the mean vector and covariance matrix associated with state j and mixture k.

Semi Continuous HMMs

- *Semi-continuous* HMMs first compute a VQ codebook of size M
  - The VQ codebook is then modelled as a family of Gaussian PDFs
  - Each codeword is represented by a Gaussian PDF, and may be used together with others to model the acoustic vectors
  - From the CD-HMM viewpoint, this is equivalent to using the same set of M mixtures to model all the states
  - It is therefore often referred to as a *Tied Mixture* HMM
- All three methods have been used in many speech recognition tasks, with varying outcomes
- For large-vocabulary, continuous speech recognition with sufficient amount (i.e., tens of hours) of training data, CD-HMM systems currently yield the best performance, but with considerable increase in computation

ASR Initialization : Iteration Issues

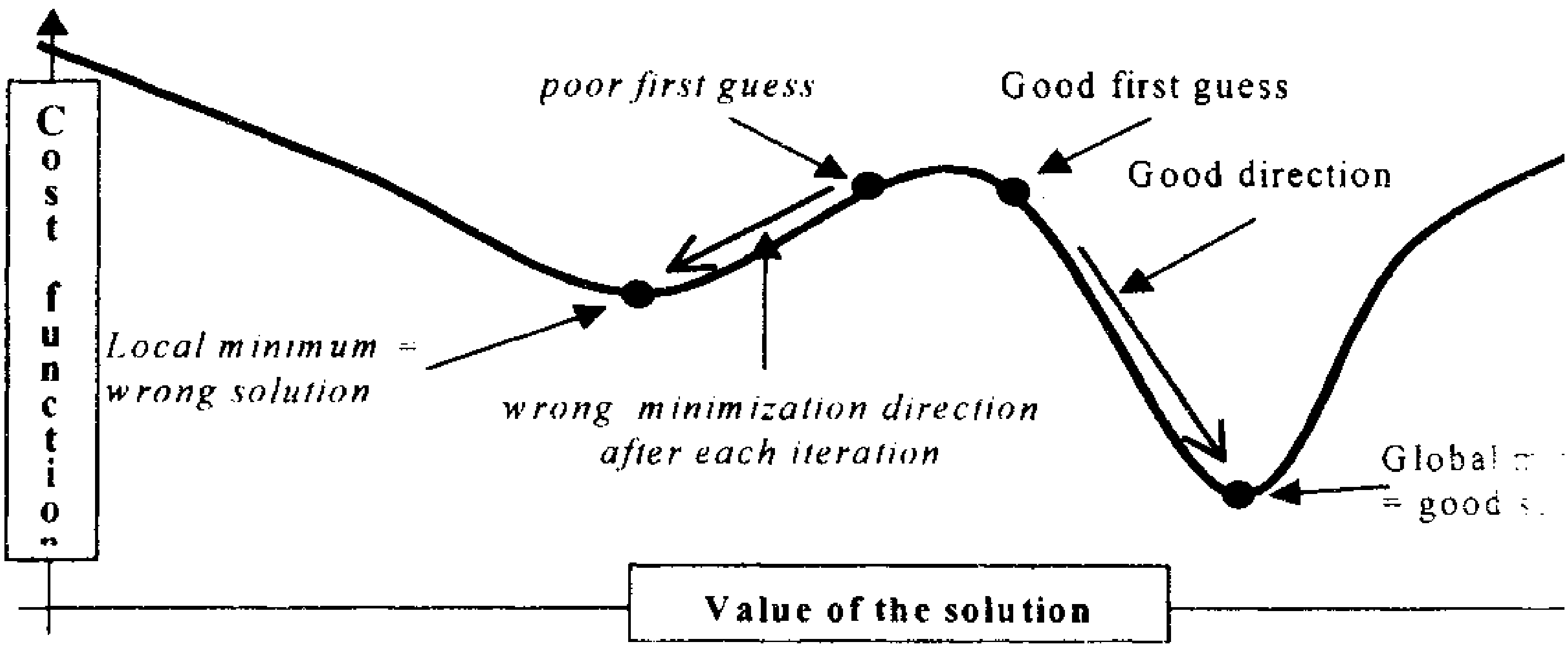


Figure 4.10 Finding the right solution in iterative algorithms

ASR Initialization

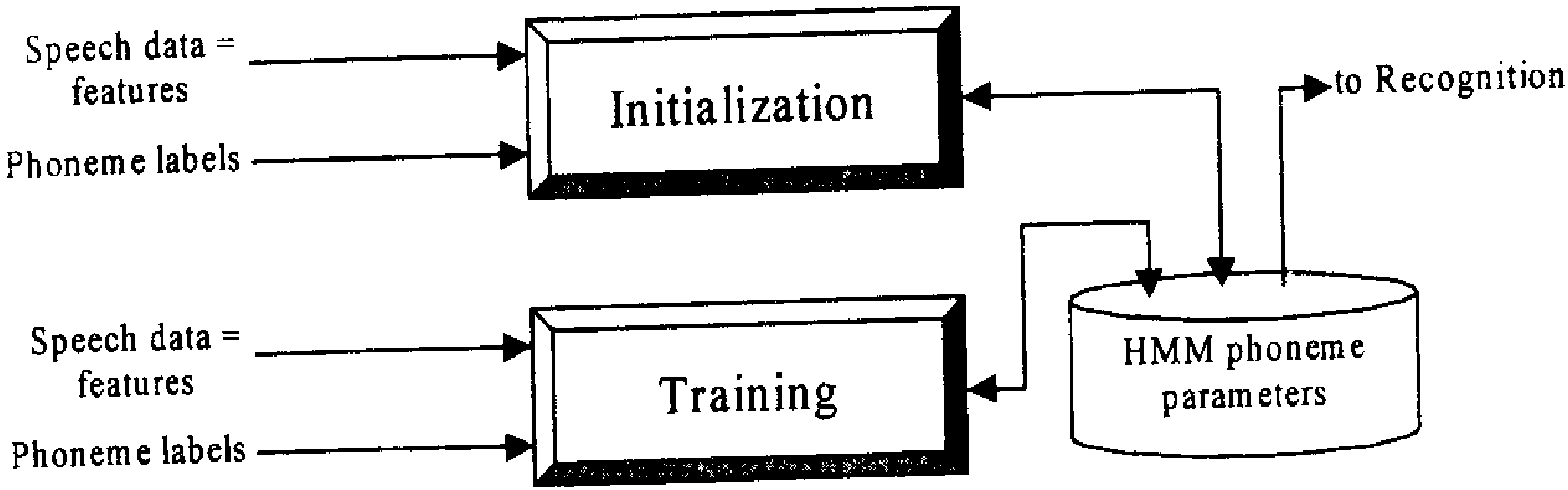


Figure 4.11 Initialization in ASR

Hmm Phoneme Model

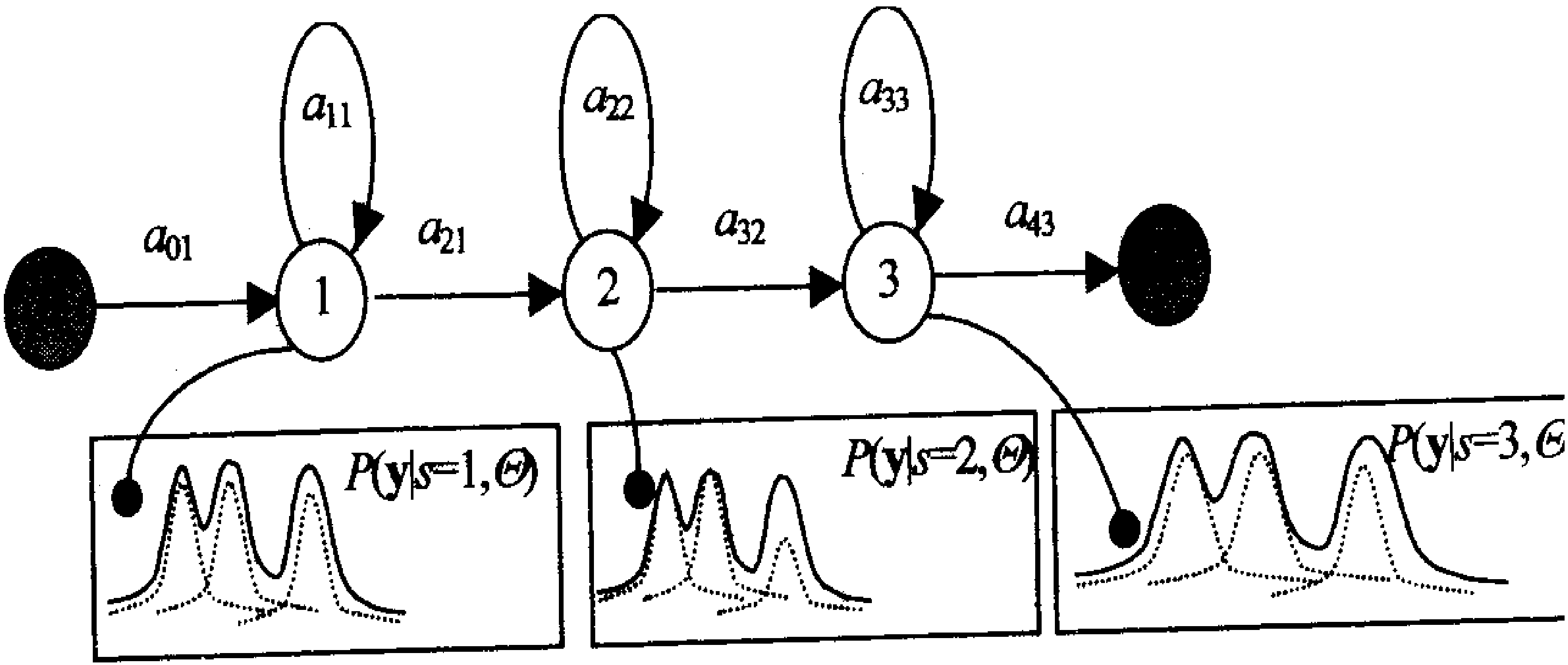
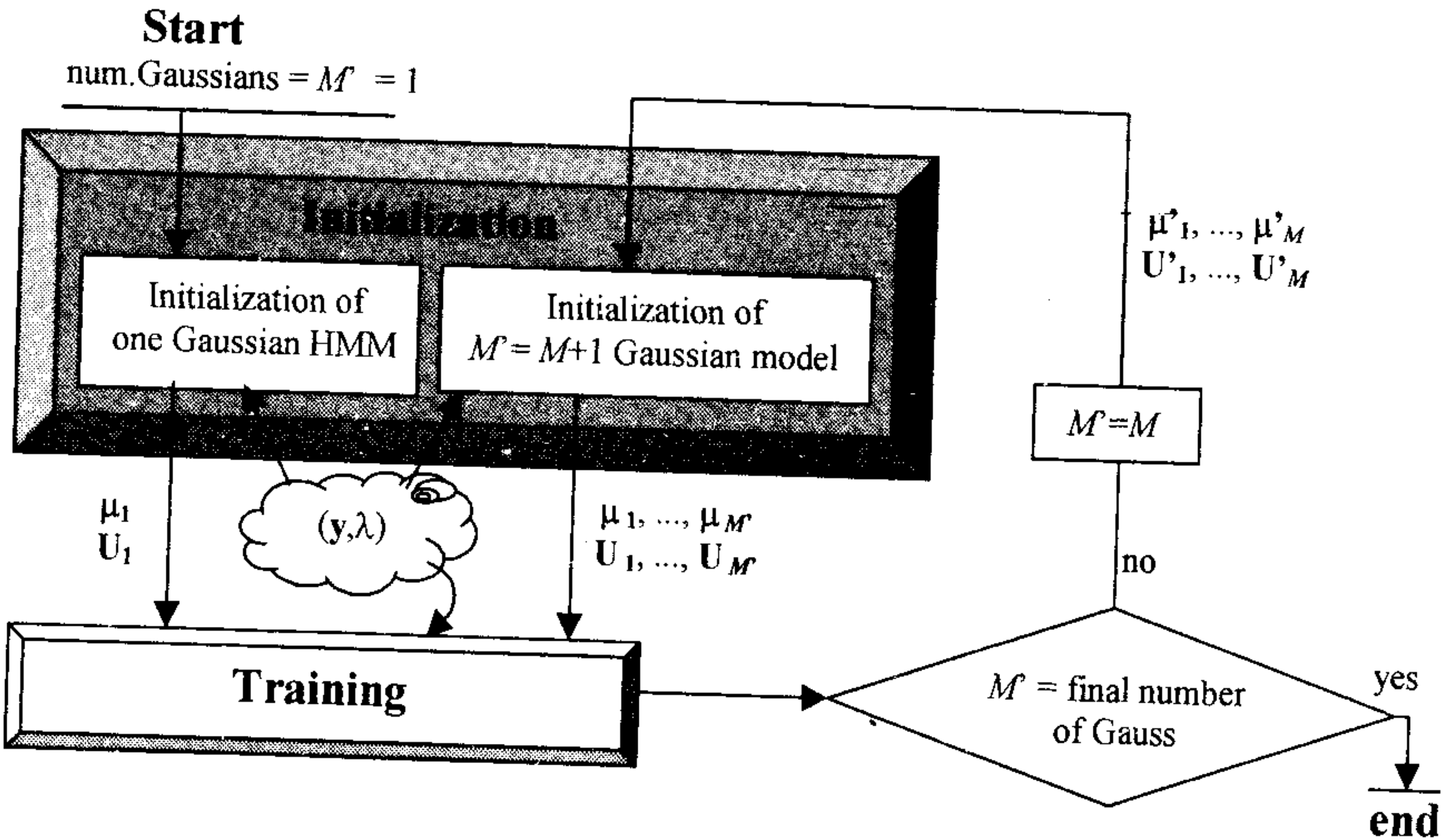
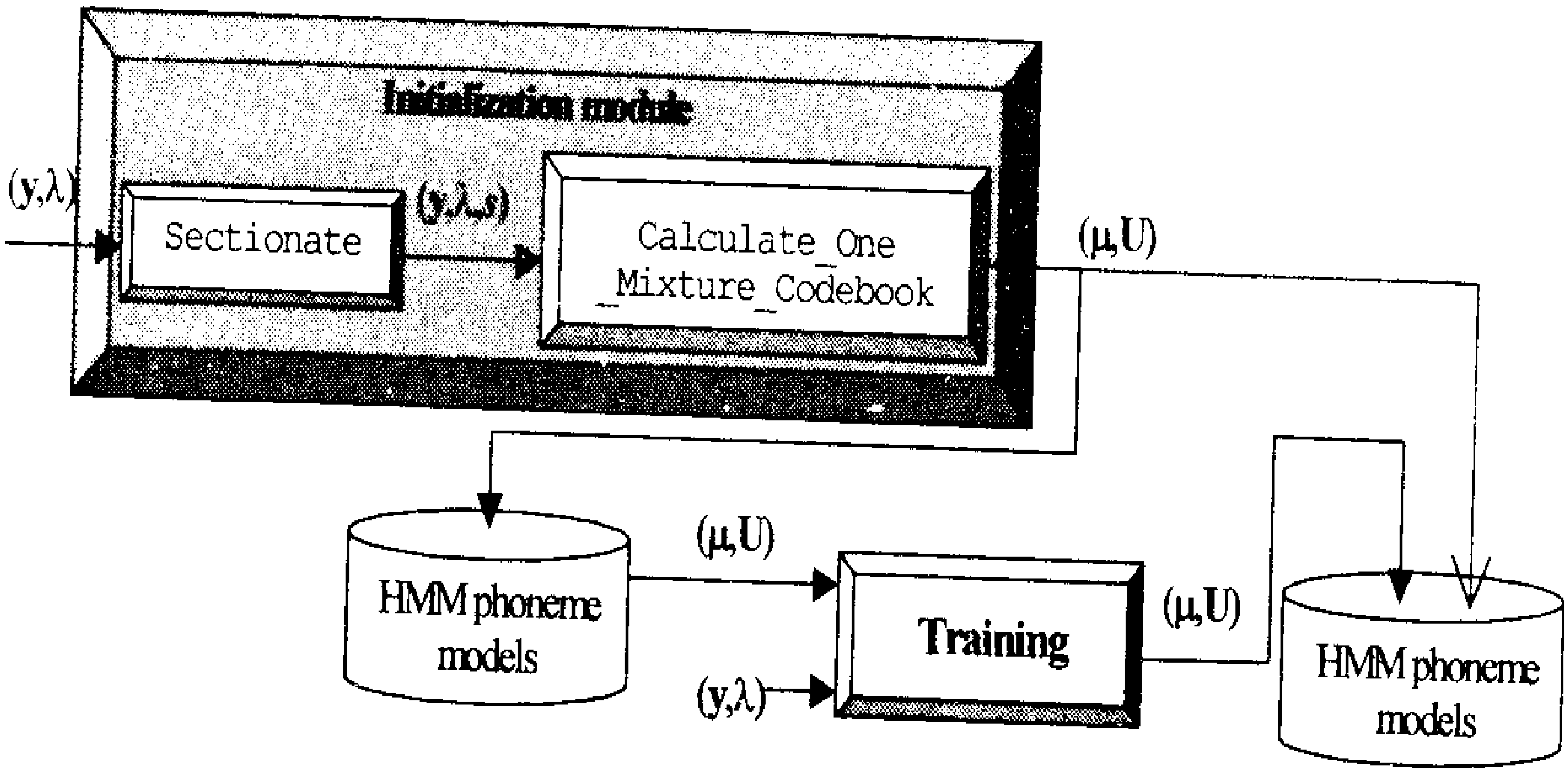


Figure 4.12 HMM graph

HMM Parameter Estimation



Initialization of Single Gaussian HMM



Increasing the Number of Gaussian pdfs in the HMM

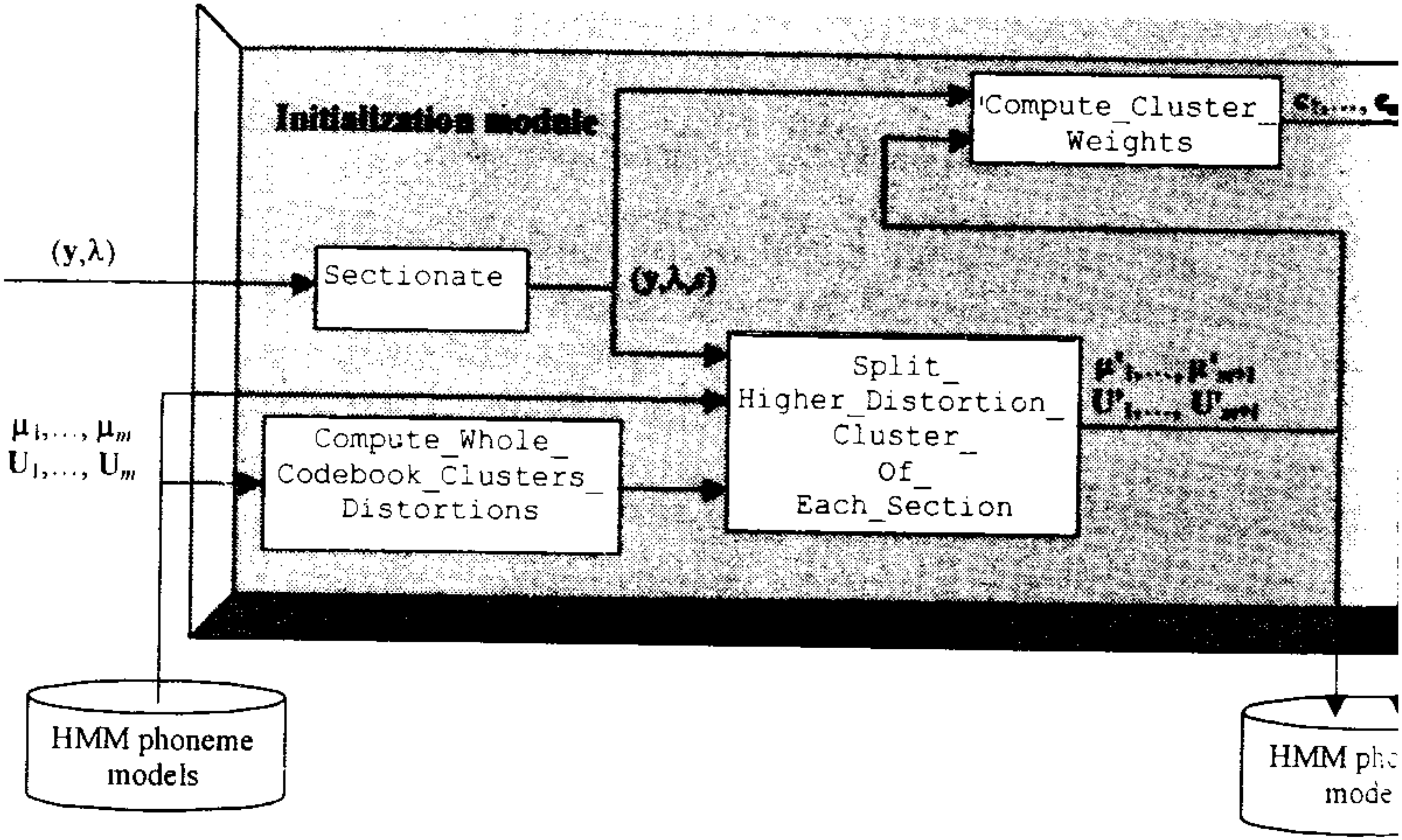


Figure 4.15 Increasing the number of Gaussian pdfs in the mixtures

Cluster Splitting : Initial Step

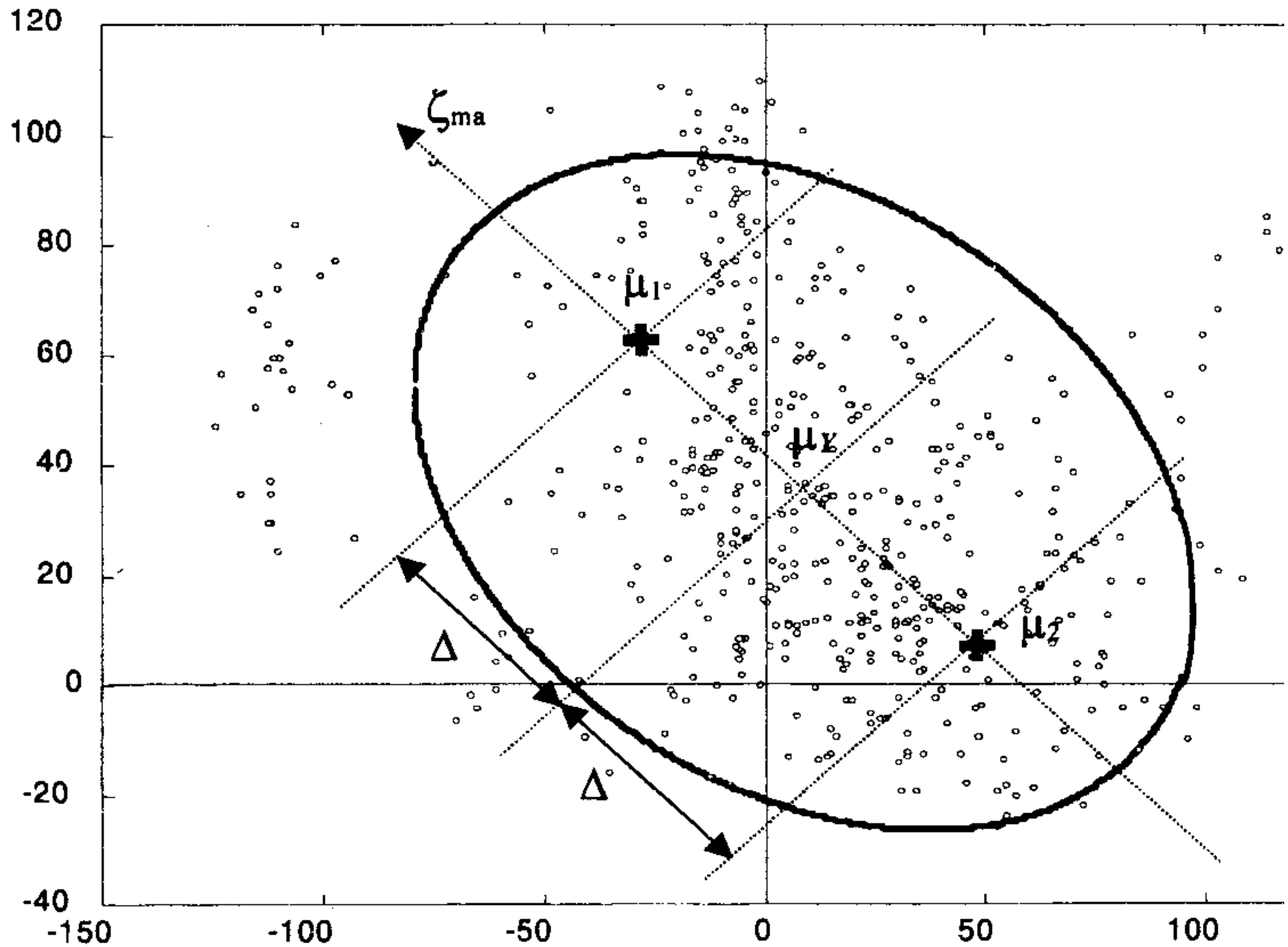


Figure 4.8 Initial step of the cluster splitting algorithm

Cluster Splitting : Next Step

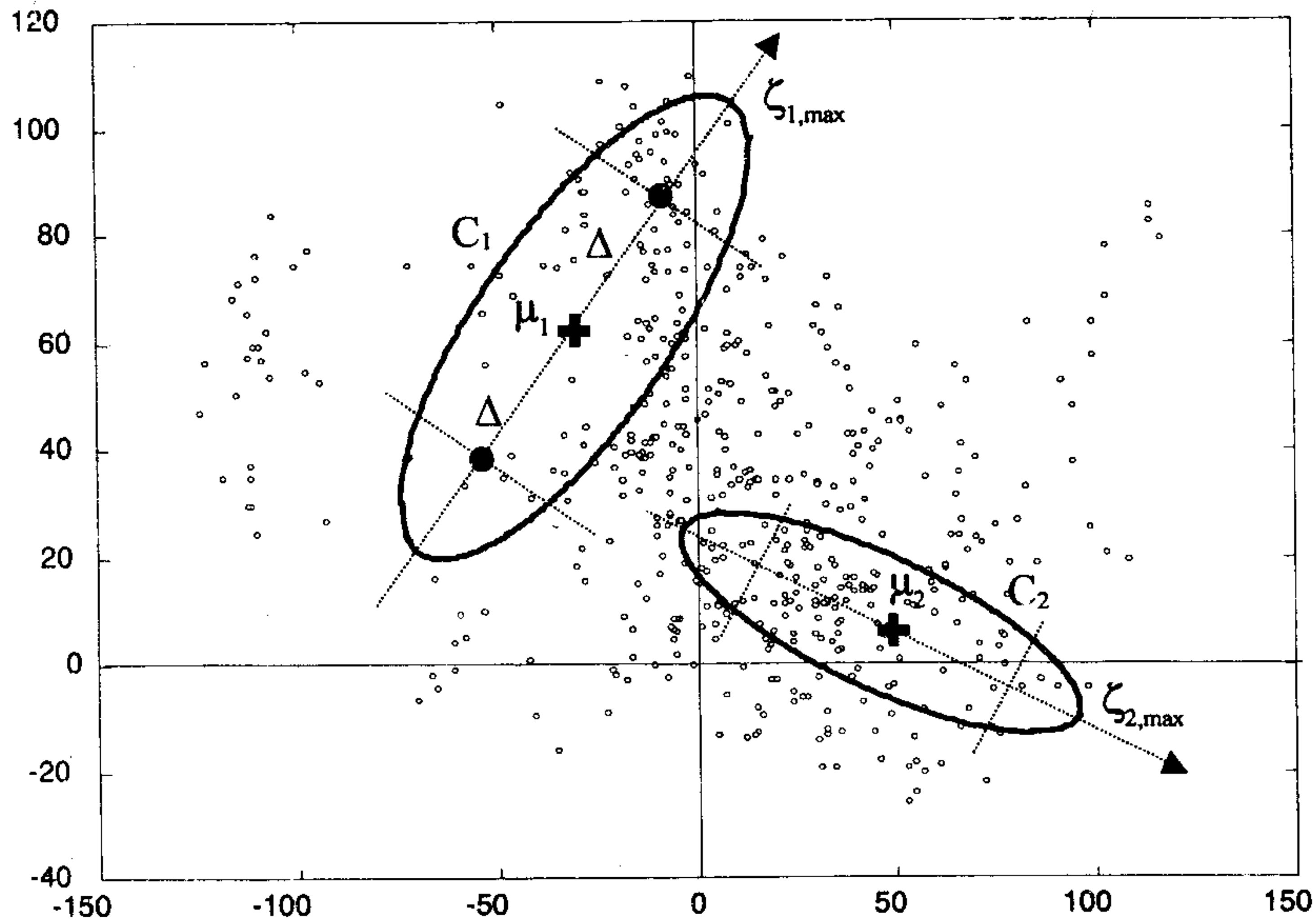


Figure 4.9 Second iterative step of the cluster splitting algorithm

Cluster Spitting Algorithm

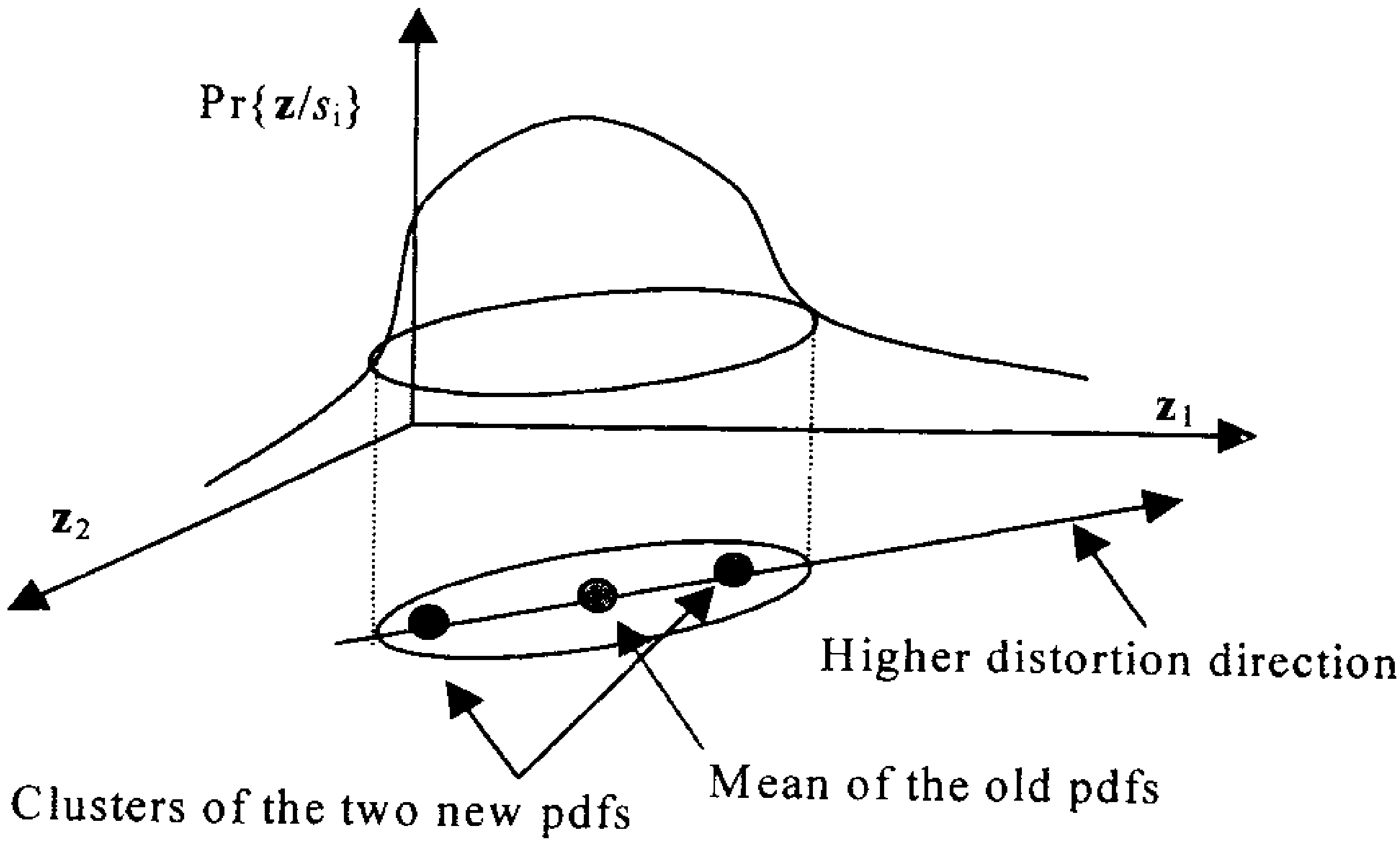


Figure 4.16 Splitting algorithm

Acoustic Model Training in Practice

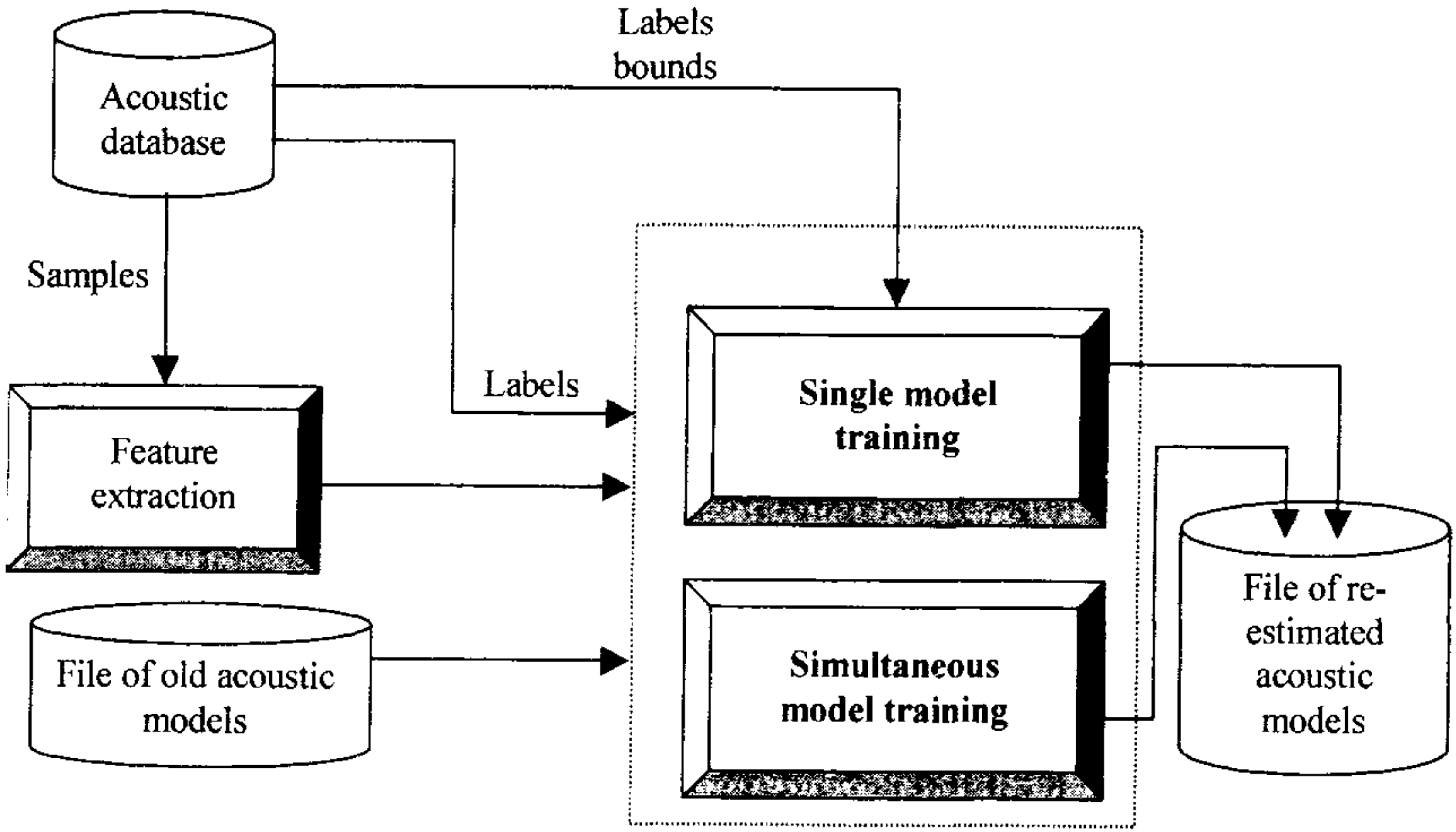


Figure 5.2 Acoustic trainer in RES

The Training Process : BW Re-estimation

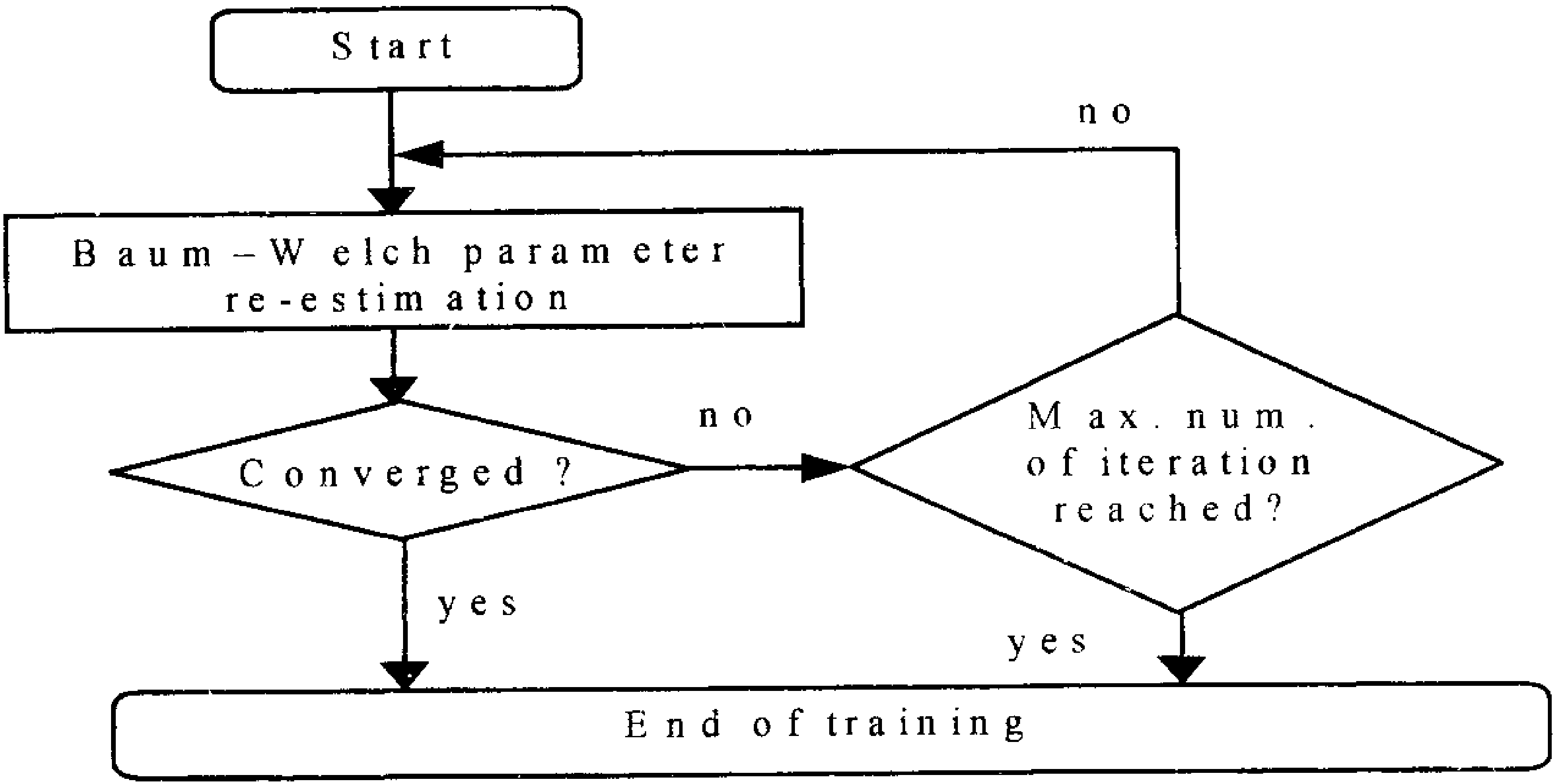


Figure 5.3 Training procedure

References

- *Rabiner and Juang, Fundamentals of Speech Recognition, Prentice Hall*
- *Rabiner, A tutorial on HMM and selected applications in Speech Recognition*
- *J Glass, Speech Recognition, Spring 2003, Open Course Ware, MIT*
- *Speech Recognition, Course AM 0282*
- *Andrew Moore, Tutorial on HMM @ http://www.autonlab.org/tutorials/hmm.html*
- *C Bechetti, Speech Recognition: Theory and C++ Implementation*
- *Various Other sources*